

A call for virtual experiments: accelerating the scientific process

Jonathan Cooper, Department of Computer Science, University of Oxford, Parks Road, Oxford, OX1 3QD, UK

Jon Olav Vik, Department of Animal and Aquacultural Sciences, Centre for Integrative Genetics, Norwegian University of Life Sciences, P.O. Box 5003, N-1432 Ås, Norway

Dagmar Waltemath, Department of Systems Biology and Bioinformatics, University of Rostock, D-18051 Rostock, Germany

All authors contributed equally to this work.

Introduction

"Experiment [noun]: An action or operation undertaken in order to discover something unknown, to test a hypothesis, or establish or illustrate some known truth" (Oxford English Dictionary online [1])

Experimentation is fundamental to the scientific method, whether for exploration, description or explanation [2,3]. In the exploration of a novel system, children and researchers alike will mess about with things just to see what happens. More formalized experimental protocols ensure reproducible results and form a basis for comparing systems in terms of their response to a specific stimulus. Finally, experiments can be carefully designed to distinguish between competing causal hypotheses based on their different testable predictions about the outcome of the experimental manipulation. One would therefore expect experiments to be central in computational biology too.

Indeed, a mathematical model embodies a thought experiment, a causal hypothesis, and its falsifiable predictions. It is easy to ask what if we were to change a parameter, an initial state, or the model structure [4]. Papers in computational biology focus on describing and analyzing the effects of such changes, and on confronting models with experimental data [5]. This confrontation often generates new hypotheses, and many if not most new models arise by modification of existing ones [6,7]. However, most virtual experiments are not built to be reproducible [8], and thus die with the paper they are published in. This inhibits the critical scrutiny of models, as models are seldom subjected to the same simulation experiments as their predecessors, or revisited later in the light of new data. Perhaps worse, the status quo fails to take full advantage of experiments as a common language between modellers and experimentalists. This limits the relevance of mathematical models for experimental biologists, who often prefer to rely primarily on mental models to develop hypotheses and design tests for them. Despite the growing availability of data and model repositories [9–12], there has been only a slow uptake of emerging tools and standards for documenting and sharing the protocols for simulation experiments and their results [8,13,14].

We define a *virtual experiment* as the *in silico* analogue to a wet lab or field experiment, performed on a computational model rather than the real system or a physical model (see Box 1 for definitions). Here we argue that promoting the reuse of virtual experiments would vastly improve the usefulness and relevance of computational models, including in biomedical endeavours such as the Virtual Physiological Human [15,16] and the Human Brain Project [17]. We review the benefits of reusable virtual experiments: in specifying, assaying, and comparing the behavioural repertoires of models; as prerequisites for reproducible research; to guide model reuse and composition; and for quality assurance in the application of computational biology models. Next, we discuss potential approaches for implementing virtual experiments, arguing that models and experimental protocols should be represented separately, but annotated so as to facilitate the linking of models to experiments and data (Figure 1). We follow with some consideration of open questions and challenges that remain before the use of virtual experiments can become widespread. Lastly, we outline a vision for how the rigorous, streamlined confrontation between experimental datasets and candidate models would enable a "continuous integration" of biological knowledge, akin to the strategy used in software development [18].

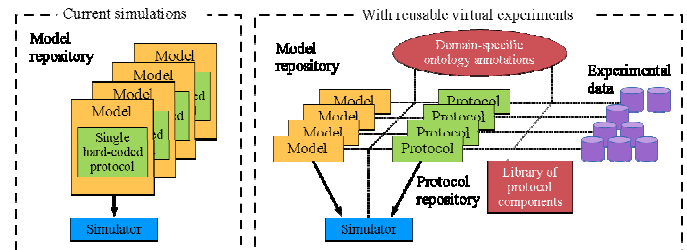


Figure 1. Integrating virtual experiments in the scientific workflow. At present (left), model repositories hold machine-readable model descriptions, typically containing a single hard-coded protocol. To perform any other virtual experiment on a given model requires substantial manual intervention. A more flexible architecture for virtual experiments (right) would define protocols independently of the model, but require models to be appropriately annotated with unambiguous identifiers for variables and parameters, so that protocols can be formulated in terms of such identifiers. This makes it possible to perform virtual experiments on all models having the annotations used by the protocol, or to find all experiments that are applicable to a given model. Similarly, annotation enables the automatic confrontation of simulations with relevant experimental data.

Box 1: Terminology

A *model* is a purposeful simplification of reality, designed to imitate certain phenomena or characteristics of a system while downplaying non-essential aspects. Its value lies in the ability to generalise insights from the model to a broader class of related systems. Thus, a lab mouse can be a model representing mammals in general; an *in vitro* heart cell can represent the cells in an intact heart; and a set of differential equations can approximate the dynamic behaviour of a biological system.

For mathematical and living biological models alike, an *experiment* is the process of inducing changes or stimuli to elicit some response from the system that can be observed and carries information about the inner workings and/or emergent properties of the system.

An experimental *protocol* is a detailed specification for carrying out an experiment. Whether involving a wet-lab, field or simulated experiment, this will include interventions, recordings, and post-processing. A protocol for a wet-lab experiment will specify environmental conditions, whereas a simulation protocol will translate these into corresponding initial/boundary conditions and parameter values. A simulation protocol may also include details of the numerical algorithm and parameters to use.

