Representing bioinformatics causality

Johannes Mandel, Niall M. Palfreyman, Jesus A. Lopez and Werner Dubitzky

Date received (in revised form): 17th June 2004

Johannes J. Mandel
is a PhD student working in a collaboration project between University of Ulster and Weihenstephan University of Applied Sciences. His research project is on deriving Gene Regulatory Networks from Expression data.

Dr Niall M. Palfreyman
is Professor for Bioinformatics at Weihenstephan University of Applied Sciences. His research interests include bioinformatics, systems biology, cognitive modelling and general dynamical modelling.

Dr Jesus A. Lopez
is a Lecturer in Bioinformatics at the University of Ulster. His research interests include systems biology, knowledge modelling and computational intelligence applications in medicine and biology.

Dr Werner Dubitzky
holds a Chair in Bioinformatics and is Head of the Bioinformatics Research Group at the University of Ulster. His research interests include bioinformatics, systems biology, data and text mining, artificial intelligence and grid technology.

Keywords: *bioinformatics, biotechnology, systems biology, dynamical modelling, simulation, formalisms*

J. Mandel,
Dept. of Biotechnology & Bioinformatics,
Weihenstephan University of Applied Sciences,
Freising,
Germany

Tel: +49 8161 71 3054
Fax: +49 8161 71 5116
E-mail: johannes.mandel@fh-weihenstephan.de

Abstract

This paper reviews a variety of different graphical notations currently in active use for modelling dynamic processes in bioinformatics and biotechnology, and crystallises from these notations a set of properties essential to any proposal for a modelling language seeking to provide an adequate systemic description of biological processes.

THE ROLE OF DYNAMICS IN BIOINFORMATICS

As bioinformatics moves into the post-genomic era it becomes more and more important that it concerns itself with a principled account of *process*. As researchers start to investigate the meaning of genomic data, it becomes clear that this meaning is inextricably entwined with the biological processes of which it forms an integral part.

Palfreyman[1] has argued that process lies at the heart of the concept of information, since it is the dynamical transformation of state that defines the informational content of that state. If we regard bioinformatics as the study of the storage and processing of information in biological systems, then the central concern of bioinformatics becomes *the study of dynamical effects in biological systems*.

This standpoint places the dynamical modelling of biological processes firmly on the centre stage of bioinformatics, and is therefore a significant point of overlap between the endeavours of bioinformatics and systems biology. Our central aim in this paper is to analyse the notations currently used in dynamical modelling work in bioinformatics with a view to crystallising from these notations a set of properties essential to the adequate description of systemic biological process.

EVALUATING DYNAMICAL MODELLING NOTATIONS

Criteria for evaluating dynamical notations

In order to evaluate dynamical modelling notations for bioinformatics, we shall need to be aware of the criteria which must be fulfilled by such a notation, and these criteria are in turn defined by the needs of the various interest groups with a stake in the notation. The stakeholders explicitly considered in this paper are: biochemists, molecular and developmental biologists, systems biologists and biotechnological process engineers. We can group the requirements of a language of biological process under the following broad headings:

- **Transformation:** crucial to the study of cellular metabolic processes are the enzymatic reactions controlled by gene products. Hence a bioinformatic process notation should adequately describe networks of catalysed metabolic transformations.

- **State-transition:** the state of a particular gene can be defined in terms of its expression rate, and the pattern of expression rates over the entire genome of a cell defines the state of that cell. Stable states define for example the differences between the different cell types in a typical

eukaryote. A dynamical notation should be able to describe both the maintenance of such states and also the transition from one state to the next.

- **Transport:** an important aspect of bionetwork modelling concerns the transport mechanisms of biotechnologically relevant substances or organisms. A process notation should be able to describe transport networks *within* cells (eg phosphate transport to mitochondria), *between* cells (eg signal transduction) and *of* cells (eg within a fermenter or developing embryo).

- **Creation/destruction:** of major importance in gene-regulatory and metabolic processes is the production and decay of chemical signals – if messenger substances survived permanently in the cell interior or environment, they could not function as signals. An important issue is therefore the convenient representation of chemical sources and sinks.

- **User acceptance:** all of the above stakeholders are in some sense *biologists*. Any tool or notation that is to be used by biologists must adhere to the cultural norms of the biologist – graphical symbols and their meaning must be palatable to the life scientist if they are to find acceptance with this user group.

- **Availability:** it is important that any dynamical notation used to model biological processes should be available for use by researchers. This means on the one hand that the notation should lend itself to informal pen-and-paper discussions, but also that the notation should be available in the form of a convenient software tool.

- **Animation/execution efficiency:** systems biology concerns itself with modelling entire networks of reactions within the cell or organism, and in particular in simulating these models *in silico*.[2] A biological process formalism should therefore provide facilities for efficiently executing a dynamical model to investigate its time development.

- **Stochastic behaviour:** there exists an inherently stochastic component in many biological systems such as the lysis/lysogeny switch in λ-phage. The animation of a process model should therefore permit the incorporation of such a stochastic element.

The idealised lac model (ILM)

Modelling the lac-operon

The ILM contains elements of signalling, gene regulation and metabolism

To provide a standard measuring stick against which to evaluate dynamical modelling notations a greatly idealised model of the lac-operon is introduced here.[3] This model has the advantage that it incorporates each of the criteria presented above, containing elements of signalling, gene regulation and metabolism. In this idealised lac model (ILM – see Figure 1), the three-gene internal structure of the lac-operon is ignored and instead the following highly schematic account of its role in the hydrolysis of lactose is given.

We start our description of the ILM when extracellular lactose (*Lx*) is present in the intercellular environment. In this case lactose *permeates* at a certain, very low, base rate into the cytosol to raise the level of intracellular lactose (*L*). *L* has the effect of switching a particular repressor protein in the cell between two possible conformational states: active (*R*) and passive (*Rp*).

