Biological pathways as communicating computer systems

Marta Z. Kwiatkowska[1] **and John K. Heath**[2,*]

[1]Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, UK
[2]CRUK Growth Factor Group, School of Biosciences, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK
*Author for correspondence (j.k.heath@bham.ac.uk)

Journal of Cell Science 122, 2793-2800 Published by The Company of Biologists 2009
doi:10.1242/jcs.039701

Summary

Time and cost are the enemies of cell biology. The number of experiments required to rigorously dissect and comprehend a pathway of even modest complexity is daunting. Methods are needed to formulate biological pathways in a machine-analysable fashion, which would automate the process of considering all possible experiments in a complex pathway and identify those that command attention. In this Essay, we describe a method that is based on the exploitation of computational tools that were originally developed to analyse reactive communicating computer systems such as mobile phones and web browsers. In this approach, the biological process is articulated as an executable computer program that can be interrogated using methods that were developed to analyse complex software systems. Using case studies of the FGF, MAPK and Delta/Notch pathways, we show that the application of this technology can yield interesting insights into the behaviour of signalling pathways, which have subsequently been corroborated by experimental data.

Key words: Computation, Modelling, Pathway

Introduction

You might be reading this paper on a screen. You might have just finished searching for a reference in PubMed, performing a sequence alignment or analysing some microscope images. Biologists are very familiar with the everyday application of computers to their research. These activities exploit computers as calculating devices – they perform calculations much faster and more accurately than biologists and never get bored or distracted. You could go the library and scan every volume in the literature on a particular protein or process; you could manually measure every pixel; you could memorise the sequence of the human genome. It is much more convenient to use a calculating device.

Computers are also communicating devices that have the same decisive advantages of speed and fidelity. You might have a mobile phone that might or might not ring in the next five minutes. You might use a web browser to look at the news or for social networking, depending on your whim. You might print out this paper on a printer down the hall or perhaps in a different building or country. In these examples, the computer doesn't know who is calling you or why you want to review a page, and doesn't look at the contents of the paper you want to print. In its manifestation as a communicating device, its job is to pass information accurately, quickly and effectively to your benefit.

The communicating computer functions by applying rules to processes and components; therefore, it is a reactive system. It responds to inputs and produces outputs that are functions of the timing, order, arrival speed, and so on, of the messages entering the system (maybe nothing happens for a while and then two communications arrive at once). A reactive system is dynamic: the inputs and outputs are in a constant state of flux, and there is no fixed endpoint to the process. Communicating devices are also concurrent: several activities can be executed at the same time and can interact with each other to change their state. Finally, communicating devices are distributed – the components of the device do not have to be in the same place to function (they could be in different buildings or continents). Rules govern communications between the parts of the computing system.

Why should this interest a biologist? In this Essay, we suggest that there is a pressing need for new ways of understanding biological processes. Cohen and Harel have argued that – similar to the communicating devices described above – biological processes are reactive, concurrent and distributed systems (Cohen and Harel, 2007). There is a large body of knowledge and tools in the computer-science community for formulating and analysing communicating computer systems that can be applied to biological problems. In this Essay, we discuss the achievements and challenges in merging these two disciplines. Our purpose is to affirm and to publicise to the cell-biology community the view, first advocated by computer scientists (Regev and Shapiro, 2002), that computers do not have to calculate the behaviour of a biological system, but can be instructed to behave like the system. If computers can indeed faithfully emulate a biological system, we can exploit their speed to accelerate the process of experimental verification.

The current limitations of cell biology

It is commonplace to remark that the last 10 years has seen a step change in the amount of data available to biologists and that the grand challenge is to make sense of this for the benefit of all (Albeck et al., 2006). Curiously, the average cell-biology paper utilises or describes comparatively few data and focuses on one or two molecular entities – there is a mismatch here that makes progress very slow. Why should this be? We now outline several problems that currently confront the cell-biology community.

Understanding by intuition and the problem of biological complexity

The traditional tool of cell-biological understanding is intuition or tacit knowledge. By this we mean a mental picture in the mind of the investigator, developed on the basis of his/her scientific experience and technical expertise, of how the system being studied functions. This mental picture (often communicated in the form of informal diagrams) is used to formulate hypotheses that are then tested by experiment, and the mental picture is adjusted or confirmed. There are two potential problems with this approach. The first is that much biological knowledge and insight becomes the private property of the investigator. There might be concepts of community consensus or divergence but, unlike the physical sciences, very little knowledge (as opposed to data) is formally and rigorously codified in a way that anyone, with appropriate training, could immediately grasp the state of understanding and start to formulate their own experiments.

The second potential problem is that biological processes can be far too complex to analyse by intuition alone. Consider the following example: biological process A activates process B, which, in a time- and concentration-dependent fashion, acts to suppress process A. How do the two processes evolve over time? Looking at process A, there are four possible outcomes: it reaches some steady-state value; it declines to zero over time; it exhibits temporal oscillation; or it displays stochastic fluctuation. Looking at B, the four outcomes apply but the road chosen will depend on what is happening to A, which is the initial driver. Can you predict the dynamic behaviour of A in your head? Let the activity of process A now be influenced by another process, C – it is getting harder! Have you explored every possible behaviour? Which experiment would you do? A temptation here is to simplify the problem by ignoring many possible outcomes or only consider those that, by intuition, lead to an immediate experimentally tractable hypothesis.

