High throughput functional curation of cellular electrophysiology models

Jonathan Cooper[a,*], Gary R. Mirams[b,1], Steven A. Niederer[c]

[a] *Oxford University Computing Laboratory, University of Oxford, Oxford, UK*
[b] *Department of Physiology, Anatomy & Genetics, University of Oxford, Oxford, UK*
[c] *Biomedical Engineering Department, Imaging Sciences & Biomedical Engineering Division, King's College London, UK*

Abstract

Effective reuse of a quantitative mathematical model requires not just access to curated versions of the model equations, but also an understanding of the functional capabilities of the model, and the advisable scope of its application. To enable this "functional curation" we have developed a simulation environment that provides high throughput evaluation of a mathematical model's functional response to an arbitrary user-defined protocol, and optionally compares the results against experimental data. In this study we demonstrate the efficacy of this simulation environment on 31 cardiac electrophysiology cell models using two test-cases. The S1-S2 response is evaluated to characterise the models' restitution curves, and their L-type calcium channel current-voltage curves are evaluated. The significant variation in the response of these models, even when the models represent the same species and temperature, demonstrates the importance of knowing the functional characteristics of a model prior to its reuse.

We also discuss the wider implications for this approach, in improving the selection of models for reuse, enabling the identification of models that exhibit particular experimentally observed phenomena, and making the incremental development of models more robust.

[*] Corresponding author. Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, OX1 3QD.

Email addresses: jonathan.cooper@comlab.ox.ac.uk (Jonathan Cooper), gary.mirams@dpag.ox.ac.uk (Gary R. Mirams), steven.niederer@kcl.ac.uk (Steven A. Niederer)

[1] Joint first author

Keywords: cardiac electrophysiology, CellML, protocol, simulation

1. Introduction

1.1. Aims

Central to the philosophy of the international IUPS Physiome and Virtual Physiological Human (VPH) modelling efforts (Bassingthwaighte, 2000; Cooper et al., 2010) is the integration and coupling of existing biophysical mathematical models, that represent both specific physiological function, and how this function changes in the presence of pathologies or pharmacological agents. This philosophy enables the development of new integrated models that represent interactions and regulatory mechanisms that span across multiple spatial and temporal scales and are intrinsic to biological systems. These integrated models enable the study of complex feedback loops and emergent phenomena, and provide an essential tool for combining extensive disparate experimental and clinical data sets. However, to reuse a mathematical model effectively requires both a correct representation of the original equations, and an understanding of the functional capabilities of the model.

The first of these requirements has been met through the development of community standards for representing mathematical models (e.g. CellML (Lloyd et al., 2004) and SBML (Hucka et al., 2004)) and the publication of models in these formats in public access databases (Li et al., 2010; Lloyd et al., 2008). These reference descriptions are tested and documented as part of the curation process to verify that they conform to language specifications, and that they can reproduce figures and results from the original publication. To minimize the chance that error is introduced in the reproduction of paper results, the simulation protocols necessary to reproduce paper figures can also be encoded (e.g. Nickerson and Buist, 2009). This transparent process of publishing curated model equations and protocols in standard formats provides users with increased confidence that the model is a valid representation of the original work.

The availability of these models and the corresponding confidence in their implementation greatly reduces barriers for the reuse, adaptation or combination of existing models to study a specific question of interest. By reusing existing model parameter sets and formulations it is hoped that the functionality present in the original model will be inherited in the model used for a new study. This poses two challenges. Firstly, with increasing numbers of models and concurrent increases in complexity, identifying the subset of models (or none at all), that exhibit a desired phenomena required for a particular study becomes increasingly challenging. Secondly, if a new model is created it is important to confirm by simulation, rather than assume, that it maintains all of the functionality present in the reused models. Therefore, contrary to the goals of the language standards, the increased availability of mathematical model equations, and the ease of their reuse without a good understanding of their capabilities, is likely to increase model misuse and introduce unintended errors into simulation results (Smith et al., 2007).

Examples of many of the challenges facing model reuse within the VPH and Physiome projects are demonstrated by cardiac cell models, a field that is mature enough to have multiple generations of models and corresponding model reuse. Cardiac electrophysiology cell models have developed over the past 50 years, routinely build on existing models and exhibit complex multiscale functionality. Furthermore previous test cases, where only two models were compared, have demonstrated discrepancies in function between models that purport to represent a common species, cell type and temperature (Cherry and Fenton, 2007; Niederer et al., 2009). The large number of cardiac cellular electrophysiology models available, representing both specific animal species but also specific regions or sub-regions of the heart, have been developed to study a diverse range of cardiac physiology, including variations in temperature, pacing rate, ionic concentrations, sex, and age (Clayton et al., 2010). This diversity means that when approaching a new area of interest an applicable model may already exist, however, evaluating the suitability of current models for a specific situation is not currently readily assessed.