In the genome of the ILM is a single promoter *P* which is responsible for initiating *transcription* of an enzyme complex *E*. The complex *E* performs the dual function of raising the permeation rate of lactose (*L*) into the cell and then splitting this lactose molecule into one molecule each of glucose (*G*) and galactose (*Ga*). *P* is in turn activated by two factors: the *presence* of catabolite-

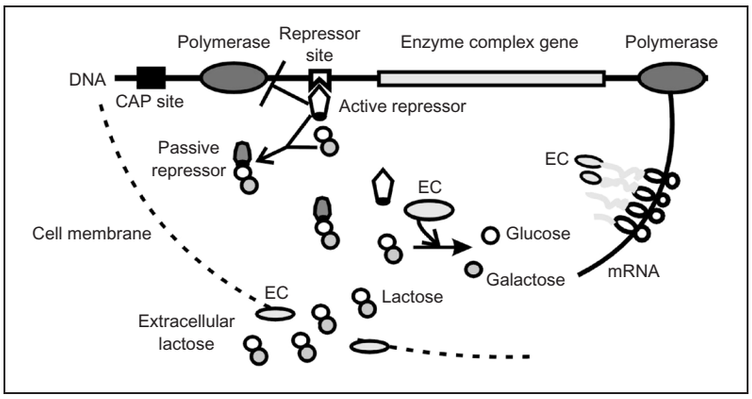

Figure 1: Conventional biological representation of the ILM. CAP = catabolite-activation protein; EC = enzyme complex

activating protein (CAP) and the *absence* of the active repressor protein conformation R. Thus there exists at least one balancing feedback loop in this model, via which L influences its own rate of splitting to *G* and *Ga*.

Although the ILM is a deliberately idealised abstraction, it contains in a realistic example all of the evaluation criteria mentioned in the section above on 'Criteria for evaluating dynamical notations' for a dynamical notation for bioinformatics. In the following section we shall critically compare a variety of dynamical formalisms available for modelling bionetworks, using the ILM as a common scale against which they can be assessed.

DYNAMICAL MODELLING NOTATIONS IN CURRENT USE

Ordinary differential equations model change over time

Description

ODEs describe systems in terms of functions and their derivatives

Probably the oldest notation for modelling change over time is the language of *ordinary differential equations* (ODE). ODEs have a long and distinguished history in dynamical modelling since their introduction by Newton and Leibnitz in the 17th century to describe the dynamics of physical systems. Since then they have been used to represent all manner of dynamical systems in the sciences. In addition, Euler was the first to offer in the 18th century an efficient means of solving ODEs numerically irrespective of their complexity. The nature of their solutions has been analysed particularly intensively in the last 30 years in relation to chaos theory (see, for example, Strogatz[4]).

By defining a number of representative constants, we arrive at an ODE formulation of the ILM presented in the equations 1 to 8 below. This set of equations is a straightforward translation of the ILM into mathematics. The variables *E*, *G*, *Ga*, *L*, *Lx*, *R* and *Rp* on the left are the *state variables* of the model. The change d*X* of the state variable *X* over the time interval d*t* is expressed by the expression d*X*/d*t*. The first seven equations on the right are the initial conditions for the model.

- *P*, CAP_site and R_site are (Boolean) variables relating to the current activation state of the promoter P;
- degrade_rate represents the degradation rate of the enzyme complex E;
- base_permeation is the very low unfacilitated base permeation rate of lactose into the cytosol; and
- the constants pToA and aToP define the exchange rate of the repressor protein R between its active and passive conformations.

$$\frac{dE}{dt} = P - \text{degrade\_rate} \cdot E \qquad E(0) = 0 \quad (1)$$

$$\frac{dG}{dt} = f(E, L, G, Ga) \qquad G(0) = 0 \quad (2)$$

$$\frac{dGa}{dt} = f(E, L, G, Ga) \qquad Ga(0) = 0 \quad (3)$$

$$\frac{dL}{dt} = (\text{base\_permeation} + E)\cdot Lx - f(E, L, G, Ga) \qquad L(0) = 0 \quad (4)$$

$$\frac{dLx}{dt} = -(\text{base\_permeation} + E)\cdot Lx \qquad Lx(0) = 5 \quad (5)$$

$$\frac{dR}{dt} = \text{aToP}\cdot L - \text{pToA} \qquad R(0) = 1 \quad (6)$$

$$\frac{dRp}{dt} = \text{pToA} - \text{aToP}\cdot L \qquad Rp(0) = 0 \quad (7)$$

$$\left.\begin{array}{l} P = 1 \text{ if (CAP\_site and not R\_site), else } 0; \\ \text{CAP\_site} = \text{True}; \\ \text{R\_site} = (R > 0.2); \\ \text{degrade-rate} = 0.1; \\ \text{pToA} = 0.1; \\ \text{base\_permeation} = 0.01; \\ \text{aToP} = 20 \end{array}\right\} \quad (8)$$

Discussion

ODEs offer a number of advantages as a dynamical formalism, not least of which is simply the overwhelming ubiquity of ODEs in the dynamical systems literature. The formalism interfaces seamlessly with the whole of mathematics, which enables the description of systems of arbitrary complexity within a unified framework. In addition, the numerical solution methods of Euler, Runge-Kutta and others enable these equations to animate a system such as the ILM in the sense of solving the equations numerically through time.

Yet there are also clear disadvantages to ODEs as a notation for bionetworks, and these relate to the representation of the flow of matter and of information in the notation. When we look at the set of ODEs 1–8 we notice that a number of essential aspects of the ILM are not made explicit by equations. One example of such an implicit dependence is the conserved material flow between *R* and *Rp*, and also between *L* and *Lx*: when lactose disappears from the extracellular environment, it appears within the cytosol. This flow is at least obscured by the mathematical notation.

Another related case is the use of symbols to represent information dependency in the model: equations 2–7 all refer to *L*, yet it is difficult for the reader to see precisely how these all fit together. In actual fact, three separate relationships are being represented here – the permeation of lactose into the cytosol, the splitting of lactose to glucose and galactose, and the lactose-activated reconformation of R. Yet the equations blur the boundaries between these distinct dynamical aspects in a way that obscures their physical interpretation.

While these issues may seem trivial to the experienced mathematician, our usability tests suggest that for the biologist they can represent a significant obstacle to understanding the model. The problem becomes far more acute when we look at a more realistic set of ODEs such as those used by Tyson[5] to model the regulation of M-phase-promoting factor in frogs' eggs. To discern the many flows and dependencies from the equations in that article is definitely a non-trivial exercise for the uninitiated.

