1. Schmidt, C. *Nat. Biotechnol.* **26**, 145–146 (2008).
2. Council of Europe. *Additional Protocol to the Convention on Human Rights and Biomedicine, Concerning Genetic Testing for Health Purposes.* (Council of Europe, Brussels, Belgium, 2008). <http://conventions.coe.int/Treaty/EN/Treaties/Html/TestGen.htm>
3. Council of Europe. *Explanatory Report to the Additional Protocol to the Convention on Human Rights and Biomedicine, Concerning Genetic Testing for Health Purposes* (Council of Europe, Brussels, Belgium, 2008). <http://conventions.coe.int/Treaty/EN/Reports/Html/TestGen.htm>
4. <http://www.hgc.gov.uk>
5. <http://www.health.fgov.be/bioeth>
6. <http://www.ccne-ethique.fr>
7. <http://ec.europa.eu/european_group_ethics>
8. van Ommen, G.B. & Cornel, M.C. *Eur. J. Hum. Genet.* **16**, 403–404 (2008).
9. The European Parliament and the Council of the European Union. *Official J. Eur. Comm.* **L331**, 1–37 (1998).
10. <http://www.eurogentest.org/web/db/news/167/index.xhtml>
11. < http://www.itas.fzk.de/eng/etag/project.htm>

Dichotomies between computational and mathematical models

To the editor:

In a timely review in the November issue entitled "Executable cell biology," Jasmin Fisher and Thomas Henzinger[1] couple descriptions of new computational approaches for cell biology science with a vision of an emerging, new field. Although we are in full agreement with their basic message and vision, the review lacks clarity on several key points that may confuse skeptical readers, prospective new students and future practitioners of this emerging field. Below, we share several observations intended to strengthen the message and clarify the original vision outlined in the review.

On page 1240, Fisher and Henzinger state, "Whereas an algorithm must be devised to simulate a mathematical model, a computational model prescribes the steps taken by an abstract machine and is therefore inherently and immediately executable. As the primary semantics of computational models are operational, we use the term execution instead of simulation—hence executable biology." Technically, both mathematical and 'computational' models used in biomedical research prescribe steps to be taken by an abstract machine. Algorithms must be devised to execute either model type. Traditional mathematics, Statecharts and Reactive Modules are all specification languages in the same general category. They are formalisms. A model written using a formalism is translated into a computer program, which is then executed by a computer. That sequence is true for both mathematical and computational models.

Thus, by claiming mathematical and computational models are differentiated by their formalisms, Fisher and Henzinger draw a false dichotomy between the two model classes, one that some readers may see as a theme of this review. The important difference between the two model classes is not in the formalisms. It is in (i) the intent and approach used by the scientist, (ii) what the computer is doing when implementations of those models are executed and (iii) why the computer behaves as it does. Both model classes are usage oriented. When a model of the type described in Box 3 of the review has been applied to biology, the result has most often been an analog device or mechanism rendered in software. The traditional use of mathematical models in biology has been to describe data: when the implementation of a mathematical model is executed, values may be generated that closely match the original data. In contrast, during execution, the mechanisms of an analog device actually function in ways that are thought to mimic a biological mechanism at some level of abstraction.

Both model types must be translated from the formalism into machine code. The authors' rhetoric hinges on an implicit assumption: that the translation between computational models and machine code is more automated than the translation between mathematical models and machine code. The existence of Mathematica, Matlab and symbolic math tools is the evidence that this assumption is false.

In the next paragraph, the authors build on this false dichotomy. "Because computational models are qualitative, they do not presuppose a precision absent from the experimental data; because they are nondeterministic or stochastic, they allow many possible outcomes of a chain of events, which is often observed in biological systems." But in reality, computational models are just as quantitative as mathematical models. Part of the problem is their implicit assumption that 'quantitative' is identical to a continuum (or mathematical metric space). Except in the case of hybrid systems mathematics, it is true that mathematical models usually assume a continuum. In contrast, it is also true that computational models often operate in or on less well-formed data (e.g., measures as opposed to metrics and nonmetrizable sets as opposed to metric spaces). Nevertheless, digital computer software must still be crisply specified because when implemented it tolerates no ambiguity in its values; thus, computational models, when translated into machine code and executed, are just as quantitative as mathematical models.

Moving to Box 3 of the review, under "Boolean networks," we disagree with the statement that all computational models are stochastic or nondeterministic, as expressed by the authors. For example, chaotic models (deterministic and nonstochastic, yet unpredictable) are also possible using computational formalisms (e.g., those in Box 3). Of course, mathematical models can also be nondeterministic, stochastic or chaotic, but the task of making them so can become complicated.

In the section "Quantitative versus qualitative modeling of biology," the authors write, "A significant advantage of qualitative models is that different models can be used to describe the same system at different levels of detail and that the various levels can be related formally." By not pointing out that the same can be said of mathematical models, the authors imply that traditional mathematical models cannot be used to describe the same system at different levels of detail and/or that the various levels cannot be related formally. It would be incorrect to draw that inference, as mathematical models can be hierarchical, and a multi-level mathematical model can contain variables at one level that are mechanisms at another. Hence, by making their statement without an equivalent one about traditional mathematical models, the authors reinforce the false dichotomy.