A *phenotype* is any observable trait of interest in an organism or a model thereof. The purpose of computational physiology is to mimic measurable phenotypes based on mechanistic descriptions of dynamical systems.

Ontologies are domain-specific lists of concepts and the relations between them. Multiple ontologies can be combined to encode biological knowledge, so that labels can be given a precise technical meaning. For example, consider the phenotype "mass of a heart cell". This is the *quality* "mass", pertaining to a heart cell, which *is a* cell and *is located in* the heart, which *is an* organ. A key feature of formal ontologies is that they are computer-processable, and automated tools can make logical deductions from the relationships stated.

Semantic interoperability (where "semantic" means "relating to meaning") denotes the ability to consistently navigate and query a set of data, model and protocol resources using terms taken from one or more ontologies [43]. With *ontological annotation*, new knowledge automatically connects to that which already exists, so that users can discover relevant knowledge without knowing its location in advance, and without having to formulate specific queries to link and select data.

As a running example we will refer to heart cell modelling, a mature research field [19] that relies heavily on experimental manipulation such as electrical pacing and cellular patch clamping (Box 2).

Box 2: Example from cardiac electrophysiology

As an example of the central role virtual experiments could play, we consider cardiac electrophysiology modelling. Here the use of mathematical models dates back over 50 years [46], making this field a relatively mature area of computational systems biology. Several generations of computational models of individual cardiac cells of varying types and complexity have now been developed, and more than 100 are available within the CellML Model Repository [38]. Such models take the form of (often large) systems of coupled nonlinear ordinary differential equations modelling processes such as the influx and efflux of ions across the cell membrane in response to, and generating, changes in the transmembrane voltage.

Except for the pacemaker cells and some others, isolated mature heart muscle cells are quiescent. To elicit behaviour typical of a cell in its normal environment we need to mimic the electrical impulses it normally receives from neighbouring cells. This involves injecting a brief stimulus current, which depolarizes the cell and allows influx of sodium ions and further sustained depolarization, known as the *action potential* (see Figure 2A). This sets off a chain of events that releases calcium into the cytosol, where it binds to regulators of motor proteins allowing the cell to contract. Where a wet-lab experimentalist will administer a stimulus current via an electrode, the computational biologist might build in a stimulus current term in the differential equation for transmembrane voltage.

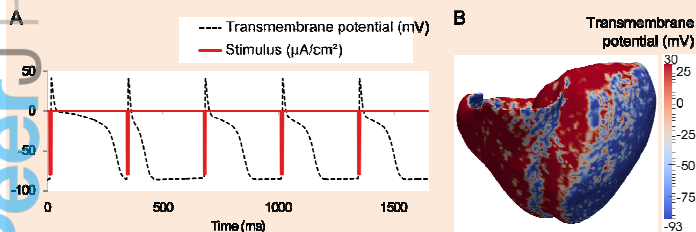


Figure 2. Examples of cardiac virtual experiments. A) Simulated stimulation of a heart cell, eliciting an action potential (approximate reproduction of Figure 12 in [59] using a CellML encoding of the model equations). B) Transmembrane potential in a human ventricular mesh after a defibrillation shock (based on [60], using the cell model in [41], simulated in Chaste 3.1 [61]).

Most of the heart cell models in the CellML repository implement a single experiment, namely regular pacing. The virtual counterpart of a wet-lab experiment then amounts to changing parameter values for the stimulus duration, amplitude, and frequency of repetition; or possibly modifying the initial state of the simulated cell. Regular pacing is informative in many ways, and can reveal shortcomings such as the long-term instability that results from failing to conserve charge (a common omission in early models). However, many other experiments are possible, each focusing on some particular aspect of heart cell dynamics. These may involve changing a parameter value, the initial state, or a structural aspect of the system. For example, the conditional knocking-out of ion channel genes can be simulated by setting the corresponding ion-channel conductance parameters to zero.

Many experiments cannot be represented by simple parameter changes, but instead require modification of the mathematical model structure. A case in point is voltage clamping [47], an ingenious technique for dissecting the behaviour of voltage-dependent ion channels. By using an amplifier, current is injected to forcibly apply, and hold, any specified voltage across the cell membrane. In model terms, this amounts to replacing a differential equation for voltage with a constant parameter. The resulting ionic current gives information about the opening and state-switching of ion channels. A wide variety of voltage clamp protocols have been designed to emphasize particular aspects of different ion channel kinetics.

In current practice, changes to models are mostly represented by duplicating the entire model and then modifying its source code. In cardiac electrophysiology this is seen most commonly in the provision of endocardial, epicardial and midmyocardial variants of many ventricular cell models (e.g. http://models.cellml.org/workspace/tentusscher_panfilov_2006 for [41]). This fragmentation of model families is equally widespread in the BioModels Database [12], where several models of cortical spiking neurons, all based on Izhikevich 2004 [48], differ primarily in parameter values (for instance BioModels IDs BIOMD0000000129 to BIOMD0000000136).