Petri nets model cumulative change

Hybrid Functional Petri Nets

Description

Petri nets (PNs) were developed by Petri (1962) to model dynamical systems governed by discrete state transition rules. Petri nets have since found application in software design, business process management and networking protocols. They possess unambiguous executable semantics for animation, and can be represented both graphically and mathematically. Reddy *et al.*[6] were the first to use PNs to model metabolic pathways; Hofestaedt and Thelen[7] then demonstrated their usefulness for modelling regulatory bionetworks.

Hybrid functional PNs (HFPN) are referred to by David and Alla,[8] and were

developed further by Drath[9] and later by Matsuno *et al.*[10] to simulate bionetworks. HFPNs extend the original PN formalism in two respects: first, they extend the semantics from discrete to continuous transitions, and secondly they permit limited dynamic reconfiguration of the net by using functions to define the transitions.

An HFPN is a directed graph containing two kinds of nodes: *places* $P = \{p_i\}$ (represented by singly and doubly drawn circles in Figure 2) and *transitions* $T = \{t_i\}$ (filled or unfilled bars). An HFPN also contains three kinds of arcs $A = \{a_i\}$: *normal* (solid line), *inhibitory* (filled line terminating in circle) and *test arcs* (dotted line). Arcs connect places with transitions to provide flow routes through the net, and with each arc can be associated a weight w_i. In Figure 2, the ILM is represented within the HFPN formalism.

Cumulative change in Petri nets

Tokens represent the current dynamic state of an HFPN at each instant, affording the animation of the net. Each place p_i contains a number of tokens $M(p_i)$ (the *marking* of p_i), acting as an accumulator of tokens until they are passed on via transitions to downstream places. Thus in Figure 2 the passage of a token from the place Lx to the place L represents the permeation of extracellular lactose into the cell as a change of state from 'outside the cell' to 'inside the cell'. Tokens are produced and consumed when a transition is activated ('fired'). Transitions are activated on fulfilment of some precondition $\mathrm{pre}({}_{\mathrm{up}}p_{ti})$, where the ${}_{\mathrm{up}}p_{ti}$ are the upstream places of the transition t. When t fires, the marking of the upstream and downstream places is updated according to the post-conditions:

$$M_{\mathrm{time}+1}({}_{\mathrm{up}}\,p_{ti}) = M_{\mathrm{time}}({}_{\mathrm{up}}\,p_{ti}) - {}_{\mathrm{up}}w_{ti}\nu(P)$$

$$M_{\mathrm{time}+1}({}_{\mathrm{down}}\,p_{tj}) = M_{\mathrm{time}}({}_{\mathrm{down}}\,p_{tj}) + {}_{\mathrm{down}}w_{tj}\nu(P_t)$$

where P_t is the set of all places connected with t; $\nu(P_t)$ is a function associated with t (the *firing speed* of t) of the elements of P_t; ${}_{\mathrm{up}}w_i$ is the arc weight associated with ${}_{\mathrm{up}}p_{ti}$; and ${}_{\mathrm{down}}p_{tj}$ is the jth downstream place from t.

For discrete transitions the firing condition is checked in discrete time steps, while a continuous transition is constantly checked and, if active, is continually fired with the associated firing speed $\nu(P_t)$. In an enzyme kinetics application the speed ν could for instance be set to the Michaelis–Menten function for enzymatic reaction rates. Isolated transitions can serve as sinks and sources.

HFPNs also distinguish between discrete and continuous places. Discrete places can contain only an integer number of tokens while continuous places can hold fractional numbers of tokens. Normal arcs represent the flow of tokens between places and transitions. Together with weights, these mechanisms are useful for representing chemical reactions with varying proportions of substrates and products. Test arcs and inhibitory arcs modulate transitions without consuming tokens from the upstream places, modelling, for example, the regulatory effects of R_site and CAP_site on P in Figure 2.

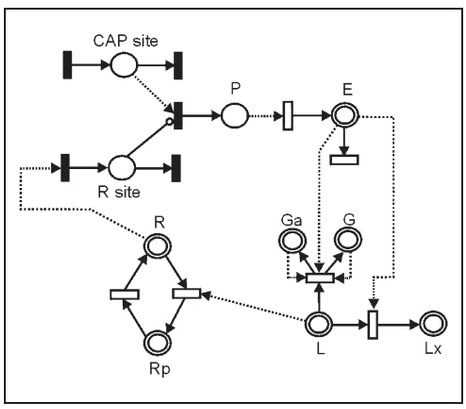

Figure 2: HFPN representation of the ILM

Discussion

The crucial contribution of Petri nets to dynamical modelling is the description of *cumulative change*. The accumulation and depletion of tokens at a place models dynamical change as a stepwise historical

process, resulting from the accumulation of many small changes as tokens flow from one place to the next. This makes HFPNs particularly suitable for the modelling of metabolic bionetworks, where the chemical composition of the cell proceeds through many cumulative reaction steps. As Figure 2 illustrates, HFPNs can be used, and indeed *are* being used, to describe catalysed biochemical reactions, balanced reactions, multimolecular reactions and gene regulation. The notation also permits encapsulation in which, for example, a transition can represent a subnetwork containing additional places and transitions.

Modelling large-scale networks with Boolean networks

Two particularly appealing aspects of the HFPN are the use of transitions to represent sources and sinks, and the use of weighted arcs. Together, these two features are responsible for a significant reduction in the complexity of the diagram, since they make it possible to create new tokens at any transition, modelling non-conserved token flows such as the multimolecular reaction $L \rightarrow G + Ga$.

However, the application to bioinformatic modelling also shows up certain failings of HFPNs. As Zevedei-Oancea and Schuster[11] note, a minor but ugly feature is the use of two transitions to model the single reversible reaction $R \leftrightarrow Rp$. A more serious problem is the awkward use of discrete places containing either one or zero tokens to represent the Boolean regulation of the promoter *P*. The two places R_site and CAP_site no longer function here as accumulators, but rather as Boolean switches, leading to a confusing and inelegant representation of regulatory pathways.

