

## Biological logic

An intuitive approach to computer modelling could reveal paths to discovery, finds **Lucas Laursen**.

**G**rabbing one of the three laptops in her office at Microsoft Research in Cambridge, UK, Jasmin Fisher flips open the lid and starts to describe how she and her collaborators used an approach from computer science to make a discovery in molecular biology.

Fisher glances across her desk to where her collaborator, Nir Piterman of Imperial College London, is watching restlessly. "I know you could do this faster," she says to Piterman, who is also her husband. "But you are a computer scientist and I am a biologist and we must be patient."

After a few moments, patience is rewarded: Fisher pulls up a screen of what looks like programming code. Pointing to a sequence of lines highlighted in red, she explains that it is a warning generated by software originally developed for finding flaws in microchip circuitry. In 2007, she, Piterman and their colleagues found a similar alert in a simulation they had devised for signalling pathways in the nematode worm *Caenorhabditis elegans*. Using that as a clue, they predicted and then experimentally verified the existence of a mutation that disrupts normal cell growth<sup>1</sup>.

'Executable biology', as Fisher calls what she's demonstrating, is an emerging approach to biological modelling that, its proponents say, could make simulations of cells and their

components easier for researchers to build, understand and verify experimentally.

The screen full of code doesn't look especially intuitive to a non-programmer. But Fisher toggles to another window that shows the same *C. elegans* simulation expressed graphically. It now looks much more like the schematic diagrams of cell-cell interactions and cellular pathways that biologists often sketch on white boards, in notebooks or even on cocktail napkins. One big goal of executable biology is to make model-building as easy as sketching. Fisher explains that each piece of biological knowledge pictured on the screen, such as the fact that the binding of one protein complex to another is necessary to activate a certain signal, corresponds to a programming statement on the first screen. Likewise, the diagram as a whole — illustrating, say, a regulatory pathway — corresponds to a sequence of statements that collectively function as a computer simulation. Ultimately, she says, this kind of software should develop to a point at which researchers can draw a hypothetical pathway or interaction on the screen in exactly the way they're already used to doing, and have the computer automatically convert their drawing into a working simulation. The results of that simulation would then show the researchers whether or not their hypothesis

**"Modelling in general is regarded sceptically by many biologists."**

— Stephen Oliver

REF. 7

corresponds to actual cell behaviour, and perhaps — as happened in the 2007 work — make predictions that suggest fruitful new experiments.

In the meantime, however, Fisher and her fellow executable-biology enthusiasts have a lot of convincing to do, says Stephen Oliver, a biologist at the University of Cambridge, UK. “Modelling in general is regarded sceptically by many biologists,” he points out.

### Born-again modeller

Fisher’s fascination with this type of modelling started in about 2000. She was studying for her PhD in neuroimmunology at the Weizmann Institute of Science in Rehovot, Israel, when she encountered David Harel, a computer scientist who was applying computational ideas to biology.

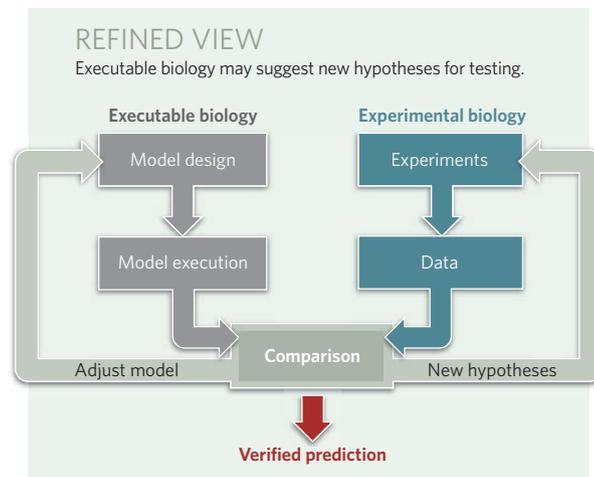
Harel wanted to get around the problems encountered in conventional simulations, which use reaction-rate equations and other tools of theoretical chemistry to describe, step by step, how reaction networks and cell interactions change over time. Such simulations can provide biologists with a gratifying level of detail for testing against reality. But the number of differential equations in these models escalates rapidly as more reactions are included, until they become a strain on even the most powerful computers. In one recent model of the networks involving epidermal growth factor, for example, 499 equations were required to describe 828 possible reactions<sup>2</sup>. Even if the computers can handle such a load, the output is often difficult to interpret.