This leads us to conclude that the cell-biology community needs more formal and rigorous ways of codifying and analysing the knowledge at its disposal, beyond the informal diagrams that are found in most papers. These methods are not static data formats or databases, which are the realm of the calculating device. Instead, they are dynamic computer models that behave like a biological pathway and that can be analysed and studied as if they were the real thing. In a sense, these models are an automation of the biologist's understanding of the dynamics of a process. Priami has argued that this approach, in effect, transforms 'collections of pictures' into 'spectacular films' of biological pathways (Priami, 2009).

Prioritising experiments

The 'experimental space' of cell biology is vast and, at least given current experimental tools, intractable. Take an arbitrary pathway consisting of ten elements. The standard approach is to eliminate (knock out or inhibit) each element in turn and to observe the effects – this would constitute ten experiments. To exhaustively determine the contribution of removing each element to pathway behaviour, we actually need to eliminate all possible combinations of components, which is 10! or 3,260,000 experiments. Some progress has been made in yeast to address this aim (Deutscher et al., 2008; McGary et al., 2007) but it remains a daunting prospect for many experimental systems, even without beginning to consider the generalisation of these findings to different cellular contexts. Many (most?) such experiments will have trivial outcomes: namely, nothing happens or the system fails. How can we focus resources on those experiments that reveal informative outcomes?

This problem becomes more challenging when we consider quantitative changes in pathway elements – for instance, the rate of an enzyme reaction, the concentration of a component or the affinity of an interaction. These are not fine details for many biological pathways; they are the essence of how the pathway has evolved and how it is fit for purpose (Gerhart and Kirschner, 2007). One possible solution to this challenge would be to compute all possible experimental outcomes from a description of the system and to focus biological effort on testing only those that seem most informative or interesting to the investigator. This would require that the investigator trusts the results of the 'in silico' experiments because s/he cannot carry them all out in the laboratory. Time and cost are the enemies of benchwork.

Integrating different types of information

Much contemporary cell biology is grounded in experimental technique. This is not bad: it is the outcome of experiments that teaches us about the organisation of the process. But which experiment to do? The temptation is to think about a system in a way that is linked to the selected method of investigation. For example, if you employ genetic tools you might think of the pathway as a set of dependencies between different components of the pathway. If you use anatomical tools, you might consider spatially encoded properties, such as the location of a particular protein, as the most prominent element. The tools risk becoming the hypotheses. Most biologists are aware of this temptation and acknowledge the need to bring different types of data (or perspectives) to one place, giving each appropriate weight. However, if you are a specialist in one technique, it is seductive to give priority to that particular type of evidence. One solution would be to have a shared model of the process to which new information of different forms can be readily added and integrated.

It comes as a surprise to scientists trained in physical techniques that much cell-biological information is context-dependent or incomplete. This does not mean that the data are of poor technical quality, but more that they depend on the context in which the data are generated. For example, a pathway might be defined in one cell type, but this understanding cannot necessarily be transferred to all cell types, as they might differ in the composition or abundance of pathway components. For technical reasons, data might be generated in genetically manipulated cells in which the expression levels of components have been boosted or suppressed. Moreover, biologists can 'throw away' data by focusing on a limited part of the pathway. Attempts to circumvent this problem through the generation of large-scale protein-protein interaction or imaging datasets are essential, but careful attention needs to be paid to the quality of the data (Wang and Zhang, 2007) for them to be useful. The potential effects of any manipulations also need to be accommodated when drawing conclusions about the behaviour of the endogenous pathway or generalising to other systems. The impact of manipulations could be investigated by laborious experimentation, with the real possibility that only a few such manipulations in reality affect pathway behaviour or architecture, but knowing which manipulations mattered would represent progress in understanding the process in question.

These reflections lead us to propose that progress in understanding cell-biological pathways would be greatly accelerated if it were possible to do three things: to formally define a hypothetical pathway in a rigorous and portable fashion, to devise methods for rapidly and cheaply exploring parameter values to identify scenarios that command resources for biological evaluation, and to automate the process of pathway interrogation. In fact, it is these considerations that have driven the development of computational methods for analysing the behaviour of communicating computer systems (Milner, 1999). It is therefore appropriate to consider the possibility that these tools could be applied to the analysis of biological pathways. Put another way, how similar is a biological pathway to a communicating computer?

Modelling cellular pathways as communication processes

In an influential paper in the computer-science literature, Regev, Silverman and Shapiro (Regev et al., 2001) built on theoretical work in computer science (Milner, 1999) to propose that the study of communicating systems could indeed provide the tools for analysing interacting dynamic biomolecular systems. The authors suggested that biomolecular processes could be articulated as a set of entities (molecules); such molecules have internal states (e.g. phosphorylated or not) and can interact with each other according to user-prescribed reaction rules (e.g. bind or dissociate, phosphorylate or dephosphorylate, synthesise or degrade, and so on). In other words, biological processes are concurrent, reactive and distributed communicating systems that can be formally described and analysed using the existing tools of computer science. The biological pathway is, in this context, defined by its components, their potential modes of interaction and changes of state, and by the rules (algorithms) that govern the types, orders and rates of events, and in turn determine the behaviour of the pathway. Such a description is thus formally equivalent to the description of a communicating computer device such as a mobile phone.

The core tools of computer science that are used for articulating a biological pathway as a computer program are process calculi. These are computational languages for describing the interactions and communications between processes, which allow the description to be analysed, manipulated and questioned. Process-calculus languages are based around the concept of entities that can change state as a result of a communication event (Milner et al., 1992; Milner, 1999). Multiple entities can communicate concurrently (dependently or independently) and, crucially, an outcome of a communication can be to change the nature of the communication. These computational concepts can be mapped onto biological counterparts, as presented in Table 1. Thus, a molecule (or a cell) can be considered to represent a communicating process; the interaction potential of the molecule can be thought of as a channel; an interaction can be considered as a communication event; and a modification of the molecule can be viewed as a state change. Using these mappings, a biological process can thus, in principle, be articulated in the form of a process-calculus language and then executed and studied in the form of a computer program.