Places as Boolean switches

Ahistorical dynamic behaviour

Finally, the execution semantics of HFPNs interface clumsily with the semantics of ODEs. The HFPN comes close, but not quite close enough, to functioning as a numerical integrator of ODEs, but lacks in particular the notion of a time interval d*t* over which integration occurs.

Boolean networks model functional change

Description

Kauffman and coworkers[12,13] originally introduced Boolean networks as an adequate model for studying generic properties of genetic networks. Boolean networks are based on Boolean logic and have been widely used to model large gene regulatory networks (Akutsu *et al.*,[14] Shmulevich[15]). Platzer[16] uses Boolean networks to construct a large executable model of cell differentiation in the *Caenorhabditis elegans* embryo.

Boolean networks are directed monopartite graphs. Each node can be in one of two states which are named variously *one/zero*, *on/off* or *true/false*. The totality of states of all nodes at each instant defines the current state of the entire system. Arcs represent the logical influence of the upstream node on the downstream node. An extension introduced by Mendoza and Alvarez-Buylla[17] uses weighted arcs w_i to model different levels of influence between nodes.

At each time-step all nodes of a Boolean network are updated in response to the incoming signals s_i from upstream nodes. The sum of these signals is compared with a threshold value θ associated with the downstream node (indicated inside the nodes of Figure 3); if $\Sigma w_i s_i \geqslant \theta$, then the state of the downstream node becomes 1 in the next time-step; otherwise the state becomes 0. It is important to recognise that this update algorithm *only* makes reference to signals from upstream nodes; it explicitly makes *no* reference to the previous state of the downstream node. Thus Boolean networks simulate *ahistorical* dynamic behaviour.

We can best understand how Boolean networks function by considering the gene regulatory components of the ILM representation shown in Figure 3. Here the nodes CAP_site and R_site are connected to the node *E* via arcs with respective weights $w_{\text{CAP\_site}} = +1$, $w_{\text{R\_site}} = -1$, and *E* has a threshold value

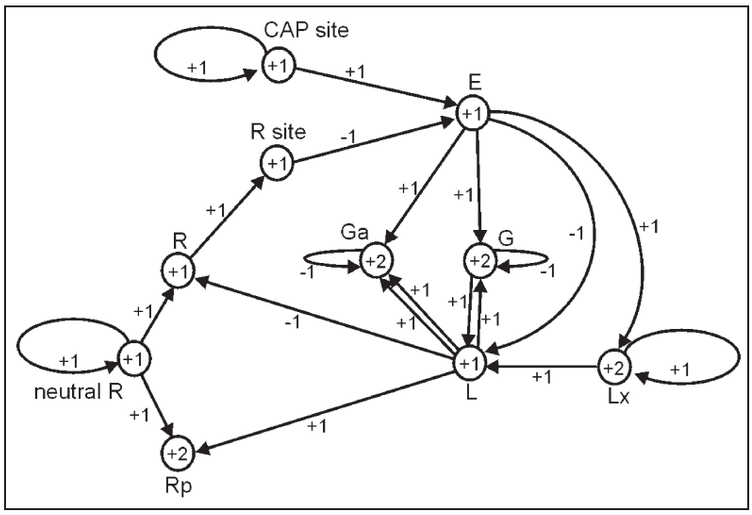

Figure 3: Boolean network representation of the ILM

Incorporating uncertainty

$\theta_e = 1$. Let us suppose that at some instant the upstream nodes have the respective signals $S_{CAP\_site} = 1$ and $S_{R\_site} = 0$. The transition to the next state of the network is then calculated by forming the inner product of the signals and weights, giving $i_{CAP\_site}w_{CAP\_site} = i_{R\_site}w_{R\_site} = 1\cdot(+1) + 0\cdot(-1)$. Since this value is greater than or equal to the threshold value θ_E, the state of E at the next time-step is 1. In this way the network models the regulation of the enzyme E by R and CAP. If the state of R_site were changed to 1, then our time-step calculation would become $1\cdot(+1) = 1\cdot(-1) = 0$; since this is less than the threshold, the state of E at the next time-step would be 0.

Discussion

Functional change in Boolean networks

The central feature of Boolean networks is *functional*, as opposed to *cumulative*, change. Whereas cumulative change proceeds in a stepwise, historically dependent fashion, *functional change* occurs instantaneously in an ahistorical fashion. Consider for example the reaction of the expression of E to changes in the nodes R_site and CAP_site: if these nodes are set to the 0 and 1 states respectively, then E is switched on regardless of its previous state history.

Cyclic dependencies within dynamic Bayesian networks

Functional change is perfectly suited to the instantaneous switching which plays such an important part in gene-regulatory networks. It is less appropriate for the historically conditioned dynamics of metabolic networks, as we see from the rather inelegant representation of the ILM reaction $L \rightarrow G + Ga$ catalysed by E. Notice also how the ongoing state of the node CAP_site must be maintained *off* by means of a self-stimulatory arc in the ILM. This is an artefact arising from the ahistorical nature of functional change in Boolean networks.

In summary, Boolean networks are well suited to representing the functional change of regulatory bionetworks, but not to the cumulative change of metabolic networks. In addition Huang[18] has noted that the restriction of the nodes to Boolean functions means that they are more appropriate for describing fundamental, generic principles rather than quantitative biochemical principles.

Dynamic Bayesian networks describe stochastic behaviour

Description

Bayesian networks (BN) model causal relationships between stochastic variables.[19] They incorporate uncertainty into the dynamics of networks.