Increasingly, researchers are also using single cell models within integrated, multi-scale models of whole heart (or at least whole ventricle) function (see Figure 2B). Software platforms exist that allow automated incorporation of multiple cell models (e.g. myocardial cells from differing regions of the heart, Purkinje fibre cells) direct from the CellML repository into the multi-scale computational model (see [37] for a comparison).

It is therefore a hindrance to the field that most curated models come with some specific experimental protocol hardcoded into the equations. To reuse or combine such models, you must "unpick" the experimental protocol so that only the model structure remains. This is frequently a complex and error-prone procedure, as discussed in [49].

The virtues of virtual experiments

Descriptions of the behavioural repertoire of models

One does not model a *system* so much as a set of *phenomena*. Insofar as a model is a *purposeful* simplification, what should be included or left out depends on what *behaviours* it is supposed to imitate. Furthermore, any useful model must be capable of *not* exhibiting the phenomenon if certain parameter values, initial states, or model structure were different. This is what makes the model causal: it is a statement about sufficient causes to exhibit the phenomenon in question. Such what-if questions are all examples of virtual experiments.

Indeed, many phenomena are *created* by experiments, under conditions so artificial as not to occur in nature, as asserted by Ian Hacking in his classic *Representing and intervening* [2]. Likewise, many phenotypes are defined by a system's response to some stimulus or perturbation. Francis Bacon, four hundred years ago, likened this to "twisting the lion's tail" [20]; a more modern example is the action potential of isolated, excitable cells, which is evoked by an electric stimulus (Box 2). In either case, experiments bring into play mechanisms whose importance may not be apparent under passive observation.

The behavioural phenotype of a dynamical system is a high-dimensional and complex thing [21]. Even a "simple" measure such as the duration of an action potential is a summary of the time-course of transmembrane voltage, which is but one of the myriad variables in a heart cell system. A rich characterization of the phenotype aids in mechanistic interpretation, in constraining parameter estimation, and in exposing models to empirical challenge. For example, the combination of calcium transient and action potential data has been shown to identify more parameter values than action potentials alone [22,23].

Virtual experiments serve as assays of a model's behavioural repertoire, both in declaring what a model should do and verifying what it actually does. For example, the Bondarenko heart cell model [24] is feature-rich and was designed to accommodate various pacing protocols and a suite of voltage-clamp protocols for different ion currents. On the other hand, the Bernus model [25] was simplified for computational efficiency and stability, while using virtual experiments to ensure that the model still exhibited the various intended behaviours.

Behavioural assays using virtual experiments form a relevant basis for *comparison* within a class of systems. Each species, cell type, or candidate model can be positioned relative to others along phenotypic axes, with the comparison focused on the phenomena of interest. For example, ten Tusscher and co-workers [26] compared heart models

based on action potential morphology, conduction velocity, spiral wave dynamics and other features. By making it easy to apply published experiments to published models, the scientific community can place each new study in the context of all previous ones, greatly enhancing their combined value. To realize these benefits, virtual experiments need to be easily reusable with different candidate models.

Reproducible research

Scientific knowledge must be independently verifiable, even in computational sciences [27], and a growing number of scientists are gathering under the banner of *reproducible research* [8,13,28,29]. The spectrum of reproducibility [28] ranges from strict "replication" as in re-running a simulation in exact detail, for instance using a virtual machine [30], through "reproduction" in the sense of independent re-implementation of the essential aspects of a carefully described experiment [31,32], to "constructive replication" [33] based only on a clear statement of the hypothesis, leaving the next researcher free to choose their own experiment and analysis methods. Depending on the implementation approach, virtual experiments can contribute at different points along this spectrum.

At the replication end, a virtual experiment can be represented as a script or program, and rerun exactly. Publications could point at particular combinations of models and associated protocols to show how a particular figure was produced or how an analysis was carried out, providing "verifiable computational results" [34]. The Simulation Experiment Description Markup Language (SED-ML) [8] is a format to store simulation setups for replication of a particular outcome.

More usefully than a script, a protocol description can be given at a higher level of abstraction, providing the details requested by reporting guidelines such as the *Minimum Information About a Simulation Experiment* (MIASE) [13]. Different tools that understand the protocol description can reproduce the experiments using their own implementations of algorithms etc., and using their own set of models.

Finally, virtual experiments even support going beyond just reproducing an experiment to extending it. Aspects of the experiment description can easily be tweaked, or parameters adjusted, in order to approach the same essential hypothesis from slightly different angles, or investigate the wider behavioural regime.

Behavioural specifications for quality assurance, model reuse and model composition

To enter the clinical mainstream, computational biology must achieve industry-level quality management, including *verification* (how accurately software implements an underlying model), *validation* (how accurately the model represents reality) and *uncertainty quantification*, collectively abbreviated VVUQ [35,36]. These questions can only properly be considered in terms of particular phenotypes, or *quantities of interest* in VVUQ jargon. For example, a heart model may accurately predict activation time [37] but provide poor approximations to other phenotypes. A virtual experiment defines quantities of interest and thus plays a key role in this endeavour.