The utility of the algorithmic approach to the biologist can be measured by the extent to which a biological pathway can be realistically described using this limited set of concepts, and by whether the application yields new and biologically useful knowledge.

The essential steps in implementing this approach in a biologically useful manner are shown in Fig. 1. In the first step, a description of the biological process or pathway is specified by the biologist. This is formulated in terms of some basic elements: the molecules participating in the reaction scheme; state changes that occur in the pathway (e.g. phosphorylation or dephosphorylation, synthesis or degradation, and binding or dissociation); the rates at which these changes occur (e.g. transitions per second); and the actual events that occur (for instance, molecule A phosphorylates molecule B, molecule C binds to the phosphorylated form of molecule B), together with a list of the order and dependencies of the reactions described. Each reaction corresponds to a rule – a biological instruction – that results in state changes. An example of such a scheme for the FGF signalling pathway is shown in Fig. 2.

Table 1. The equivalence relationships of the 'biology as computation' concept*

Biological term	Computational term
Molecule	Process
Interaction capability	Channel
Interaction	Communication
Modification	State and/or channel change

*(Regev et al., 2001).

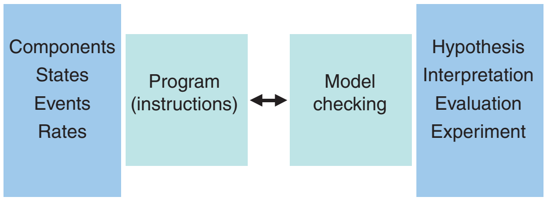

Fig. 1. The essential steps of executable biology [adapted from Fisher and Henzinger (Fisher and Henzinger, 2007)]. In the first step, the biologist defines the biological pathway in terms of components, states, events and rates (see Table 1). This is translated into an executable computer program. The properties of the program are interrogated by model checking. The outcomes of model checking can be tested against biological data, and used to design new experiments or formulate new hypotheses.

At the very least, this description of the process provides the biological community with a formal 'manual' of the process, which can be readily updated and adapted as biological knowledge increases. However, as the description can also be executed ('brought alive'), the description is not static but dynamic, and can be studied as an animation of the biological model (Priami, 2009).

In the second step (Fig. 1), the biological reaction scheme is translated into a computer language that encodes the biological instruction set. There are a number of such languages, each devised with particular applications or properties in mind. Examples that have been used to model biological pathways include Statecharts (Harel, 1987), BioSPI (Priami et al., 2001), Beta-binders and Beta Workbench (Priami and Quaglia, 2005; Dematte et al., 2008), PEPA and BioPEPA (Hillston, 1996; Calder et al., 2006; Ciochetta and Hillston, 2008), Kappa (Danos et al., 2007), Ambients (Regev et al., 2004) and more. Fisher and Henzinger have provided a very useful guide to the different types of computational methods that can be currently used to realise biologically motivated computer programs (Fisher and Henzinger, 2007). New languages and variants are being developed continuously and there is currently no universally applicable or utilised language for biological pathways. We discuss this further below.

In the third step (Fig. 1), the instruction set (program) is executed using the specified rate parameters to generate an output (simulation). The program itself can be modified with ease, for example, to represent a modified binding mechanism or to incorporate (or delete) new components or events. All manifestations of process-calculus programs derived from biological reaction schemes are stochastic, in that they assume that interactions or state changes are governed by probability distributions, and are discrete (the individual molecule is the unit of reaction) (Gillespie, 1977); hence, such process-calculus programs can be thought of as representing a discrete stochastic model of system dynamics. As a result, there is no unique (or idealised) output of the simulation. Consequently, evaluation of the simulation output is initially compared qualitatively to the biological process (Fisher and Henzinger, 2007) rather than tested for an exact numerical fit. The discrete stochastic approach also has the useful property that the behaviour of very small numbers of molecules (even one molecule) can be modelled by this approach.

This is an important conceptual departure from the continuous deterministic approach, which regards the behaviour of the pathway as a continuous and predictable process performed by a population of reactants that is governed by a set of coupled, ordinary differential equations (the 'reaction-rate equations') that yield averaged quantities. In reality, the two approaches can have complementary attributes for the investigator depending upon the question that is posed (Fisher and Henzinger, 2007; Hunt et al., 2008). See, for example, the work by Gaffney and colleagues in which a process calculus model of a pathway is reformulated for analysis by equations (Gaffney et al., 2008).

It is at this third stage that the behaviour of the model can be determined and tested against the investigator's data and hypotheses. For this purpose, a powerful framework known as 'model checking' (Clarke et al., 1999) has been put forward, supported by a family of software tools that test whether a computer program (or hardware device) satisfies specified conditions. In particular, so-called probabilistic model-checking tools have been developed for analysing stochastic models of the type described here (Kwiatkowska et al., 2007), namely, discrete stochastic models arising from biochemical reaction schemes. In this approach, the computational model can be interrogated to study its properties and dynamic behaviour. These include qualitative properties (does the model contain contradictions or other logical errors?) and quantitative properties (e.g. what is the probability of a particular outcome or expected time to degradation?) The analysis of the model is exhaustive, in the sense that the properties are established for all possible executions without actually executing or simulating the program, provided that the model size and complexity does not exceed the computing power.