BNs were developed and introduced by Pearl[20] and have been applied in the areas of text analysis, medicine, engineering and image processing. Their application to bionetworks has been investigated by Friedman *et al.*,[21] Pe'er *et al.*[22] and Hartemink *et al.*[23] Hartemink, for example, used BNs to study gene regulation of galactose metabolism in *Saccharomyces cerevisiae*. Since BNs cannot model cyclic dependencies, the full application of BNs to dynamical systems with feedback only became possible with the work of Perrin[24] on dynamic Bayesian networks (DBN), which unfold dependencies into relationships between the successive time-steps t and $t + 1$ (see Figure 4). Friedman first used DBNs to reverse-engineer gene-regulatory

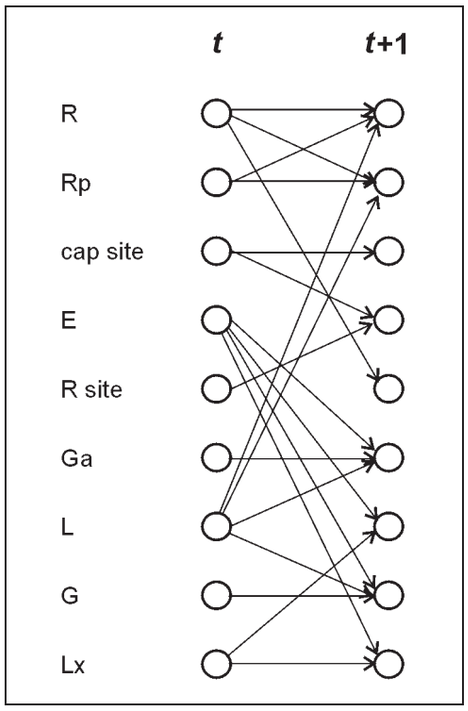

Figure 4: DBN representation of the ILM

networks from microarray time series data.

Block diagrams unravel complex structures

Bayesian networks are directed *acyclic* graphs representing *joint probability distributions* $P(X_1,X_2,\ldots,X_n)$. Nodes represent dynamical variables, and arcs represent postulated causal dependencies between these variables.[25] As shown in Figure 4, by unfolding the dynamical states at instants t and $t + 1$, DBNs become capable of representing causal feedback loops in an acyclic fashion. In a DBN the probability $P_{t+1}(X_i)$ at time $t + 1$ is conditional on a set of prior probabilities $P_t(X_i)$ at time t; for example in Figure 4 the causal dependence of R at time $t + 1$ on R and Rp at time t is formulated as $P(R_{t+1} \mid R_t, Rp_t)$.

Discussion

The application of BNs to gene regulation rightly asserts the stochastic nature of many biological systems such as the lysis/lysogeny switch in λ-phage, and even, arguably, the inhibition of repressor protein by lactose in the ILM. Bayesian networks are statistical models that integrate uncertainty into the modelling process, and so are effective in modelling such systems. In addition, by introducing the possibility of circular feedback, DBNs open the possibility of reverse-engineering feedback networks from expression data using standard optimisation techniques for Bayesian networks.

Yet there are also a number of disadvantages to Bayesian networks. First, they suffer from many of the same problems as Boolean networks, since their functioning is essentially ahistorical in nature. This makes the modelling of metabolic reactions at least problematical. Second, the very trick that enables DBNs to cope with cyclic dependencies also plays havoc with the readability of the formalism. If we compare Figure 3 with Figure 4, we can see how the enzymatic reaction $L \rightarrow G + Ga$ becomes in the Bayesian network formalism even more opaque, if possible, than in the Boolean network notation.

Signal-flow diagrams link functional change to cumulative change

Description

A fundamental tool in control systems engineering is the block diagram, used to analyse the structure of complex systems into collections of simpler relationships. Block diagram models originate from engineering areas such as signal processing and feedback control theory,[26] and are widely used to simulate the behaviour of mechanical, thermodynamic, electronic and control systems.[27]

Signal-flow diagrams are a form of block diagram that describe dynamical systems in a very detailed way by resolving the blocks of the block diagram into basic mathematical functions and signals, and mapping these onto ODEs. A block diagram model is a multipartite graph consisting of different blocks and directed arcs: blocks define the dynamical subsystems of the system under study, and arcs define the relationships in form of signals between these subsystems. The

Combination of functional change and cumulative change in signal-flow

purpose of the nodes in a signal-flow diagram is therefore to accept incoming signals from upstream nodes, process them, and pass on the result to downstream nodes.

In a signal-flow diagram, unfilled nodes indicate mathematical processing of incoming signals while small filled nodes represent signal branching. Looking at the signal-flow formulation of the ILM in Figure 5, we see for example that integration of inputs is indicated by the marking '$\int dt$' and multiplication of inputs by '×'. The arcs of a signal-flow diagram map the passing of signals from upstream nodes to downstream nodes – simulation is achieved by iterating this passing of signal values through time.

Discussion

Signal flow diagrams are mathematically detailed

The signal-flow formalism is a familiar tool in control engineering applications, where it is commonly used to model biotechnological and chemical industrial processes. It represents an interesting amalgamation of Boolean networks and Petri nets in that it combines functional with cumulative change in one diagram, thus making it a potential candidate for modelling general bionetworks. This merging of the two forms of change is achieved by the annotation of its nodes: functions represent functional change, while integrations represent cumulative change. Thus a function node such as those marked '×' in Figure 5 generates a functional change in its output value (output equals product of inputs), whereas an integration node (marked '$\int dt$' in Figure 5) generates a cumulative change in its output value (output equals integral of inputs over time). Nodes marked 'C' represent constants.

Furthermore, signal-flow diagrams interface seamlessly with the mathematical ODE notation earlier in the section 'Ordinary differential equations model change over time' to the extent that an experienced mathematician can read off the equations represented by the diagram. Yet this very 'mathematicalness' of signal-flow diagrams also makes them rather inaccessible to workers in biology. Their extreme mathematical flexibility makes few concessions to readability, with the consequence that the diagram rapidly becomes filled with many confusing intermediate nodes (for example the product nodes in Figure 5) which are essential to the mathematical accuracy of the model.

A further cosmetic deficiency of signal-flow diagrams is that signals affect only downstream nodes, and not upstream nodes. This presents a problem when modelling transport or transformation processes such as the splitting of lactose, which must orchestrate the accumulation and depletion of upstream and downstream nodes. This orchestration is achieved only by splitting outputs into a feedforward and a feedback component, leading to the somewhat messy representation of the reaction $L \rightarrow G + Ga$ in Figure 5.