Cooper et al. [14] argued that effective reuse of published models is impossible without knowing the behaviours a model can produce. They suggested as a solution "functional curation" of models, i.e. automated confrontation between data and models based on (real and virtual) experiments. This would complement the existing curation of model repositories [12,38] that ensures that a model contains valid mathematics, that units are consistent on both sides of an equation, and so on. With models, data, and virtual experiments available as distinct entities from shared repositories in standard formats, newly published models or datasets can be picked up by the system and analysed automatically under *all* suitable virtual experiments. The results would be checked against the expected behaviours, and thus the region of operation for a model would be determined, and limitations for the use of that model identified. Nevertheless, there are many subtleties in performing a rigorous comparison of simulation outputs with experimental data, including the implications of different experimental techniques and the range of quantities that can determine suitability (e.g. temperature, species, gender, pH, osmolarity) [39].

Comparing models against behaviour specifications could also be tightly integrated into model development. This would allow model developers to ensure that in specialising a model for a particular experiment, "standard" behaviour is not unexpectedly lost. For parameter estimation the difference between a virtual experiment's output and experimental data can be used as the objective function. With more virtual experiments applicable to a given model, many parameters may potentially be fitted simultaneously.

Model composition is crucial for integrative research programmes [40] such as the Virtual Physiological Human and the Human Brain Project. By clearly specifying model requirements in terms of expected behaviours under standardised experiments, we envision that model composition could be made much more straightforward, focused and reliable. Functional curation would help both in screening for candidate submodels, checking whether submodels remain valid if computationally simplified [41], and for assessing the performance of the compound model. The relevant consequences of any modifications or computational shortcuts would be immediately apparent, enabling a test-driven development cycle similar to that used in software engineering.

How to realize virtual experiments

Two key design principles can help realize the benefits described above. Firstly, a clear separation of the experimental protocol from the model makes it easier to perform new experiments with a model, or run an existing experiment with new models. Secondly, data, model and protocol resources need to be labelled in a common language that facilitates automated processing. Although both the benefits and the principles are generic across disciplines, the way forward is clearer for research fields that have already established some level of standardization, modularity and abstraction. As an illustration from heart modelling, we outline how the existing groundwork (databases, model repositories, markup languages and open standards to describe biological knowledge) can be adapted to foster the reuse and sharing of virtual experiments, and to streamline the confrontation of models with data.

As discussed in Box 2, the main problem with today's models is that most hardcode a single experiment, with the model and protocol interwoven in one piece of "model code" (Figure 1, left). Applying new experiments to such a model requires a lot of reverse-engineering, tweaking and technical knowledge that is beside the conceptual essence of the model. What would be better is to build one model, then represent its different behavioural aspects in terms of different virtual experiments performed on that model.

Separate experiment and model descriptions

At the simplest level, the separation of experiment and model involves restructuring the software that implements experiments on models. This may involve "unpicking" an existing, "hardwired" experimental protocol from the model code, so that only the model structure remains. The key requirement is to identify clear interfaces between the experimental and model components. Doing so also clarifies what the experiments performed actually are, making them easier to modify for different purposes.

Such a separation is a goal of the Simulation Experiment Description Markup Language (SED-ML) [8]. An experiment encoded in SED-ML may change parameters and initial states, collect specific outputs for post-processing, and specify technical options e.g. for numerical solvers. However, it still refers to hardcoded parameter names and model identifiers, which precludes its use with other, similar models, or even with an updated version of the same model. Thus, while SED-ML ensures *replicability*, it does not yet offer *constructive replication* in the sense of performing the virtual experiment on any model that encodes a particular feature of interest.

There are trade-offs between implementation options for virtual experiments with regard to ease of construction, scope for automation, and generality. General-purpose programming languages have the advantage of being human-friendly, expressive and flexible. In contrast,

markup languages such as SED-ML are verbose and complex, but designed for machine processing of the encoded biological meaning. They are thus easier to work with in an automated fashion.

Depending on the level of detail, abstraction and generality of the model, the virtual counterpart of a wet-lab procedure may involve changing a parameter, an initial state, or a structural aspect of the system. The different scenarios of parameters and initial state can be viewed as members of a family of systems that the model describes in a unified manner. Thus, it is straightforward to represent and perhaps automate those experiments that correspond just to changing a parameter or initial state. A stimulus period parameter is easy to change; the internal details of a hardcoded cell model are not.

To perform experiments that are not built directly into a model, it is often necessary to modify the model structure. While this can be done manually for quick prototyping, applying the same modification to another model will be easier and more reliable if it can be done programmatically. For instance, a differential equation for transmembrane voltage might be replaced with a constant to represent voltage clamping. Such modifications are easier with domain-specific languages and a clean separation of model and experiment. Then, only the key features of the model and experiment are encoded, divorced from "bookkeeping" details required by a programming language or software framework.

Provide semantically rich interfaces between models and experiments (and researchers)

An experimental protocol needs a way to recognize the model components, parameters and variables it manipulates. The pacing of a heart cell should be applicable to any model that includes an input current and a transmembrane potential, regardless of whether the latter is called V or V_m . To achieve this, connections between protocol and model descriptions need to be done at the level of *biological concepts*, rather than depending on details of a particular encoding [42]. What is needed is a common language between models and protocols (and researchers), and fortunately this already exists [43] through the use of *ontological annotation* (see Box 1). Such an unambiguous identification of every variable, parameter or process in a model, data set or experimental procedure is already in place for many models in the BioModels database [12]. Models are free to use their own variable names if in addition they use globally agreed identifiers [44]. The ability to consistently navigate and query a set of data and model resources using terms taken from one or more ontologies is called *semantic interoperability* (Box 1). Together, ontologies, standard formats, and associated infrastructure enable automated *reasoning*: the translation of human-meaningful queries to machine-processing of the relevant parts of vast data and model resources [43]. One such application could be a protocol that clamps all intra- and extra-cellular ion concentrations, without needing to know which ion species are included in each model. Another is to compute the sum of all membrane currents involving potassium ions, as new potassium currents have been discovered and added to models over the years.