Model checking therefore allows the investigator to thoroughly test, in an automated fashion, the behaviour of the model against real-world biological data, and to explore scenarios such as changing the rate of a particular reaction (parameter exploration) or removing a particular molecule (in silico genetics). If the behaviour of the model diverges from the experimental data, we can conclude either that the model does not accurately represent the 'real' biological pathway (in which case the model needs to be modified or refined with the incorporation of further biological knowledge), or that the biological data are incomplete (in which case model checking can be employed to identify further experimental conditions that could confirm or refute the model). Model checking can therefore be used as a guide to designing experiments; for example, by identifying time courses or concentrations that might be particularly informative. Model checking is the tool that allows exhaustive (but computing-power-limited) exploration of experimental scenarios and possibilities in silico to identify those that seem most promising or interesting for experimental study in the laboratory. This goes a long way towards meeting the biological challenges outlined above.

Case studies of 'pathways as programs'

Having described the general algorithmic approach to modelling pathways as programs, we need to know whether it actually works in practice as a useful tool for the biologist. We describe three examples that illustrate different ways in which 'pathways as programs' have been used to identify new biological findings through the application of process-calculus and model-checking methods.

Exploring the RAS-RAF-MAPK pathway

An early study of the application of process-calculus formulations combined with model checking used a model of the well-known RAS-RAF-MAP kinase (MAPK) pathway (Calder et al., 2006). One of the first findings was that the original description of the pathway (which had been employed for mathematical modelling) 'deadlocked' when articulated as a computer program, indicating that the biological formulation was incomplete or inconsistent. Having reformulated the model, it was shown that the dynamics of MAPK activation were inhibited by the RAF-binding protein RAF kinase inhibitor protein (RKIP), with differential effects on the singly and doubly phosphorylated forms of MAPK. This finding was in accord with experimental data. The lessons from this study are that a well-understood pathway can be successfully formulated and studied using the process-calculus and model-checking techniques, and that biologists can make errors that formal logic (see below) can detect.

Analysing FGF signalling dynamics

Our own work has centred on a process-calculus model of the fibroblast growth factor (FGF) pathway using stochastic simulation (using BioSPI) and probabilistic model checking (using PRISM) (Kwiatkowska et al., 2006; Heath et al., 2007). Our motivation in this study was to analyse the dynamics (i.e. the duration, amplitude and time-dependent behaviour) of FGF signalling by evaluating a number of different positive and negative regulation mechanisms that had been reported in the literature. These include the action of a tyrosine phosphatase (SHP2), the role of FGF-receptor (FGFR) inactivation by Src-kinase-mediated internalisation, the role of ubiquitin-mediated proteolysis, and the action of the signal attenuator Sprouty, which has been argued to act by sequestration of the signalling adaptor growth factor receptor-bound protein 2 (GRB2) (Hanafusa et al., 2004).

We first constructed and verified a full version of the model (Fig. 2), showing that it yielded outcomes that accorded well with experimental data. We then analysed versions of the model in which individual components had been systematically eliminated (Fig. 3) to study the relative significance of different means of controlling signal propagation. This is in effect an 'in silico genetics' approach. We also undertook a parameter exploration approach, in which we systematically varied the rate of certain reactions to emulate the effects of known (or hypothetical) modes of drug inhibition.

Our main conclusion from analysing the model was that the most prominent determinants of the dynamics of FGF signalling are the relative rates of receptor-dependent kinase activation pathways and Src-kinase-activated inhibition pathways. Thus, removal of Src from the model leads to extended duration of MAPK activation compared to the full model (Fig. 3), and this was subsequently experimentally verified (Sandilands et al., 2007). Another prediction from these experiments is that inhibiting the phosphatase activity of Shp2 paradoxically destabilises and then suppresses MAPK activation (Fig. 3). This arises because the phosphatase acts concurrently on both positive and negative pathways, and the negative pathway in the model 'wins'. The conclusion here is that process calculi can be used to reason about a complex signalling pathway in an informative manner.

Further work in our group on formulating and interrogating models of the Wnt signalling pathway (Tymchyshyn and Kwiatkowska, 2008) and JAK-STAT (Janus kinases; signal transducers and activators of transcription) pathway (Guerriero et al., 2007; Guerriero et al., 2009) has indicated the broad applicability of the algorithmic approach to the analysis of signal transduction networks.

Vulval-cell fate determination in *C. elegans*

Fisher and coworkers (Fisher et al., 2005; Fisher et al., 2007) have created a formal computer model for determination of the fate of *Caenorhabditis elegans* vulval cells that is based on the biological model of Sternberg and Horvitz (Sternberg and Horvitz, 1989). In the Sternberg model, the developmental fate of six vulval precusor cells (VPCs) depends on the integration of two signals, an inductive EGF signal emanating from an anchor cell (AC) and a lateral signal (of the Notch/Delta class) that is induced in response to the primary inductive signal in a time-dependent manner (Fig. 4). The developmental fate of the VPCs is determined by their distance from the AC (and thus the primary induction signal) and their receipt of the lateral inhibition signal: those receiving the strongest primary induction signal adopt a primary fate (1° in Fig. 4), those receiving the strongest lateral inhibition signal adopt a secondary (2°) fate, and the default tertiary (3°) fate is adopted by cells that receive neither class of signal. The system therefore exhibits interdependency, feedback and time delays.