In this paper we consider, design and implement the infrastructure necessary to perform high-throughput *functional curation* of models—subjecting a large number of existing models to commonly-used electrophysiology protocols. Thus we may examine which of the models are capable of replicating experimentally observed results, and evaluate their suitability for reuse in future modelling studies.

1.2. Existing Tools and Extensions

Functional curation requires the capacity to perform simulation protocols on multiple cell models, post-process model outputs and report the results. The proposed implementation for functional curation is outlined in Figure 1. There are three distinct types of components that need to be defined or developed. Firstly, language components are required to provide a formal

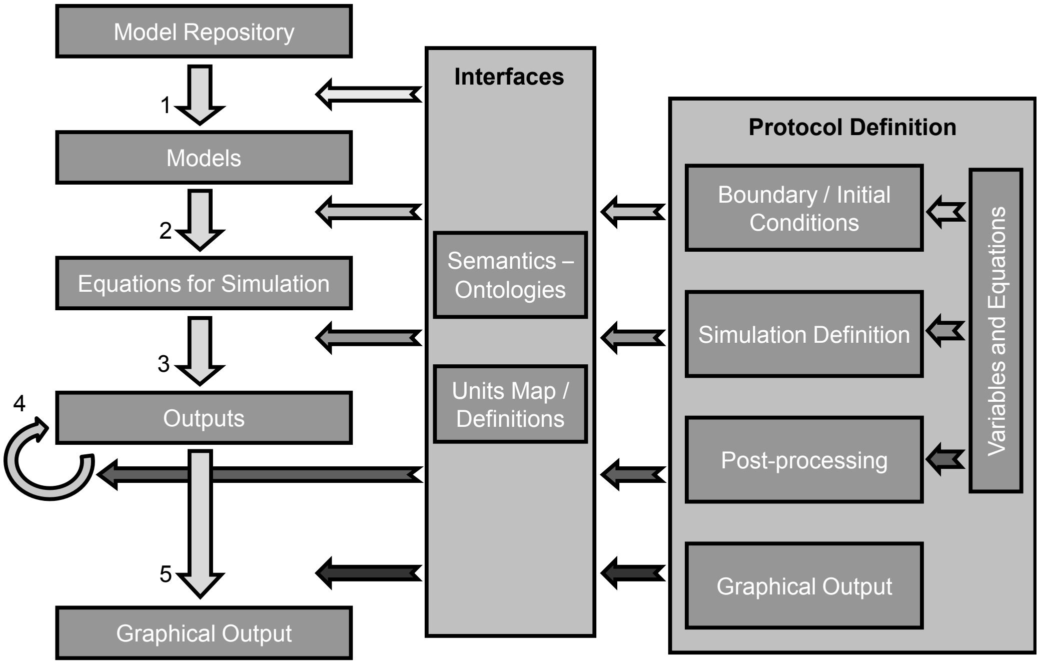

Figure 1: Components of a functional curation system. The steps involved are: (1) annotate models with ontological tags; (2) combine model with protocol mathematics and simplify; (3) run simulation; (4) post-process model outputs; and (5) graphically plot results. All interaction between the protocol definition, model equations, and simulation implementation is filtered through an interface definition, ensuring that biophysical entities and physical units are appropriately mapped.

definition of model equations, simulation protocols, post-processing steps, and desired outputs. Secondly, interface components need to be defined to allow these different languages to be translated into common variable names and units. Thirdly, compute components are required to interpret the languages, implement simulation protocols, perform simulations, post-process model outputs, and produce graphical plots.

Presently there are recognized community standard languages which meet the requirements for some of the language components. As noted above, CellML and SBML provide convenient language standards to formally define cell model equations. Two mark-up language projects currently provide standards for defining both model simulation protocols and graphical outputs—SED-ML (Köhn and Le Novère, 2008), and the CellML Simulation and Graphing Metadata draft standards[2] (CSM and CGM respectively; used in Nickerson and Buist, 2009, for example). These could potentially be used for defining simulation protocols for multiple models. However, in their current forms both languages do not have sufficient functionality for our needs. Protocols defined in SED-ML are tightly tied to a particular model, with all references to variables being model specific, hence typically precluding the definition of a single protocol that can be applied to multiple models; it is thus incapable of supporting the necessary interfacing. Current implementations of both languages only support 'time course' simulations—a single solve of model equations. Neither language is capable of expressing parameter sweeps, or any general form of repeated simulation (although such features are planned for later versions of SED-ML). Finally, neither language supports anything more than extremely basic post-processing steps.[3]