For virtual experiments, the key benefit of ontological annotation is that it defines *interfaces*. What features (components, variables, parameters) must a model contain to be eligible for a given experiment? What variables, collected under what circumstances, must a dataset contain to be useful in validating a given behavioural aspect of a model? This has applications such as determining which experiments may usefully be applied to which models, ensuring that the interfaces are compatible: the model provides the biological concepts which the experiment is probing. Model databases that link models, ontologies and associated simulations already enable queries such as "Which simulation experiments study the change of concentration in 'm-phase inducer phosphatase'?" [45], and could easily integrate virtual experiments. Similarly, queries over experimental databases can help to discover data available for calibrating or validating a model. These allow the many-to-many linking of models to experimental data via virtual experiments (Figure 1, right).

Challenges, and visions of a future

We have argued that virtual experiments have the potential to accelerate the scientific process of computational modelling, and identified some key principles for implementing them effectively. In the following, we identify some of the challenges that impede the way to a future where virtual experiments are routinely incorporated in computational research.

The power of virtual experiments extends to all fields of modelling, not just those from which we have drawn our examples above (see Box 3). However, the concepts required to represent virtual experiments vary widely across domains, and current solutions only capture a small subset. These differences in requirements depend partly on the maturity and homogeneity of modelling work. Where high flexibility is needed, the power of a full general purpose programming language may well be required in order to encode at least some aspects of experiments. Where there is greater consistency in the kinds of experiments performed, there is much more scope for a community standard with restricted semantics but greater stability.

Box 3: Expanding the scope of virtual experiments

The principal strategies to realize virtual experiments, namely ontological annotation and standardized interfaces, are generic and applicable even in research fields that are less standardized than e.g. ordinary differential equation modelling of cell biology. Many experiments, particularly for biomedical applications, involve spatial variation and/or composed multi-scale multi-physics models. In such scenarios one often wishes to modify model parameters within a particular spatial subdomain, or probe particular component models; these needs increase the complexity of the model/protocol interface [42].

One specific example is agent-based modelling, which is an eclectic approach with applications in ecology [50], economics [51] and medicine [52]. Recent years have seen the gradual adoption of guidelines for model documentation and reproducibility of such models in ecology [53,54]. It would be straightforward to extend the current structured-text-based scheme with ontological annotation. On the other hand, it is harder to standardise model implementations, which are very ad hoc in most agent-based models [50]. However, certain frameworks such as NetLogo [55] are widely used for models of medium complexity. In general, separating experiment from model becomes a programming issue, in that all relevant experimental parameters must be user-controllable and discoverable through annotation.

Virtual experiments could also become applicable in the wet lab. For instance, patch clamping of muscle, nerve or brain cells is a prime candidate for tight integration of wet-lab experiment and computer modelling. The robotized, high-throughput systems available [56–58] already have their own formats for specifying voltage-stepping protocols, and excitable cells are among the best-modelled biological systems in public repositories (Box 2). It is a small step to use the same protocol specifications for wet-lab and virtual experiments, and thus streamline the confrontation of simulated and experimental data.

Awareness of the potential benefits of standardized and reusable virtual experiments is important, but not sufficient. Experience from other standardization efforts has shown that ideas will only be taken up when there is tool support. For example, most published SBML models arise only from modellers working with tools that directly export SBML. Encouragingly, a few model repositories have started to provide simulation descriptions in SED-ML format as a reproducible proof of their curation figures. However, whether or not a SED-ML file is available for a model so far depends on the curator, as very few researchers provide their experiments in a standard format. High-quality virtual experiments will only be provided if they can easily be generated and explored by software tools.

Together, such tools could enable a continuous integration of knowledge. We envision a world where, as novel data are recorded and hypotheses are generated, they can be incorporated into existing or

extended models *in the context of existing knowledge*, rather than in an ad-hoc manner as is often currently the case. Once in the system, each piece of information added provides new context for the existing ones, and thus the invested efforts bring rewards that continue to grow cumulatively. The high-throughput screening of data versus models will rapidly identify gaps in understanding, limitations in the data, and potential contradictions in current hypotheses. In concert with systems for difference detection between models (e.g. [6]), one could identify the specific model change(s) leading to differences in behaviour. By integrating functional curation as a public service, both for model repositories and model developers, researchers may easily characterise or benchmark their models under a wide range of scenarios (including easily varying published experiments to address their own specific setup), while checking whether standard behaviours are retained. We believe this will lead to faster development of better models.

These visions have exciting implications for new research paradigms, with applications across biomedical research, particularly for embedding modelling into clinical use (somewhat fancifully imagined in Box 4). However, many non-trivial challenges remain to be tackled, by experimentalists, by modellers, by developers of tools and standards, and by data and model curators. We hope our paper will stimulate our readers to be involved.