In the computer model, quantitative rate parameters were simplified as HIGH (over-expression of the primary induction

A

1. *FGF* **binds** to the FGF receptor (*FGFR*), forming *FGF:FGFR*
2. *FGF:FGFR* **binds** to *FGF:FGFR*, forming *FGF:FGFR:FGF:FGFR*
3. *FGF:FGFR:FGF:FGFR* **phosphorylates** *FGFR* residues Y653 and Y654 [(Y653,Y654)*FGFR*], yielding (Y653P,Y654P)*FGFR*
4. (Y653P,Y654P)*FGFR* **phosphorylates** *FGFR* residues Y663, Y583, Y585 and Y766 [(Y663, Y583, Y585, Y766, Y653, Y654)*FGFR*], yielding (Y663P, Y583P, Y585P, Y766P, Y653P, Y654P)*FGFR*
5. (Y653P, Y654P)*FGFR* **phosphorylates** *FRS2* on residues Y306, Y349 and Y471 [(Y306, Y349, Y471)*FRS2*], yielding (Y306P, Y349P, Y471P)*FRS2*
6. *Src* **binds** to (Y306P)*FRS2*, forming *Src:*(Y306P)*FRS2*
7. *GRB2:SOS* **binds** to (Y349P)*FRS2*, forming *GRB2:SOS:*(Y349P)*FRS2*
8. *Shp2* **binds** to (Y471P)*FRS2*, forming *Shp2:*(Y471P)*FRS2*
9. *Shp2:*(Y471P)*FRS2* **dephosphorylates** (Y306P, Y349P, Y471P)*FRS2*, forming (Y306, Y349, Y471)*FRS2*
10. *Src:*(Y306P)*FRS2* is **removed** from the model
11. *PLCγ* **binds** to (Y766P)*FGFR*, forming *PLCγ:*(Y766P)*FGFR*
12. *PLCγ:*(Y766P)*FGFR* is **removed** from the model
13. *Sprouty* is **synthesised** in response to *GRB2:SOS:*(Y349P)*FRS2*
14. *Sprouty* **binds** to *Src*, forming *Src:Sprouty*
15. *Src:Sprouty* **phosphorylates** *Sprouty* residue Y55, forming *Src:*(Y55P)*Sprouty*
16. *Src:*(Y55P)*Sprouty* **binds** to *Cbl*, forming *Cbl:Src:*(Y55P)*Sprouty*
17. *Cbl:Src:*(Y55P)*Sprouty* is **removed** from the model
18. *Shp2:*(Y471P)*FRS2* **dephosphorylates** (Y55P)*Sprouty* forming (Y55)*Sprouty*
19. *Grb2:SOS* **binds** to (Y55P)*Sprouty*

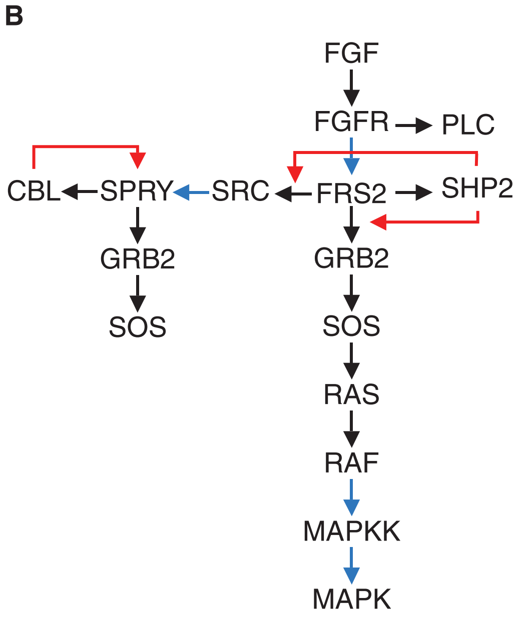

Fig. 2. A computational approach to analyzing FGF signalling dynamics. (A) Example of a text-based narrative of the FGF signaling pathway, which provides the basis for translation into a process-calculus language for execution and model checking (from Kwiatkowska et al., 2006). The full execution can be studied at http://www.prismmodelchecker.org/casestudies/fgf.php#results. In the narrative, molecules (processes) are denoted in italics (e.g. FGF). Specified sites on molecules are denoted in brackets [e.g. (Y653,Y654)FGFR]. Interactions between molecules (communications) are denoted by colons (e.g. FGF:FGFR). Complexes (e.g. FGF:FGFR) are treated as new processes. Modifications (state changes) are denoted in bold (e.g. phosphorylates). Phosphorylation (state change) of specified sites is denoted by adding P after the site identification (e.g. Y653P). Modified molecules [e.g. (Y653P,Y654P)FGFR] are treated as new processes. Each step (line) of the narrative describes an interaction (communication) between molecules (processes), resulting in a modification (state change). Note that, as the narrative develops, the number of types of molecules (processes) changes as the result of previous events. Note also that some steps exhibit dependencies (i.e. a requirement for a particular molecular species to be created in the course of execution) whereas others are present from the start. Thus, multiple steps in the narrative occur concurrently. The narrative can be readily modified by the removal of specific steps or addition of new steps. (B) Diagrammatic version of the FGF signalling pathway articulated in A. Binding reactions are denoted by black arrows, phosphorylation reactions by blue arrows and inhibitory (dephosphorylation or degradation) reactions by red arrows. The RAS, RAF, MAPKK and MAPK components are not explicitly included in the model in A.