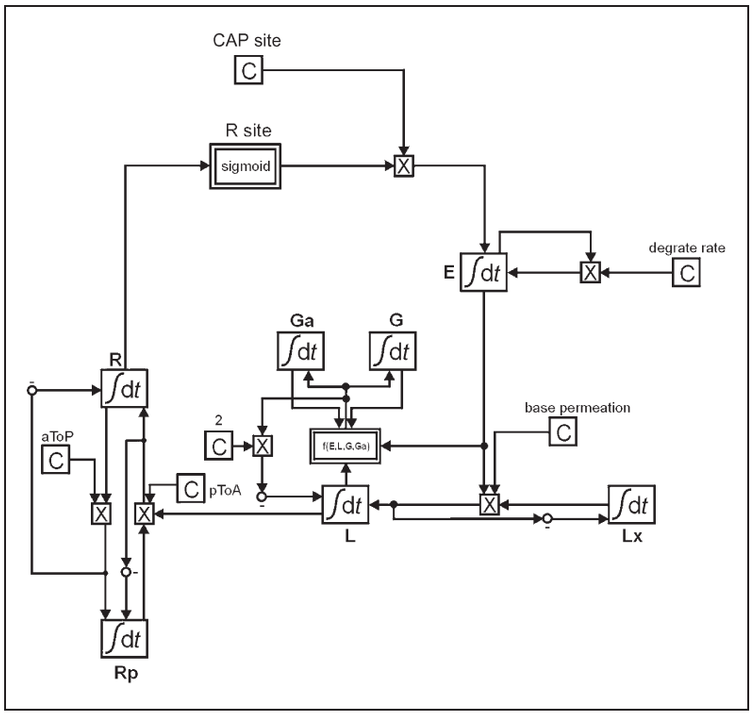

Figure 5: Signal-flow representation of the ILM

System dynamics models cumulative change as material flow

Description

System dynamics (SD) is a mature, well-documented body of techniques used in management science since its conception by Forrester[28] to analyse business and other social systems. It embodies a wealth of experience in such varied modelling areas as: corporate and public management, biological and medical modelling, environmental policy-making and complex non-linear dynamics. Its user acceptance is such that it has been selected in several American states as the basis for integrated teaching in disciplines as varied as physics, geography and English literature.

Describing the flow and accumulation of conserved material

A *stock and flow diagram* is used in SD to model some simplified part of the world in terms of information dependency, material flow and accumulation. To understand this notation, let us take our ILM and represent it as the stock and flow diagram in Figure 6.

This representation of the ILM is fairly self-explanatory. The rectangles are *stocks* (for example *L*), and the doubly drawn arrows leading to and from the stocks are *material flows* (for example from *Lx* to *L*). This flow represents the permeation of lactose into the cell, which *accumulates* in the lactose stock. On the other hand the stock of lactose is also drained by being split into glucose and galactose. Thus the two flows connected to *L* represent the following ODE:

$$dL/dt = \text{permeation} - \text{splitting}$$

In general, a stock is nothing other than the accumulation, or integral, of the material flows to and from it, and the execution semantics of a stock and flow diagram amount to nothing other than the numerical integration (Euler or Runge-Kutta) of the material flows in the diagram.

Finally, in the stock and flow diagram a singly drawn arrow represents a functional *dependency* between different variables. For example, in Figure 6 there exists a dependency between the enzyme complex *E* and the rate of permeation of lactose into the cell; this dependency is represented as a single arrow from *E* to the permeation flow. Notice that dependencies can never lead to a stock, since a stock can never be determined functionally, but rather only by accumulation (integration). The circles in a stock and flow diagram are called *converters*, and represent useful constants and functions relevant to the model. Clouds represent sinks or sources of material flow.

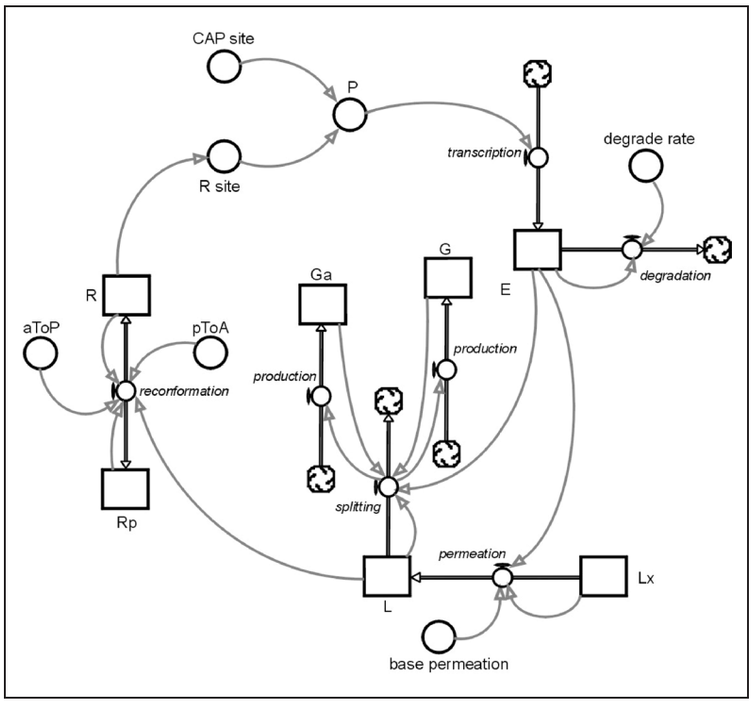

Figure 6: Stock and flow representation of the ILM

Discussion

Like signal-flow diagrams, stock and flow diagrams combine cumulative and functional change within a single formalism, yet *un*like signal-flow diagrams, stock and flow diagrams tie this dichotomy to a physical analogue: the distinction between material flow and information dependency. Material flows correspond approximately to normal Petri net arcs, and dependencies to HFPN test arcs. SD makes explicit the idea that

stocks *integrate* flows, which is why it corresponds so well to the ODE formalism, and the incorporation of an explicit integration time interval d*t* into the SD formalism makes this correspondence complete. On the other hand the dual mechanisms of converters and dependencies introduce an explicit representation of functional change that is eminently suited to the representation of gene regulation logic (compare the rather unwieldy HFPN representation of R_site and CAP_site with the compacter representation in Figure 6).