Such models quickly become “an impossibly unwieldy black box”, says Vincent Danos, a computational biologist at the University of Edinburgh, UK. And if the models have such a hard time simulating the behaviour of a single set of signalling pathways, he adds, then it’s hard to imagine they will ever be of much use in systems biology, which might, for example, seek to understand all the pathways in a cell as an integrated whole.

Harel’s approach was to represent networks of biological events by a considerably smaller set of logical statements. For example, instead of specifying the number of signal molecules involved in a particular cell–cell interaction, or the sensitivity of the various receptors, a statement might simply say ‘when cell X is near cell Y for long enough, cell Y switches from one type of behaviour to another’. And, unlike the conventional equations, the rules tend to be independent of one another — an important part of why the simulations are so much easier to build.

An additional advantage of the logic-based approach was that standard model-checking algorithms — widely used by industry for testing computer hardware — could check whether the statements were logically consistent, and capable of producing the behaviour seen in cells. This analysis would highlight points in the model at which the behaviour was going awry, which in turn might suggest experiments to look for previously unsuspected reactions and molecular species at that point (see graphic).

Fisher became so caught up in the idea that in 2003 she joined Harel’s lab as a postdoc. She continued to work in the field during a three-year postdoc appointment under Thomas Henzinger at the computer-science department of



the Swiss Federal Institute in Lausanne (EPFL). Piterman, whom she had married in 1998, came to the EPFL as well, and the three of them collaborated with their colleague Alex Hajnal to build the *C. elegans* model.

They started by recording all the rules they could find in the literature pertaining to the maturation of a simple, well-studied system of six vulval precursor cells. “I wrote it all down first in a diagram,” says Fisher, pointing to a figure in a research article on her desk, “then we formalized all the arrows and feedback loops into the computer program.” Because the model needed only rules, not numbers, most of the information was qualitative (for example, this cell is closest to the cell sending the signal so the messenger molecules reach it first).

### Lab confirmation

The team knew that genetic mutations could nudge the cells into different roles during maturation, but they wanted to know more about the cascade of signals that dictate the fate of each cell. The model-checker explored the set of 48 mutations known to affect vulval development, which could have up to 92,000 possible outcomes. All but four of the perturbations predicted normal cell fates, so the team concentrated on simulating different timings of those four cases. They found two previously unknown effects. First, a set of inhibitory genes collectively known as *lst* genes have to be activated for vulval cells to convert to their ‘primary’ fate, meaning that their daughter cells will make up the vulval opening. Second, if another gene was disrupted and signals between the cells weren’t timed just in just the right sequence, the cell would adopt a different fate. A laboratory experiment confirmed both predictions.

“We used this qualitative model because we simply didn’t have the quantitative knowledge,” says Fisher. But now that the approach and its predictions have been verified in the lab, she says, “you can’t argue with it”.

Since then, Fisher has become one of the world’s most energetic proponents of executable biology<sup>3</sup>, but she is far from being the only enthusiast. In 2007, for example, biologist John Heath of the University of Birmingham, UK, was trying to model signal transduction pathways



Jasmin Fisher wants to be able to model complex cellular interactions.

and protein–protein interactions. “The processes are just really just too complicated to understand using intuition,” he says. He discussed his problem with University of Oxford computer scientist Marta Kwiatkowska, who was then working in the adjacent building at Birmingham, and she gave him a paper on model-checking. “I was reading the opening paragraph on the train and I thought, “This is exactly what I want,” says Heath. In collaboration with Corrado Priami, who leads the Centre for Computational and Systems Biology at the University of Trento in Italy, Heath was soon modelling the gp130/JAK/STAT signalling pathway<sup>4</sup>, a well-studied system involved in human fertility, neuronal repair and embryonic stem-cell renewal. Their model reproduced the dynamic behaviour of the pathway as observed in the laboratory, and has allowed them to make testable predictions about which parts of the pathway are most sensitive to mutation or other perturbation. Heath, like Fisher, is now actively promoting executable biology, and has joined with Kwiatkowska to publish a review paper on the approach<sup>5</sup>.

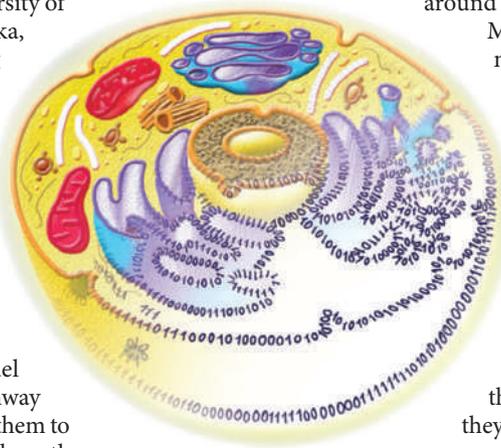
### Another level

Executable biology does have limitations, Fisher acknowledges. At present, for example, such models can handle only one level of narrowly defined biological activity at a time — the level of protein–protein interaction, say, or the level of cell–cell interaction. “We know there is feedback between the levels,” Fisher says, “but we don’t know enough about it” to get a computer to simulate that feedback.