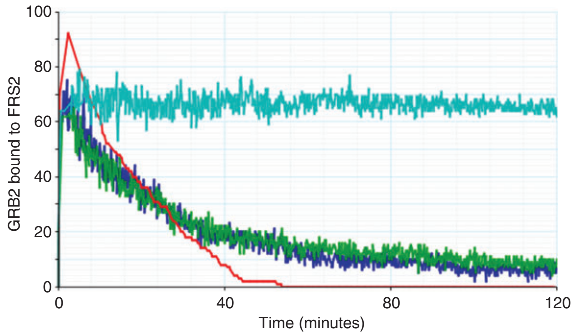

Fig. 3. Example of the investigation of the dynamic behaviour of the FGF pathway by testing the removal of components. The output represents the activation of MAPK (denoted by GRB2 bound to FRS2). Traces show the behaviour of the full model (blue), no SPRY (green), no SHP2 (red) and no Src (turquoise). The model predicts that removal of Src produces extended signalling duration or failure to decay (Sandilands et al., 2007).

signal), MEDIUM (wild-type signal) or OFF (absence of signal). This model has 48 possible initial states corresponding to 48 combinations of mutations or genotypes. Model checking was employed to interrogate the model by calculating the fate of the six VPCs and the reproducibility of all 48 conditions. These conditions correspond to mutations that had been described in the literature and mutant combinations that had not been generated. Of the initial 48 conditions, 44 yielded a stable fate state (i.e. repeated execution of the model yielded the same outcome), including those that conformed to the published mutant phenotypes. The remaining four mutants yielded an unstable fate (i.e. repeated execution of the model yielded different outcomes). Again, using model checking to query the program, it was found that unstable fates were dependent upon variations in the timing of the lateral inhibition signal, which would not have been obvious from analysis of the published mutant phenotypes. The computer-generated phenotypes were then verified by creating the appropriate *C. elegans* mutants and revealing the unstable fate, providing evidence for the significance of timing in the sequential induction of the inductive and lateral signals during VPC development.

This example shows the ability of model checking to test all possible behaviours of a program (in this case, all possible mutant phenotypes) to guide selection of a biological experiment. This is a case in which modelling has accelerated the pace of biological discovery by enabling the prioritisation of experiments.

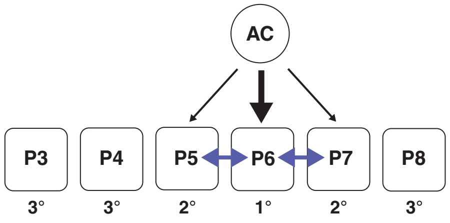

Fig. 4. Graphical model of the *C. elegans* vulval-cell fate specification pathway [modelled by Fisher et al. (Fisher et al., 2007)]. AC is the anchor cell and P3-P8 are the vulval precursor cells. 1°, 2° and 3° denote the normal fates of particular vulval cells. The thick black arrow represents the primary inducing signal from the AC. The thickness of the arrows indicates the relative levels of signal received by the three VPCs shown. In the absence of an inducing signal (as in P3, P4 and P8), the pathway is below the threshold needed for induction and the VPCs adopt the 3° fate. A high level of inducing signal (P6) induces the 1° fate. A high inducing signal also results in the production of a strong lateral signal (blue arrows) by P6 and the suppression of 1° responses in P5 and P7. P5 and P7 thus adopt the 2° fate.

The examples discussed above are encouraging and indicate that the ‘pathway as computer program’ concept can generate new biological insights, has the potential to dramatically reduce the time and cost of exploring experimental ‘space’, and can be used to reason about complex biological processes in a rigorous and formal manner.

Challenges for biologists and computer scientists

Logic and formal methods

The central concept in the science of communicating computing systems is formal methods (FM). FM demands that the means by which the pathway is specified has to be absolutely precise and must represent rigorously provable statements in mathematical logic. FM is needed to identify areas of incompleteness, ambiguity or impossibility in the human-derived model, and to automate the process of exploration once the model is correctly specified. FM is central to the application of model checking, and there is no room for omission or ambiguity if the tool is to have value.

FM approaches were developed to rigorously test ‘mission-critical’ communicating computer systems (such as aeroplane control software) for robustness and reliability without having to crash numerous planes. This is a key point for biologists. If we trust a computer program (as we implicitly trust avionics on our way to a conference) we can share it, which is what we do with an email system or a web browser. Better still, we can explore its value, teach others how to use it and make it a workaday tool like a mobile phone.

There is a price to pay for biologists in meeting FM criteria. We have to be absolutely precise about everything we say to the communicating computer regarding a biological pathway. We cannot rely on a shared community or intuition. The FM computer will reject our ideas as nonsense or flawed. It does not know about our relationship to the pathway; it mechanically checks the rigour of our thinking. However, even the process of defining the model can be valuable in highlighting issues of uncertainty or ignorance in the biological domain.

Language

Biologists do not naturally think in process-calculus languages, but instead speak and write in a code of their own devising. The process of translating a biological vision into a computer language is fraught with the possibility of the vision being ‘lost in translation’ and is a current impediment to the adoption of process-calculus techniques in the biology community. There are multiple languages that are used for the executable-biology approach, each with different attributes and advantages – which to choose? One possible solution is to develop biologically oriented high-level languages that are intuitive to the biologist and automatically translatable into an executable program. We have suggested and implemented such a language for signalling processes, which involves creating a narrative of the pathway from pre-specified elements and reactions (Guerriero et al., 2007; Guerriero et al., 2009). Another approach might be to use one of the emerging data standard formats such as SBML (Hucka et al., 2003; Ciocchetta et al., 2008) or WikiPathways (http://www.wikipathways.org/index.php/WikiPathways) (Pico et al., 2008) to enable biologists to formulate, share and test their ideas.