Box 4: A brave new world

Contributed by Gary Mirams.

Vladimir sat down at his desk, and glanced at the computer clock, 09:00 Thursday 20th February, 2035. His task for the day was to make a new <ion channel> model, because a novel methylation state was discovered on Tuesday that forms in <very specific cell type and disease circumstances>.

Experiments had been performed on Wednesday, guided by protocols used to fit the previous model of <ion channel> for <very specific cell type>. The experimental team had looked up the existing model of <ion channel> for healthy <very specific cell type> and had access to the protocols that were needed to fully describe the kinetics of this channel, at a range of temperatures, pHs, ion concentrations, etc. They downloaded the protocols, ran their wet-lab experiments using the PatchClamper7000+, and the results had been uploaded to the WHO central database that evening, along with a computer-readable description of the corresponding protocols, conditions and all meta-data.

Vladimir downloaded all this information by 09:10 and was immediately able to simulate the predictions of the existing model using the machine readable protocol. He had a quick glance at the difference between the existing model's predictions and the new recordings; this novel methylation state made quite a difference! Together with SuperFancyFitting algorithms and full access to the raw data he selected a range of sensible objective functions, and re-fitted the previous model's parameter co-variance distributions to describe the new methylation state; unfortunately it couldn't describe the data whatever re-fitting he tried. Luckily NovelSuperFancyFitting algorithms had been invented in 2034 that suggested a range of possible new kinetic states and transitions, and he was able to select the minimal one for a great fit and evaluated its predictive power by 11:00. The community had standard guidelines for the adoption of novel models, and he had run the automated checks by 11:30: it passed and was deemed of sufficient predictive value to be widely useful. By lunchtime, the virtual physiological human model (in use in every hospital in the world) had been upgraded to version 10.4.1.3 and was providing more accurate predictions of <arrhythmic risk, pain relief, exercise guide, diet supplements> for physicians looking after <very specific disease circumstances> patients.

Acknowledgements

We thank Arne B. Gjuvslund and Gary Mirams for valuable discussion and comments on this manuscript, and Gary for the tale in Box 4.

References

1. experiment, n. (n.d.). OED Online. Available: <http://www.oed.com/view/Entry/66530>. Accessed 5 November 2013.
2. Hacking I (1983) Representing and intervening: introductory topics in the philosophy of natural science. Cambridge; New York: Cambridge University Press.
3. Radder H (2003) The philosophy of scientific experimentation. Pittsburgh, Pa.: University of Pittsburgh Press.
4. Morgan SL, Winship C (2007) Counterfactuals and causal inference: methods and principles for social research. New York: Cambridge University Press.
5. Hilborn R (1997) The ecological detective: confronting models with data. Princeton, N.J.: Princeton University Press.
6. Waltemath D, Henkel R, Hälke R, Scharm M, Wolkenhauer O (2013) Improving the reuse of computational models through version control. *Bioinformatics*. Available: <http://bioinformatics.oxfordjournals.org/content/early/2013/01/18/bioinformatics.btt018>. Accessed 24 January 2013.
7. Smith NP, Crampin EJ, Niederer SA, Bassingthwaite JB, Beard DA (2007) Computational biology of cardiac myocytes: proposed standards for the physiome. *J Exp Biol* 210: 1576–1583. doi:10.1242/jeb.000133.
8. Waltemath D, Adams R, Bergmann FT, Hucka M, Kolpakov F, et al. (2011) Reproducible computational biology experiments with SED-ML - The Simulation Experiment Description Markup Language. *BMC Syst Biol* 5: 198. doi:10.1186/1752-0509-5-198.
9. Joyce AR, Palsson BØ (2006) The model organism as a system: integrating “omics” data sets. *Nat Rev Mol Cell Biol* 7: 198–210. doi:10.1038/nrm1857.
10. Le Novère N, Bornstein B, Broicher A, Courtot M, Donizelli M, et al. (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucl Acids Res* 34: D689–691. doi:10.1093/nar/gkj092.
11. Lloyd CM, Lawson JR, Hunter PJ, Nielsen PF (2008) The CellML Model Repository. *Bioinformatics* 24: 2122–2123. doi:10.1093/bioinformatics/btn390.
12. Li C, Donizelli M, Rodriguez N, Dharuri H, Endler L, et al. (2010) BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst Biol* 4: 92. doi:10.1186/1752-0509-4-92.
13. Waltemath D, Adams R, Beard DA, Bergmann FT, Bhalla US, et al. (2011) Minimum Information About a Simulation Experiment (MIASE). *PLoS Comput Biol* 7: e1001122. doi:10.1371/journal.pcbi.1001122.
14. Cooper J, Mirams GR, Niederer SA (2011) High-throughput functional curation of cellular electrophysiology models. *Prog Biophys Mol Biol* 107: 11–20. doi:10.1016/j.pbiomolbio.2011.06.003.
15. Hunter P, Chapman T, Coveney PV, Bono B de, Diaz V, et al. (2013) A vision and strategy for the virtual physiological human: 2012 update. *Interface Focus* 3. Available: <http://rsfs.royalsocietypublishing.org/content/3/2/20130004>. Accessed 3 March 2013.
16. Hunter P, Coveney PV, de Bono B, Diaz V, Fenner J, et al. (2010) A vision and strategy for the virtual physiological human in 2010 and beyond. *Philos Trans R Soc Math Phys Eng Sci* 368: 2595–2614. doi:10.1098/rsta.2010.0048.
17. Markram H (2012) The Human Brain Project. *Sci Am* 306: 50–55. doi:10.1038/scientificamerican0612-50.
18. Duvall PM, Matyas S, Glover A (2007) Continuous integration: improving software quality and reducing risk. Upper Saddle River, NJ: Addison-Wesley.
19. Fink M, Niederer SA, Cherry EM, Fenton FH, Koivumäki JT, et al. (2011) Cardiac cell modelling: Observations from the heart of the cardiac physiome project. *Prog Biophys Mol Biol* 104: 2–21. doi:10.1016/j.pbiomolbio.2010.03.002.
20. Kuhn TS (1976) Mathematical vs. Experimental Traditions in the Development of Physical Science. *J Interdiscip Hist* 7: 1–31. doi:10.2307/202372.
21. Gjuvslund AB, Vik JO, Beard DA, Hunter PJ, Omholt SW (2013) Bridging the genotype–phenotype gap: what does it take? *J Physiol* 591: 2055–2066. doi:10.1113/jphysiol.2012.248864.
22. Sobie EA (2009) Parameter Sensitivity Analysis in Electrophysiological Models Using Multivariable Regression. *Biophys J* 96: 1264–1274. doi:10.1016/j.bpj.2008.10.056.
23. Sarkar AX, Sobie EA (2010) Regression Analysis for Constraining Free Parameters in Electrophysiological Models of Cardiac Cells. *PLoS Comput Biol* 6: e1000914. doi:10.1371/journal.pcbi.1000914.
24. Bondarenko VE, Szigeti GP, Bett GCL, Kim S-J, Rasmusson RL (2004) Computer model of action potential of mouse ventricular myocytes. *Am J Physiol Heart Circ Physiol* 287: H1378–1403. doi:10.1152/ajpheart.00185.2003.

