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

there are fundamental boundaries to what computers can do, or it may be feasible only with limited precision. On the other hand, a computational model prescribes explicitly for every state the possible next states of the model. Although modern compilers are extremely complex and include many optimizations, we cannot compare them with programs like Matlab and Mathematica.

Undergraduate students can easily program a basic compiler from a high-level language to machine code. The translation is a straightforward change of each high-level step into several instructions of machine code. By contrast, Matlab and Mathematica include very sophisticated algorithms. They make powerful tools available to people that do not need to understand them, but to say that they resemble compilers that take a high-level programming language and produce machine code is taking it one step too far. Indeed, there is an entire sub-discipline of computer science, called the semantics of programming languages, which studies deep relationships and differences between mathematical (or 'denotational') and computational (or 'operational') models.

Hunt *et al.* go on to say that the real dichotomy between mathematical and computational models is one of usage rather than formalism. It is not a coincidence that mathematical models are used mainly for understanding relations between variables and computational models for understanding interactions between components. Every modeling approach has its strengths and drawbacks. This underlines our point that there is a real dichotomy between the two approaches. It is entirely correct that both mathematical and computational models are constructed using formal language, and that most models can be made nondeterministic, stochastic, hierarchical, etc. But it is also true, as Hunt *et al.* point out, that many such models would become too complex to be developed and analyzed. The question is not what a modeling framework can do in principle, but what it is best at: for which situations is it an intuitive fit, and for which problems does it offer strong analytical and computational support.

As to their statement, "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 objects), which are capable of managing ambiguity," we completely agree. This is exactly why we have emphasized that a major strength of computational modeling is nondeterminism. This does not mean that biology does not need to be more formal. The way of communicating results of experiments and mechanistic ideas needs to become more formal. The development of better models requires the collaboration of many laboratories and requires a precision in the communication that cannot be afforded without formality. This does not mean that ambiguity has no place in biology. It means that we need to be formal about ambiguity.

HLA-haplotype banking and iPS cells

To the editor:

The production of induced pluripotent stem (iPS) cell lines from human somatic cells, such as skin fibroblasts, recently reported in your January issue[1] and elsewhere[2–4], opens the exciting new possibility of producing personalized pluripotent stem cell lines from individual patients without using human oocytes or embryos. In addition to circumventing ethical problems associated with human embryonic stem (ES) cells, this new approach could also address immunological rejection associated with cell therapies because iPS cell lines could be created with human leukocyte antigen (HLA)-haplotypes matching those of individual patients. Although such personalized stem cells are possible, the time and cost necessary for the production of clinical-grade iPS cell lines, the production of differentiated cell types for transplantation and the safety validation of all these procedures (including the final cell products for therapy) could limit wide adoption in clinical practice.

An alternative approach for reducing or avoiding immunological rejection of cell therapies would be to construct an HLA-haplotype bank of pluripotent stem cell lines, the possibility of which was first proposed for the UK population[5]. In a previous study, we calculated the appropriate size of human ES cell lines necessary for providing the majority of the Japanese population with beneficial HLA-matching[6]. This showed that 200 ES cell lines derived from randomly donated embryos can provide beneficial matching for 80% of people with at least two-locus matching for HLA-A, HLA-B and HLA-DR. Moreover, 100 parthenogenetic ES cell lines from randomly donated oocytes, can be matched to three locuses in 90% of the population because such ES cell lines are homozygotes for all loci. In this context, the recent production of parthenogenetic human ES cell lines is of note[7], although clinical application would likely be compromised by abnormal (maternal) genomic imprinting across all loci.

If iPS cell lines could be derived from donors' somatic cells with an efficiency and safety profile that was comparable to ES cell lines, an opportunity exists to establish more efficient HLA-haplotyped iPS cell banks. Under such a scheme, individuals with certain HLA-haplotypes could be chosen for production of iPS cell lines, rather than depending on randomly donated embryos or oocytes. It would be particularly useful if somatic cells, such as a small piece of skin, used to derive iPS cells, could be obtained from HLA-homozygous individuals. Two questions directly pertain to the feasibility of such an approach: first, how many such cell lines would be needed for HLA matching to the majority of iPS cell recipients, and second, how easy would it be to find homozygote donors?

To answer these questions, we estimated the frequencies of HLA homozygotes in the Japanese population from the observed frequencies of HLA three-locus haplotypes (A-B-DR) at broad serology-level resolution, based on the same data used in our previous study[5]. This data set comprises HLA-A, HLA-B, and HLA-DR types from a total of 2,578 unrelated individuals, including unrelated