Parameters

One of the more disheartening roadblocks for biologists in adopting modelling as part of their toolkit has been the insistence in some schools of modelling on the availability of accurate values for the reactions encoded in a model. In most cases these values are unknown or have been derived under non-physiological conditions, represent sparse time series (e.g. western blotting or imaging experiments) or might change in the course of the process. However, a more relevant question than asking the value of every parameter is 'what values dictate the behaviour of the system?' This question can be answered by the model-checking approach, which, by synthesising parametric rate values to ensure the validity of selected properties (Han et al., 2008), can focus the investigator's attention on those values that need to be understood. These techniques are still in their infancy, and should be exploited in biological modelling in due course. However, in the process-calculus approach it is only necessary – at the beginning of the project at least – to know the relative rates of processes encoded in the model (e.g. Calder et al., 2006; Fisher et al., 2007). For instance, protein-protein interactions are fast, enzyme-catalysed reactions are slower, and movement between cellular compartments is slower still. This type of qualitative information can be derived without undue effort for most types of cell-biological investigation. Thus, the ability to easily interrogate the model in a quantitative manner, that is initially qualitative (Fisher and Henzinger, 2007), becomes an advantage in that it becomes possible to focus on those events that have most influence on the behaviour of the model. In this way, the pace of biological discovery can be accelerated.

Conclusions

This Essay has argued that there are strong similarities between biological pathways and communicating computing systems, and that modelling languages and associated computational tools for their analysis exist that can be applied to both types of systems. The key to this lies in being able to describe biological pathways in terms of their components, state changes and interactions in a very precise manner, and to instruct the computer in executing the biologist's vision of the pathway. We have shown that this can be applied in practice to yield biologically informative outcomes, and so the idea of automating biological understanding to accelerate discovery is a realistic vision.

We conclude with two remarks. The 'soft' conclusion is entirely pragmatic. If computer scientists come up with ways of helping biologists do better experiments, have better ideas and publish more papers, then that would be a Good Thing. Every postdoctorate and graduate student would be likely to use the tools, and progress in cell biology would be accelerated. We have witnessed the rapid adoption of useful computational tools for biologists on many occasions and time will tell whether the approach described here will work.

The 'hard' conclusion is more interesting. Computer-science techniques such as process calculus and model checking can help biologists to perform better science, but the similarity between a biological process and a communicating computer system can be taken further. The logical conclusion, for a computer scientist, is a 'biological Turing test'. Harel has proposed that if the 'molecules and cells as computation' approach is valid, it will be possible to create a computer model of a complete cell (or organism) 'which will be deemed valid/complete/adequate if it cannot be distinguished from the real thing by an appropriate team of investigators' (Harel, 2005). Even an approximately specified Turing organism would take biology into new dimensions – we might acquire the ability to study processes that fall outside the lifespan of our research grants (e.g. the accumulation of oncogenic mutations over a human lifespan or the evolution of pathways) and those for which no biological experimental tools have yet been developed.

For reasons of space we have been unable to acknowledge many important contributions to this problem. We are very grateful to Juliet Coates and Robert Insall for comments, and the process-calculus community for insights and explanations. Research in the authors' laboratories is supported by EPSRC and Microsoft Research (M.Z.K.), and Cancer Research UK and FP6 Endotrack (J.K.H.).

References

Albeck, J. G., MacBeath, G., White, F. M., Sorger, P. K., Lauffenburger, D. A. and Gaudet, S. (2006). Collecting and organizing systematic sets of protein data. *Nat. Rev. Mol. Cell Biol.* **7**, 803-812.

Calder, M., Duguid, A., Gilmore, S. and Hillston, J. (2006a). Stronger computational modelling of signalling pathways using both continuous and discrete-state methods. In *Computational Methods in Systems Biology* (*Lecture Notes in Computer Science*, Vol. 4210) (ed. C. Priami), pp. 63-77. Berlin: Springer.

Calder, M., Gilmore, S. and Hillston, J. (2006b). Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. *Trans. Comput. Syst. Biol.* VII **4230**, 1-23.

Ciocchetta, F. and Hillston, J. (2008). Bio-PEPA: an extension of the process algebra PEPA for biochemical networks. *Electr. Notes Theor. Comput. Sci.* **194**, 103-117.

Ciocchetta, F., Priami, C. and Quaglia, P. (2008). An automatic translation of SBML into beta-binders. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **5**, 80-90.

Clarke, E. M., Grumberg, O. and Peled, D. (1999). *Model Checking*. Cambridge, MA: MIT Press.

Cohen, I. R. and Harel, D. (2007). Explaining a complex living system: dynamics, multi-scaling and emergence. *J. R. Soc. Interface* **4**, 175-182.

Danos, V., Feret, J., Fontana, W., Harmer, R. and Krivine, J. (2007). Rule-based modelling of cellular signalling. In *Concur 2007* (*Lecture Notes in Computer Science*, Vol. 4703), pp. 17-41. Berlin: Springer.

Dematte, L., Priami, C. and Romanel, A. (2008). The Beta Workbench: a tool to study the dynamics of biological systems. *Brief. Bioinformatics* **9**, 437-449.

Deutscher, D., Meilijson, I., Schuster, S. and Ruppin, E. (2008). Can single knockouts accurately single out gene functions? *BMC Syst. Biol.* **2**, 50.

Fisher, J. and Henzinger, T. A. (2007). Executable cell biology. *Nat. Biotechnol.* **25**, 1239-1249.

Fisher, J., Piterman, N., Hubbard, E. J., Stern, M. J. and Harel, D. (2005). Computational insights into Caenorhabditis elegans vulval development. *Proc. Natl. Acad. Sci. USA* **102**, 1951-1956.

Fisher, J., Piterman, N., Hajnal, A. and Henzinger, T. A. (2007). Predictive modeling of signaling crosstalk during C. elegans vulval development. *PLoS Comput. Biol.* **3**, e92.

Gaffney, E., Heath, J. K. and Kwiatkowska, M. Z. (2008). A mass action model of a fibroblast growth factor signaling pathway and its simplification. *Bull. Math. Biol.* **70**, 2229-2263.