- PeerJ PrePrints
25. Bernus O, Wilders R, Zemlin CW, Verscelde H, Panfilov AV (2002) A computationally efficient electrophysiological model of human ventricular cells. *Am J Physiol - Heart Circ Physiol* 282: H2296–H2308. doi:10.1152/ajpheart.00731.2001.
 26. Ten Tusscher KHWJ, Bernus O, Hren R, Panfilov AV (2006) Comparison of electrophysiological models for human ventricular cells and tissues. *Prog Biophys Mol Biol* 90: 326–345. doi:10.1016/j.pbiomolbio.2005.05.015.
 27. Morin A, Urban J, Adams PD, Foster I, Sali A, et al. (2012) Shining Light into Black Boxes. *Science* 336: 159–160. doi:10.1126/science.1218263.
 28. Peng RD (2011) Reproducible Research in Computational Science. *Science* 334: 1226–1227. doi:10.1126/science.1213847.
 29. LeVeque RJ, Mitchell IM, Stodden V (2012) Reproducible research for scientific computing: Tools and strategies for changing the culture. *Comput Sci Eng* 14: 13.
 30. Howe B (2012) Virtual Appliances, Cloud Computing, and Reproducible Research. *Comput Sci Eng* 14: 36–41. doi:10.1109/MCSE.2012.62.
 31. Drummond DC (2009) Replicability is not Reproducibility: Nor is it Good Science. Available: <http://cogprints.org/7691/>. Accessed 17 June 2013.
 32. Ince DC, Hatton L, Graham-Cumming J (2012) The case for open computer programs. *Nature* 482: 485–488. doi:10.1038/nature10836.
 33. Lykken DT (1968) Statistical significance in psychological research. *Psychol Bull* 70: 151–159. doi:10.1037/h0026141.
 34. Gavish M, Donoho D (2012) Three Dream Applications of Verifiable Computational Results. *Comput Sci Eng* 14: 26–31. doi:10.1109/MCSE.2012.65.
 35. United States National Research Council (2012) Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification. Washington, D.C.: National Academies Press. Available: http://www.nap.edu/catalog.php?record_id=13395. Accessed 28 November 2013.
 36. Pathmanathan P, Gray RA (2013) Ensuring reliability of safety-critical clinical applications of computational cardiac models. *Front Card Electrophysiol* 4: 358. doi:10.3389/fphys.2013.00358.
 37. Niederer SA, Kerfoot E, Benson AP, Bernabeu MO, Bernus O, et al. (2011) Verification of cardiac tissue electrophysiology simulators using an N-version benchmark. *Philos Trans R Soc Math Phys Eng Sci* 369: 4331–4351. doi:10.1098/rsta.2011.0139.
 38. Yu T, Lloyd CM, Nickerson DP, Cooling MT, Miller AK, et al. (2011) The Physiome Model Repository 2. *Bioinformatics* 27: 743–744. doi:10.1093/bioinformatics/btq723.
 39. Cooper J, Niederer SA (2011) A software engineering approach to computational modelling research. In: Viceconti M, Clapworthy G, editors. VPH-FET research roadmap: advanced technologies for the future of the Virtual Physiological Human. VPH-FET consortium. Available: https://www.biomedtown.org/biomed_town/VPHFET/reception/vphfetpubl icrep/VPH-FET_final_roadmap.pdf.
 40. Kohl P, Hunter PJ, Winslow RL (2011) Model interactions: “It is the simple, which is so difficult.” *Prog Biophys Mol Biol* 107: 1–3. doi:10.1016/j.pbiomolbio.2011.07.003.
 41. Tusscher KHWJT, Panfilov AV (2006) Cell model for efficient simulation of wave propagation in human ventricular tissue under normal and pathological conditions. *Phys Med Biol* 51: 6141. doi:10.1088/0031-9155/51/23/014.
 42. Cooper J, Osborne J (2013) Connecting models to data in multiscale multicellular tissue simulations. *Procedia Comput Sci* 18: 712–721. doi:10.1016/j.procs.2013.05.235.