The interface between the protocol language, model equations, and simulation platform is a key aspect of functional curation, in order to support running a protocol on multiple models and coherently comparing the results. Since variable names in models are arbitrary, a mapping is required labelling variables so as to uniquely relate them to a biophysical quantity or protocol construct. Both SBML and CellML support such labelling through RDF annotations, using terms taken from ontologies (Beard et al., 2009; Le Novère

[2] `http://www.cellml.org/specifications/metadata/simulations` and `http://www.cellml.org/specifications/metadata/graphs`

[3] The only post-processing supported is essentially 'map'-type operations, applying an operator element-wise to one or more arrays of outputs to produce a new array. CGM also includes some limited filtering capabilities.

and Finney, 2005).

To compare outputs correctly requires a second interface component for interpreting and converting the physical units associated with variables, since again models vary in the units used for the same quantity. Every quantity in a model and protocol must have its units indicated, and suitable scaling conversions can be added automatically (Cooper and McKeever, 2008). More complex conversions may also be required, utilizing biophysical knowledge to define conversion functions that reference quantities within the model using ontological terms, as above. A common case within cardiac modelling is converting ionic currents, for example to convert a current defined in Amperes per unit area (of a cell membrane) to a current defined in Amperes requires multiplication by the cell surface area (Cooper et al., 2011, this issue).

While there are several simulation environments available that support simulation of CellML models (see Garny et al., 2008), none have the capabilities required of a functional curation system. Both COR (Garny et al., 2003) and OpenCell (formerly known as PCEnv, Garny et al., 2008), for example, support only time-course simulations of single models through a graphical user interface, and Chaste (Pitt-Francis et al., 2009) has primarily used CellML models within cardiac tissue simulations. We are therefore developing a new open source system, based on and extending the Chaste simulation platform, in order to provide high throughput evaluation of models' functional responses to arbitrary user defined protocols. Since available protocol languages, and in particular SED-ML, do not provide sufficient expressivity to encode the protocols we wish to run (as described above), we are also proposing novel language features to support such uses. We focus on complex post-processing and model modifications, which have not been central in the development of existing languages.

The features of our functional curation implementation, and particularly the protocol language features incorporated within it, are detailed in Section 2. Case studies demonstrating the capabilities of the system are described in Section 2.5, and the results shown in Section 3. Finally, in Section 4 we discuss some salient features of our solution within the wider context of physiological modelling, finishing with some concluding thoughts in Section 5.

2. Methods

In this section we outline our proposed implementation of functional curation, treating the steps in the order given in Figure 1, followed by two test cases that demonstrate the utility of this method. Our implementation is guided by twin aims: to produce a system with sufficient expressive power to be able to run a range of complex protocols, and yet also for the protocol language to be relatively straightforward to implement. The software is open source, and available to download from `http://www.comlab.ox.ac.uk/chaste`.

2.1. Language Interfaces

Applying a protocol language to a repository of arbitrary models requires a machine readable mapping of variable names and units between the protocol description and each of the models. This is achieved through metadata labelling of variables within the cell model files and the development of functional unit conversion.

2.1.1. Metadata Labelling

To apply the protocols defined in the protocol language onto existing models requires a translation of variable references in the protocol to the variable names in the model, allowing the protocol to interface with a model purely at the mathematical level. Rather than identifying variables by name, or by location within the model file, we thus require models to be annotated with biological metadata, using an ontology to provide standard names for entities of interest. This technique is independent of a specific ontology—providing that the model and protocol use the same ontology terms.

2.1.2. Unit Conversion

The diversity of modelling strategies and frameworks, compounded with the range of measurement techniques means that the same biological function can be represented by equations with incompatible units. This can be demonstrated with a cell membrane current as described above, or more subtly a concentration flux from the cytosol to an organelle with the units of Mole per litre per second does not explicitly state the volume of the concentration. This means it could be a concentration from the cytosol or the organelle. If comparing two fluxes from two models with these units, they must be with reference to the same volume for valid comparisons to be made. We use the framework described in (Cooper et al., 2011, this issue) to support the definition of units conversion functions, which may use the values of model variables (identified by ontological labels) to perform complex conversions and hence enable meaningful comparisons between models.