The next section begins: "Computational models can be analyzed by model checking. Computational models can be used for

testing and comparing hypotheses. Suppose that we have collected experimental data. A computational model represents a hypothesis about the mechanism that results in the data." Again, the same can be true of mathematical models. An important distinction is that mathematical models formulate hypotheses about relations, variables and magnitudes, whereas the computational models that are the focus of the review formulate hypotheses about mechanisms or processes. However, as noted above, a multi-level mathematical model can also formulate a hypothesis about mechanisms or processes, and a relation at one level (or from one aspect) can be an object at another level (or from another aspect). The authors' statement thus further encourages the false dichotomy. The real dichotomy is one of usage, aspects or context, not formalism: how the model will be used, the aspects of the biological system's phenotype on which the model is focused or the context in which the model is executed. Traditional mathematical models are typically used to understand relations between variables, and most often focus on data. Computational models are used to understand interactions between the system components. For example, one can plug together autonomous software components anticipating that their interaction during execution may (or may not) mimic aspects of a biological mechanism.

Turning to Figure 1 and its legend, we agree with the implied message, but would add the following points to help make the message more precise and correct. Experimental biology requires model systems. Figure 1 makes false distinctions between computational models (including analytic and synthetic) and wet-lab, biological models, and thereby reinforces the false dichotomy. The implication of Figure 1 is that an analogy can be drawn between executable biology models and experimental biology. It is not an analogy. The methods are the same. The following five observations clarify their methodological equivalence and provide critically important, practical elements that are missing from Figure 1 and from the text.

First, one cannot directly compare an executing model to data. One must take measurements from the executing model just as one does during execution of a wet-lab, experimental model. The two sets of measures are then compared during validation. Second, model construction is not a unique attribute of computational modeling. *In vitro* wet-lab experimental systems must also be constructed. In fact, the *in vitro* experimental apparatus, the living, biological materials and the methods of observation within the laboratory context actually constitute the *in vitro* model; the system is constructed in the exact same sense as a computational model. Third, one must perform experiments using a computational model in the same sense that one performs experiments using a wet-lab, biological model. Fourth, model adjustment is not unique to computational models. One can adjust a biological model as well; examples include changing media composition, treatments or methods of observation. And finally, as a consequence of computational models being designed for experimentation, they can, like biological models, also suggest new experiments.

On page 1247, the authors state, "At the same time, a major challenge for biologists is to apply more formal approaches in biology and to develop precise, unambiguous and standardized representations of biological knowledge and data." This idea is reasonable only if we assume identification or a very close analogy between biology and machines (like a digital computer). However, biology is not like a machine and it is not a formalism. The very purpose of modeling is to abstract out some of the details of the system under study to identify likely principles of operation. Building a completely accurate model would be analogous to making the map as complicated as the territory it describes, which is useless. The map is not the territory. Models have a purpose and uses that are different from those of the systems to which they refer. At an abstract level, drawing an analogy between aspects of biology and precise, unambiguous machines has pedagogic utility. At the scientific level, such a tight analogy glosses over the important hypothesis that biology not only tolerates ambiguity, it actually seems to require it[2]. We speculate that increasingly useful computational analogs will be those that have been constructed using the more relaxed formalisms, such as partially ordered sets (of events), which are capable of managing ambiguity.

Later in the same paragraph the authors observe, "A main goal is to make computer science tools accessible to biologists. This requires, on one hand, an understanding by computer scientists of what kind and style of tools might prove useful to biologists and on the other hand, an understanding by biologists of what kind of assistance could be provided by computer science tools." The authors' implication seems to be that wet-lab biologists will redirect some of their research time to the conduct of computational experiments. However, as computational biological models (regardless of specification languages) become more complex, it becomes impractical for any single modeler to work in isolation. Thus, we envision a new cadre of scientist-engineers who specialize in computational biology and operate as members of interdisciplinary teams containing dedicated biologists and computer scientists, in much the same way that engineers and physicists have specialized in advanced, four-dimensional optical methods for visualizing and better understanding the dynamic details of biological systems[3].

In summary, we congratulate Fisher and Henzinger on highlighting central challenges facing the use of computational models in cell biology and agree with the basic message they put forth. Our criticism is offered in a good faith attempt to clarify the message and help ensure that it is well understood within the broader biotech community. We should strive to discuss computational and mathematical models with the same precision and accuracy used to design, build and validate them.

C Anthony Hunt[1,2], Glen E P Ropella[1,3], Sunwoo Park[1] & Jesse Engelberg[2]

[1]The Department of Bioengineering and Therapeutic Sciences, University of California, San Francisco, California 94143–0912, USA. [2]The UCSF/UCB Joint Graduate Group in Bioengineering, University of California, Byers Hall Room 216, San Francisco, California 94158–2330, USA. [3]Tempus Dictum, Inc., 27891 S.E. Starr Rd. Eagle Creek, Oregon 97214, USA. e-mail: a.hunt@ucsf.edu

1. Fisher, J. & Henzinger, T.A. *Nat. Biotechnol.* **25**, 1239–1249 (2007).
2. Kercel, S.W. *Chem. Biodivers.* **4**, 2369–2385 (2007).
3. Megason, S.G. & Fraser, S.E. *Cell* **130**, 784–795 (2007).

Fisher & Henzinger reply:

We appreciate the interest and support that Hunt *et al.* have shown for our attempt to draw attention to differences between mathematical and computational modeling in biology. Although we agree with many of their points, we counter that some of their statements need clarification.

They say "Technically, both mathematical and 'computational' models used in biomedical research prescribe steps to be taken by an abstract machine." This is incorrect. A mathematical equation does not describe an algorithm but a relationship between quantities. One can try to devise an algorithm to compute the quantities, but this may not be possible, because