Homeostasis requires the regulation of flow by feedback

An additional perk of stock and flow diagrams are double-ended material flows. The double-ended flow between the repressor conformations *R* and *Rp* is a notational convenience which reduces the visual complexity of many reversible chemical reactions relevant to bionetworks.

The major *dis*advantage of stock and flow diagrams for bioinformatics becomes apparent when modelling chemical reactions. SD is concerned exclusively with conserved flows in which material is neither created nor destroyed outside explicit sinks and sources. This leads to the unwieldy 'co-flow' representation of the reaction $L \rightarrow G + Ga$ in Figure 6, involving three material flows coupled by information flows.

SUMMARY

In this paper a variety of formalisms in current use in dynamical modelling have been considered. It has been shown that the criteria presented in the section 'Evaluating dynamical modelling notations' lead to a number of properties desirable in any systemic description of biological process. In the following, the findings on the various notations are organised according to their ability to implement these properties from the section 'Evaluating dynamical modelling notations'.

Desirable properties in the systemic description of biological process

- **Transformation:** this concerns support for balanced, multimolecular and catalytic reactions. These can best be modelled with a combination of *cumulative change* and *weighted flows*, implemented most effectively in HFPNs.

- **State transitions:** by this is meant the ability of a system, first, to establish and maintain a homeostatic state, and secondly to execute transitions between such states. The implementation of homeostasis requires the use of *feedback loops* in which flow is regulated by a feedback *dependency*. The implementation of transitions involves *cumulative change*, while the regulation of transitions requires *functional change*. Signal flow diagrams and stock and flow diagrams provide the combination of dependency and flow required for feedback, and Boolean and Bayesian networks implement state transition through functional change.

- **Transport:** support for the various kinds of transport (mass, volume, heat, etc) is provided straightforwardly by *conserved flow*, which is implemented as an integral part of the HFPN and SD formalisms.

- **Creation and destruction:** sinks and sources predicate the existence of both *cumulative change* and *non-conserved flow*. Only HFPN and SD provide explicit support for representing sinks and sources.

- **User-acceptance:** this encompasses two related considerations. The biologist requires from a formalism that (to his or her mind) important distinctions are describable within the formalism, but also that irrelevant distinctions are suppressed in the formalism. One lay user's reaction to BNs was that they were 'simpler' simply because they contain fewer species of nodes and arcs; the ensuing plethora of arcs in the BN was for this user not a consideration. On the other hand, it has been shown that HFPNs'

inelegant formulation of the ILM operator is due to their failure to make the very essential distinction between *functional* and *cumulative change*. Furthermore, the confusing representation of cumulative change in signal-flow diagrams is traceable to their lack of a distinction between *flow* and *dependency*. The only formalism to implement both of these distinctions elegantly is the stock and flow diagram.

- **Availability:** of the notations considered here, stock and flow diagrams, signal-flow diagrams and Petri nets can all be of use in informal pen-and-paper discussions, while ODEs, Boolean networks and Bayesian networks tend to be of more use from a mathematical/computational perspective. Table 1 includes a list of software tools we have found useful in working with the various notations.

- **Animation/execution efficiency:** animation is provided by all the reviewed formalisms. However, only signal-flow diagrams and stock and flow diagrams make explicit the link between animation and the *numerical solution* of ODEs. In our opinion this tends to reduce the impedance mismatch between the biological system and its model, thereby facilitating the optimisation of the model's implementation. Since all of the notations reviewed here involve time-step iteration through frequent large numbers of operations, another relevant aspect of efficiency is the provision of a distinction between functions, which must be re-evaluated in each iteration, and variables, which need not. Of the notations reviewed here, only signal-flow diagrams explicitly address this concern.

- **Stochastic behaviour:** while Bayesian networks are the only notation considered here that explicitly address stochastic behaviour, randomness can be incorporated into all of these notations by using functions returning random values. The λ-phage lysis/lysogeny switch has been implemented in stock and flow diagrams, for instance. One area of modelling that includes an explicit

Table 1: Comparing properties of the reviewed formalisms

	ODE	HFPN	BN	DBN	Signal-flow	SD
Cumulative change	Explicit support	Explicit support	Unsupported	Unsupported	Confusingly supported	Explicit support
Flows	Implicit support	Explicit support	Unsupported	Unsupported	Implicit support	Explicit support
Weighted flow	Implicit support	Explicit support	Unsupported	Unsupported	Confusingly supported	Unsupported
Sinks/sources	Implicit support	Explicit support	Unsupported	Unsupported	Implicit support	Explicit support
Functional change	Explicit support	Confusingly supported	Explicit support	Explicit support	Explicit support	Explicit support
Dependency	Explicit support	Explicit support	Explicit support	Explicit support	Explicit support	Explicit support
Feedback loops	Opaque	Explicit support	Explicit support	Explicit support	Explicit support	Explicit support
Diagram complexity	High	Medium	Low	Low	Medium	Medium
Clear cumulative/functional distinction	No	No	No	No	Yes	Yes
Clear flow/dependency distinction	No	Yes	No	No	No	Yes
Animation semantics	Numerical	Quasi-numerical	Rule based	Probabilistic	Numerical	Numerical
Software tool	**Mathematica**, Wolfram Research	**Visual Object Net**, Ilmenau, University of Technology	None known	**BayesiaLab**, Bayesia	**WinFACT** Vol. 6, **BORIS** Ingenieur-büro Dr Kahlert	**Stella**, ISEE-Systems

ODE = ordinary differential equations; HFPN = hybrid functional Petrinets; BN = Bayesian networks; DBN = dynamic Bayesian networks; SD = system dynamics

stochastic element, but that has been excluded from our discussion for reasons of space, is agent-based modelling.

Our findings in this paper are summarised in Table 1. HFPNs are particularly strong in representing non-conserved reaction flows; Boolean networks are well suited to representing the Boolean switching of gene activations; SD links these ideas together into an intuitively coherent framework; and signal-flow diagrams make explicit the link between these formalisms and the mathematical underpinning provided by ODEs. In addition, the probabilistic approach of dynamic Bayesian networks explicitly incorporates uncertainty into the modelling process.