2.2. Determining Equations for Simulation

The equations to be simulated will not always be identical to those in the original reference model implementation. Firstly, the model equations must be changed to impose boundary and initial conditions. Setting initial conditions may take the form of setting the values of model variables at the start of a simulation, or additional equations may be added to calculate suitable initial conditions from other parameters, including the results of previous simulations. Thus the protocol language includes the ability to represent mathematical equations, and the same kinds of equations as might appear in a model. Setting 'boundary conditions' requires this ability to a greater extent, and may also involve replacing parts of the model with new equations. For example, the trans-membrane potential V in an electrophysiology model is generally a time-varying state variable defined by an ordinary differential equation (ODE). If a 'voltage clamp' protocol is applied, however, the value of V is instead fixed by the protocol, and its defining ODE should be removed from the model to avoid a conflict.

Secondly, the protocol language allows the specification of 'outputs'—those variables whose values over the course of the simulation should be recorded for post-processing. Often it will be desirable to save only the relevant subset of the data produced by a model, and so the variables to be recorded, and their sampling intervals, should be specified. These outputs may not appear directly in the model, in which case new equations need to be added, as above. Also, it may not be necessary to simulate all of the model equations in order to determine the desired outputs. For some protocols, this may even mean that there is no time-varying component to the simulation at all, for example when evaluating the substrate sensitivity of rapid steady state transporter models or enzyme kinetics. Faster simulations may therefore be achieved in these cases by analysing the model equations and removing those which are not required in computing the outputs.

Standards are currently available within existing languages for defining the numerical algorithm and algorithm parameters for evaluating a model. Both SED-ML and CSM allow time-course simulations to be specified with various numerical methods and parameters thereof, and we have adopted such standards within our language.

Solving repeated time-course simulations while adjusting input information allows simulations to sweep parameters for fitting, or to characterise a response as a function of an input. In order to allow these more complex simulations, we use the ideas of the nested simulation proposal for SED-ML (Bergmann, 2010). This allows loops to be applied to simulations, and with each iteration of the loop a parameter of the model or protocol is altered. This mechanism enables the implementation of a parameter sweep, while multiple layers of nesting then allows for multi-dimensional sweeps.

2.3. Post-processing

Post-processing of simulation outputs (and of post-processed outputs) must be included in a protocol language used for functional curation. Often complex processing must be performed in order to obtain the quantities of interest. Careful balancing of this goal of expressivity, with the need to minimise the burden for implementations, is therefore required. We thus define a small set of elementary language features, which can be combined within the language itself to achieve complex behaviour. Any implementations of the language then need merely to support the basic set.

The form of the model output data strongly influences the types of the elementary language features. Previous languages, including SED-ML and CellML-based languages, only supported single simulations, with simulation results therefore stored as single dimension arrays. Our protocol language supports nested simulations, which leads naturally to outputs being represented as n-dimensional arrays of double-precision floating point values. The number of dimensions is given by the number of levels of nesting, and the size of each dimension by the number of (sampling) points in the range over which the corresponding simulation loops.

We therefore define an array-based language building on this single datatype. Many similar, and general purpose, array-based languages exist (e.g. Burke, 1996; Jenkins and Jenkins, 1993; McGraw et al., 1985; Scholz, 2003), perhaps most notably MATLAB®, NumPy (Oliphant, 2007), and APL (Iverson, 1962). To reduce the implementation burden and complexity our language will be less powerful, with features only added if necessary to express desired protocols. A key feature is that arrays cannot be modified once created (i.e. it is a pure functional language), which simplifies processing. Each operation in the language takes arrays as input, and produces new arrays as output. The language supports the definition of functions, and collections of functions, in order to allow libraries of common operations to be developed, thus reducing the burden for protocol authors.

The language includes basic arithmetic operations that are available within MathML, and used by both SBML and CellML. Supporting these should therefore not impose any additional burden on implementations. In addition to this, the language contains operations for working with n-dimensional arrays. There are just six foundational operations, from which all of the post-processing required by our case studies can be constructed.

Array creation Arrays can be created directly within the protocol language. As array content can not be changed, arrays must be populated at initiation. This can be done simply by listing the entries, or defining the values as functions of the element indices, existing arrays, or as constants.

Map This operation allows basic arithmetic functions, (i.e. `+`, `-`, `>`, `<`, `abs`, etc.) to be applied element-wise to one or more arrays.

Fold This allows an operation to be performed on each element along one dimension of an array to produce a single value, yielding a result array with one fewer dimensions than the input. For example addition along an array will produce the sum, or a comparison along the array can be used to determine the maximum or minimum value.