Gerhart, J. and Kirschner, M. (2007). The theory of facilitated variation. *Proc. Natl. Acad. Sci. USA* **104 Suppl. 1**, 8582-8589.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* **81**, 2340-2361.

Guerriero, M. L., Heath, J. K. and Priami, C. (2007). An automated translation from a narrative language for biological modeling into process algebra. In *Computational Methods in Systems Biology* (*Lecture Notes in Computer Science*, Vol. 4695) (ed. M. Calder and S. Gilmore), pp. 136-151. Berlin: Springer.

Guerriero, M. L., Dudka, A., Underhill-Day, N., Heath, J. K. and Priami, C. (2009). Narrative-based computational modelling of the Gp130/JAK/STAT signalling pathway. *BMC Syst. Biol.* **3**, 40.

Han, T., Katoen, J. P. and Mereacre, A. (2008). Approximate parameter synthesis for probabilistic time-bounded reachability. In *Proceedings of the Real-Time Systems Symposium* (RTSS), pp. 173-182. Piscataway, NJ: IEEE.

Hanafusa, H., Torii, S., Yasunaga, T., Matsumoto, K. and Nishida, E. (2004). Shp2, an SH2-containing protein-tyrosine phosphatase, positively regulates receptor tyrosine kinase signaling by dephosphorylating and inactivating the inhibitor Sprouty. *J. Biol. Chem.* **279**, 22992-22995.

Harel, D. (1987). Statecharts: a visual formalism for complex systems. *Sci. Comput. Program* **8**, 231-274.

Harel, D. (2005). A Turing-like test for biological modeling. *Nat. Biotechnol.* **23**, 495-496.

Heath, J., Kwiatkowska, M., Norman, G., Parker, D. and Tymchyshyn, O. (2007). Probabilistic model checking of complex biological pathways. *Theor. Comput. Sci.* **391**, 239-257.

Hillston, J. (1996). *A Compositional Approach to Performance Modelling*. Cambridge: Cambridge University Press.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A. et al. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19**, 524-531.

Hunt, C. A., Ropella, G. E., Park, S. and Engeleberg, J. (2008). Dichotomies between computational and mathematical models. *Nat. Biotechnol.* **26**, 737-738.

Kwiatkowska, M., Norman, G., Parker, D., Tymchyshyn, O., Heath, J. and Gaffney, E. (2006). Simulation and verification for computational modelling of signalling pathways. In *Proceedings of the Winter Simulation Symposium on Modelling and Simulation in Systems Biology*, pp. 1666-1674. Piscataway, NJ: IEEE.

Kwiatkowska, M., Norman, G. and Parker, D. (2007). Stochastic model checking. In *Formal Methods for Performance Evaluation: 7th International School on Formal Methods for the Design of Computer, Communication and Software Systems* (*SFM 2007 Advanced Lecture Notes in Computer Science*, Vol. 4486) (ed. M. Bernardo and J. Hillston), pp. 220-270. Berlin: Springer.

McGary, K. L., Lee, I. and Marcotte, E. M. (2007). Broad network-based predictability of Saccharomyces cerevisiae gene loss-of-function phenotypes. *Genome Biol.* **8**, R258.

Milner, R. (1999). *Communicating and Mobile Systems: The Pi-calculus*. Cambridge: Cambridge University Press.

Milner, R., Parrow, J. and Walker, D. (1992). A calculus of mobile processes, Pts 1 and 2. *Inf. Comput.* **100**, 1-40.

Pico, A. R., Kelder, T., van Iersel, M. P., Hanspers, K., Conklin, B. R. and Evelo, C. (2008). WikiPathways: pathway editing for the people. *PLoS Biol.* **6**, 1403-1407.

Priami, C. (2009). Algorithmic systems biology: an opportunity for computer science. *Commun. ACM* **52**, 80-88.

Priami, C. and Quaglia, P. (2005). Beta binders for biological interactions. In *Computational Methods in Systems Biology* (*Lecture Notes in Computer Science*, Vol. 3082) (ed. D. Vincent and S. Vincent), pp. 20-33. Berlin: Springer.

Priami, C., Regev, A., Shapiro, E. and Silverman, W. (2001). Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Inf. Process. Letters* **80**, 25-31.

Regev, A. and Shapiro, E. (2002). Cells as computation. *Nature* **419**, 343.

Regev, A., Silverman, W. and Shapiro, E. (2001). Representation and simulation of biochemical processes using the pi-calculus process algebra. *Pac. Symp. Biocomput.* **2001**, 459-470.

Regev, A., Panina, E. M., Silverman, W., Cardelli, L. and Shapiro, E. (2004). Bioambients: an abstraction for biological compartments. *Theor. Comput. Sci.* **2004**, 141-167.

Sandilands, E., Akbarzadeh, S., Vecchione, A., McEwan, D. G., Frame, M. C. and Heath, J. K. (2007). Src kinase modulates the activation, transport and signalling dynamics of fibroblast growth factor receptors. *EMBO Rep.* **8**, 1162-1169.

Sternberg, P. W. and Horvitz, H. R. (1989). The combined action of two intercellular signaling pathways specifies three cell fates during vulval induction in *C. elegans*. *Cell* **58**, 679-693.

Tymchyshyn, O. and Kwiatkowska, M. (2008). Combining intra- and inter-cellular dynamics to investigate intestinal homeostasis. In *Proceedings of First International Workshop in Formal Methods in Systems Biology*, pp. 63-76. Berlin: Springer.

Wang, Z. and Zhang, J. (2007). In search of the biological significance of modular structures in protein networks. *PLoS Comput. Biol.* **3**, 1011-1021.