The next logical step is of course to construct a novel formalism that combines the strengths of these various notations. Each of the notations reviewed here originated in areas of application whose requirements only partially overlap with those of the bioinformatics or systems biology endeavour. The authors are currently engaged in developing precisely such a new formalism, the details of which will be published in a future paper.

Acknowledgments
We should like to thank Ute Platzer for kindly offering her advice on modelling the ILM with Boolean networks.

References

1. Palfreyman, N. (2004), 'The construction of meaning in computational integrative biology', *OMICS: J. Integrat. Biol.*, Vol. 8(2).
2. Tomita, M., Hashimoto, K., Takahashi, K. *et al.* (1999), 'E-CELL: Software environment for whole cell simulation', *Bioinformatics*, Vol. 15(1), pp. 72–84.
3. Jacob, F. and Monod, J. (1961), 'Genetic regulatory mechanisms in the synthesis of proteins, *J. Mol. Biol.*, Vol. 3, 318–356.
4. Strogatz, S. H. (1994), 'Nonlinear Dynamics and Chaos', Perseus Books, Reading, MA.
5. Tyson, J. (2001), 'Analysis of complex dynamics in cell cycle regulation', in 'Computational Modeling of Genetic and Biochemical Networks', Bower, J. M. and Bolouri, H., Eds, MIT Press, Cambridge, MA p. 287.
6. Reddy, V. N., Mavrovouniotis, M. L. and Liebman, M. N. (1993), 'Petri net representation in metabolic pathways', in 'Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology', Hunter, L. *et al.*, Eds, AAAI Press, Menlo Park, CA, pp. 328–336.
7. Hofestaedt, R. & Thelen, S. (1998), 'Quantitative modeling of biochemical networks', *In Silico Biol.*, Vol. 1, pp. 39–53.
8. David, R. and Alla, H. (1992), 'Petri nets and Grafcet – tools for modeling discrete event Systems, Prentice Hall, New York.
9. Drath, R. (1998), 'Hybrid object nets: An object oriented concept for modeling complex hybrid systems; in 'Hybrid Dynamical Systems', 3rd International Conference on Automation of Mixed Processes, ADPM'98, Reims.
10. Matsuno, H. Tanaka, Y., Aoshima, H. *et al.* (2003), 'Biopathways representation and simulation on hybrid functional Petri net', *In Silico Biol.*, Vol. 3, p. 32.
11. Zevedei-Oancea, I. and Schuster, S. (2003), 'Topological analysis of metabolic networks based on Petri net theory', *In Silico Biol.*, Vol. 3, p. 29.
12. Kauffman, S. (1993), 'The Origins of Order: Self-Organization and Selection in Evolution', OUP, New York.
13. Kauffman, S., Peterson, C., Samuelsson, B. and Troein, C (2003), 'Random Boolean network models and the yeast transcriptional network', *Proc. Natl Acad. Sci. USA*, Vol. 100, pp. 14796–14799.
14. Akutsu, T., Miyano, S. and Kuhara, S. (1999), 'Identification of Genetic Networks from a Small Number of Gene Expression Patterns Under the Boolean Networks Model', *Pacific Symp. Biocomputing*, Vol. 4, pp. 17–28.
15. Shmulevich, I. (2002), 'Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks', *Bioinformatics*, Vol. 18, pp. 261–274.
16. Platzer, U. (2003), 'Simulation of Genetic Networks in Multicellular Organisms', PhD Thesis, Heidelberg University.
17. Mendoza, L. and Alvarez-Buylla, E. R. (1998), 'Dynamics of the genetic regulatory network for *Arabidopsis thaliana* flower morphogenesis', *J. Theor. Biol.*, Vol. 193(2), pp. 307–319.
18. Huang, S. (1999), 'Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorgenesis and drug discovery', *J. Mol. Med.*, Vol. 77, pp. 469–480.
19. Kim, S. Y. (2003), 'Inferring gene networks from time series microarray data using

Bayesian networks', *Brief. Bioinform.*, Vol. 4(3), pp. 228–235.

20. Pearl, J. (1986), 'Fusion, propagation, and structuring in belief networks', *Artificial intelligence*, Vol. 29(3), pp. 241–288.
21. Friedman, N., Murphy, K. and Russell, S. (1998), 'Learning the structure of dynamic probabilistic networks', in 'Proceedings of the 14th Conference on the Uncertainty in Artificial Intelligence', Morgan Kaufmann, San Mateo, CA, pp. 139–147.
22. Pe'er, D., Regev, A., Elidan, G. and Friedman, N. (2001), 'Inferring subnetworks from perturbed expression profiles', *Bioinformatics*, Vol. 17, pp. 215–224.
23. Hartemink, A. J., Gifford, D. K., Jaakola, T. S. and Young, R. A. (2001), 'Using graphical models and genomic expression data to statistically validate models of gene regulatory newtworks', *Pacific Symp. Biocomput.*, Vol. 5, pp. 291–433.
24. Perrin, B. (2003), 'Gene networks inference using dynamic Bayesian networks', *Bioinformatics*, Vol. 19, pp. 138–148.
25. Murphy, K. (1999), 'Modelling Gene Expression Data using Dynamic Bayesian Networks', University of California, Berkeley.
26. Doyle, J. C., Francis, B. A. and Tannenbaum, A. R. (1992), 'Feedback Control Theory', Macmillan, New York.
27. Palm, W. J. (2000), 'Modeling, Analysis, and Control of Dynamic Systems', 2nd edn, Wiley, New York.
28. Forrester, J. (1961), 'Industrial Dynamics', Pegasus Communications.

Further reading

Bower, J. M. and Bolouri, H., Eds (2001), 'Computational Modeling of Genetic and Biochemical Networks', MIT Press, Cambridge, MA.

CLE (2004), 'Creative Learning Exchange website (URL: http://www.clexchange.org).

H. De Jong (2002), 'Modeling and simulation of genetic regulatory systems: A literature review', *J. Comp. Biol.*, Vol. 9, pp. 67–103.

L. Hunter *et al.*, Eds (1993), 'Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology', AAAI Press, Menlo Park, CA.