 43. De Bono B, Hoehndorf R, Wimalaratne S, Gkoutos G, Grenon P (2011) The RICORDO approach to semantic interoperability for biomedical data and models: strategy, standards and solutions. *BMC Res Notes* 4: 313. doi:10.1186/1756-0500-4-313.
 44. Juty N, Le Novère N, Laibe C (2011) Identifiers.org and MIRIAM Registry: community resources to provide persistent identification. *Nucleic Acids Res* 40: D580–D586. doi:10.1093/nar/gkr1097.
 45. Henkel R, Le Novère N, Wolkenhauer O, Waltemath D (2012) Considerations of graph-based concepts to manage of computational biology models and associated simulations. *GI-Jahrestagung*. pp. 1545–1551. Available: <http://drupal.lenoverelab.org/perso/lenov/PUBLIS/Henkel2012.pdf>. Accessed 29 December 2013.
 46. Noble D (1960) Cardiac action and pacemaker potentials based on the Hodgkin-Huxley equations.
 47. Molleman A (2002) Patch Clamping: An Introductory Guide to Patch Clamp Electrophysiology. New York: Wiley.
 48. Izhikevich EM (2004) Which Model to Use for Cortical Spiking Neurons? *IEEE Trans Neural Netw* 15: 1063–1070. doi:10.1109/TNN.2004.832719.
 49. Cooper J, Corrias A, Gavaghan D, Noble D (2011) Considerations for the use of cellular electrophysiology models within cardiac tissue simulations. *Prog Biophys Mol Biol* 107: 74–80. doi:10.1016/j.pbiomolbio.2011.06.002.
 50. Grimm V, Railsback SF (2005) Individual-based modeling and ecology. Princeton: Princeton University Press.
 51. Farmer JD, Foley D (2009) The economy needs agent-based modelling. *Nature* 460: 685–686. doi:10.1038/460685a.
 52. An G, Mi Q, Dutta-Moscato J, Vodovotz Y (2009) Agent-based models in translational systems biology. *Wiley Interdiscip Rev Syst Biol Med* 1: 159–171. doi:10.1002/wsbm.45.
 53. Grimm V, Berger U, Bastiansen F, Eliassen S, Ginot V, et al. (2006) A standard protocol for describing individual-based and agent-based models. *Ecol Model* 198: 115–126. doi:10.1016/j.ecolmodel.2006.04.023.
 54. Grimm V, Berger U, DeAngelis DL, Polhill JG, Giske J, et al. (2010) The ODD protocol: A review and first update. *Ecol Model* 221: 2760–2768. doi:10.1016/j.ecolmodel.2010.08.019.
 55. Wilensky U (1999) NetLogo. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. Available: <http://ccl.northwestern.edu/netlogo/>. Accessed 19 December 2013.
 56. Wood C, Williams C, Waldron GJ (2004) Patch clamping by numbers. *Drug Discov Today* 9: 434–441. doi:10.1016/S1359-6446(04)03064-8.
 57. Kodandaramaiah SB, Franzesi GT, Chow BY, Boyden ES, Forest CR (2012) Automated whole-cell patch-clamp electrophysiology of neurons in vivo. *Nat Methods* 9: 585–587. doi:10.1038/nmeth.1993.
 58. Kolb I, Holst G, Goldstein B, Kodandaramaiah SB, Boyden ES, et al. (2013) Automated, in-vivo, whole-cell electrophysiology using an integrated patch-clamp amplifier. *BMC Neurosci* 14: P131. doi:10.1186/1471-2202-14-S1-P131.
 59. Noble D (1962) A modification of the Hodgkin–Huxley equations applicable to Purkinje fibre action and pacemaker potentials. *J Physiol* 160: 317–352.
 60. Bernabeu MO, Pathmanathan P, Pitt-Francis J, Kay D (2010) Stimulus Protocol Determines the Most Computationally Efficient Preconditioner for the Bidomain Equations. *IEEE Trans Biomed Eng* 57: 2806–2815. doi:10.1109/TBME.2010.2078817.
 61. Mirams GR, Arthurs CJ, Bernabeu MO, Bordsas R, Cooper J, et al. (2013) Chaste: An Open Source C++ Library for Computational Physiology and Biology. *PLoS Comput Biol* 9: e1002970. doi:10.1371/journal.pcbi.1002970.