An additional complication is that the different levels are best handled by different computer languages. To model the molecules that travel between cells, for instance, the most natural languages are those known in computer science as ‘process calculi’, which were devised to model information flow through communication webs. But to model the behaviour of an individual cell and its components, as in the various signalling and regulatory pathways, the most natural languages are those based on the theory of interacting ‘state machines’, which was developed to describe how objects transition from one state to another.

The long-term goal, says Fisher, is to develop more sophisticated and complete simulations that would help researchers explore a wider range of biological phenomena, both by integrating behaviour at the genetic, molecular and cellular levels, and by integrating executable models with more mathematical models. Indeed, as a group of bioengineers led by C. Anthony Hunt of the University of California, San Francisco, pointed out in a response<sup>6</sup> to Fisher and Henzinger’s 2007 review, it’s not an either–or choice between the executable biology and conventional mathematical modelling: both have their uses and limitations, depending on the level of biological activity being simulated.

Fully integrated modelling is still a long way off, admits Fisher. But now that executable-biology predictions have been verified in the lab, the field has begun to attract more attention. Labs worldwide are starting to use execut-



**“The model is not an oracle, it is an automation of your understanding.”**

— John Heath

able biology to study systems, and Fisher herself is giving invited lectures on the subject 15–18 times per year around the world.

Meanwhile, she and Piterman are trying to make the software more accessible to biologists, so that researchers can make executable-biology simulations a routine part of their work. Other research groups are working towards the same end. Priami’s group is trying to write interfaces so simple that biologists can fill in tables with their data, specify the rules they want to use in spatially organized diagrams and sit back while the program translates the data into a computer-readable language that can execute a simulation<sup>7</sup>. “We develop languages that allow people to program without knowing they are programming,” says Priami.

### Commercial efforts

In another effort to make the executable-biology approach more intuitive, Walter Fontana of the Harvard Medical School in Boston, Massachusetts, has joined with colleagues at the start-up firm Plectix to launch Cellucitate, an online visual interface for biological-pathway modelling that generates statements in an executable computer language called Kappa, which Fontana developed explicitly to model molecular interactions. Cellucitate — available for free during its trial period — allows collaborators to add information to a shared online model and revise it Wikipedia-style, something Fontana says is increasingly important because the empirical facts on which models are based are continually being revised.

Fisher hopes that the excitement will catch on in more groups and suggests that some of the computer-inspired ideas she is testing in her group’s latest *in vivo* experiments, which now extend to fruitflies and yeast cells, should entice more interest in executable biology among lab-based biologists.

But in the end, Fisher emphasizes, the fact that using executable rules could make the models easier to visualize is only an added bonus. Executable biology’s real pay-off is that it can help biologists to understand the complexity of living things, whether at the level of groups of molecules, such as Kappa describes, or at that of signals sent between cells, as in the nematodes Fisher herself studies. And that enhanced understanding, in turn, helps biologists ask new questions, design new experiments and make new discoveries. “But however good the models are, “you still need a good scientist to implement them”, says Kwiatkowska.

“The model is not an oracle,” Heath agrees, “It’s an automation of your understanding.” ■

**Lucas Laursen is a freelance journalist in Cambridge, UK.**

1. Fisher, J., Piterman, N., Hajnal, A. & Henzinger, T. A. *PLoS Comput. Biol.* **3**, e92 (2007).
2. Chen W. W. *et al. Mol. Syst. Biol.* **5**, 239 (2009).
3. Fisher, J. & Henzinger, T. A. *Nature Biotechnol.* **25**, 1239–1249 (2007).
4. Guerriero, M. L., Dudka, A., Underhill-Day, N., Heath, J. K. & Priami, C. *BMC Syst. Biol.* **3**, 40 (2009).
5. Kwiatkowska, M. Z. & Heath, J. K. *J. Cell Sci.* **122**, 2793–2800 (2009).
6. Hunt, C. A., Ropella, G. E. P., Park, S. & Engelberg, J. *Nature Biotechnol.* **26**, 737–738 (2008).
7. Priami, C. *Commun. ACM* **52**, 80–88 (2009).