Find This locates non-zero entries within the input array, and returns an array of their indices. The result array can be used as an input to the `index` operation.

Index This generates a new array consisting only of those entries in the input whose indices appear in the indices array supplied.

Views New arrays may be created by taking sub-arrays of existing arrays. These can contain contiguous elements, or be formed by striding across the input, taking every second element along one dimension for example.

A key feature, reflected also in the core language operations, is that functions in our language can be agnostic as to the dimensionality of the inputs.

This allows them to function without the number of dimensions being hard-coded, and hence permits multi-dimensional parameter sweeps without any limit on the number of parameters being varied.

2.4. Graphical Outputs

Finally, to formalise the link between model equations and results as they might appear in a published paper, it must be possible to define the graphical figure outputs produced from the model outputs and subsequent post-processing steps. In this area we will also leverage existing work, since both SED-ML and CGM contain features for describing graphs (our prototype implementation currently uses hard-coded gnuplot scripts). Results can also be output to file if further processing for comparison is required. Again little has thus far been done in this area, but formats such as BiosignalML[4] or SBRML (Dada et al., 2010) would be potential targets for future work.

2.5. Case Studies

To demonstrate the utility of functional curation and show the functionality enabled through the proposed combination of languages, interfaces, and computation code we have performed two standard protocols, specifically the S1-S2 and the $\mathrm{I_{CaL}}$ current–voltage plot, on a repository of cardiac cell models.

2.5.1. S1-S2 Protocol

The first test case is the "S1-S2" restitution curve protocol first described by Nolasco and Dahlen (1968). This protocol consists of a series of stimuli applied at a steady pacing cycle length (S1) until action potentials become regular (reach a limit-cycle, mathematically speaking), followed by a single stimulus applied at a different cycle length (S2). The action potential duration resulting from the S2 stimulus is plotted against the diastolic interval (DI) preceding that stimulus, as shown in Figure 2.

Experimental restitution slopes for healthy cells typically follow the trend shown in Figure 2, with APD shortening following a smaller DI, and tending to a steady size as the DI is increased.

This protocol was implemented as follows. Firstly, to allow the protocol language to interact with the models the CellML implementations of these

[4] `http://www.embs.org/techcomm/tc-cbap/biosignal.html`

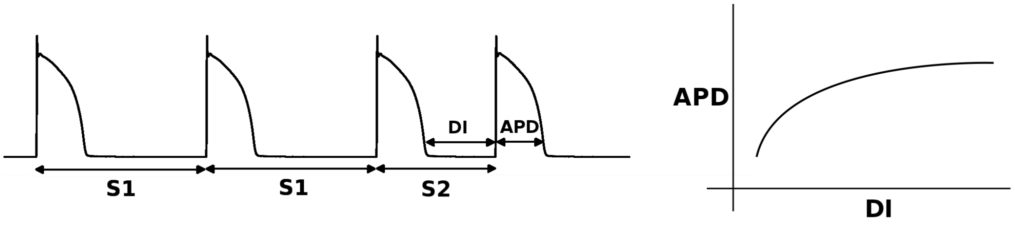

Figure 2: Left: the S1-S2 pacing protocol with S1, S2 pacing cycle lengths, diastolic interval (DI) and action potential duration (APD) marked. Right: the axes and shape of a typical restitution curve, formed by repeating the protocol with varying S2 intervals, for a fixed choice of S1.

models were annotated with metadata to identify the variables used in the protocol definition. These included the transmembrane potential, simulation time, and the stimulus current, amplitude, duration, and period variables.

The simulation definition for this protocol consists of two nested loops. The outer loop modifies a variable specifying the S2 interval, and the inner loop performs a time-course simulation. For each inner loop iteration the protocol defines all of the state variables to be initialised to be on the limit cycle for S1 pacing. For each outer loop the period of the stimulation current is incremented to adjust the S2 interval. As the S1-S2 protocol is based solely on the membrane potential transient only the membrane potential is specified as being output every 1 ms within the problem definition. The membrane potential output is thus a 2 dimensional array in time and S2 interval.

The final outcome of the S1-S2 protocol is a plot of APD against DI. These quantities are evaluated by a single `apd` function, given voltage V and time t as inputs. It has been designed to be dimension-agnostic, supporting input voltage traces from arbitrarily nested simulations. The function operates on a N dimensional V array and a 1D time array to produce a N dimensional APD array, where the time dimension from the V array has been replaced with a dimension corresponding to the maximum number of APs detected in one time-course simulation.[5]

We note a few salient features of how the computation is performed here;

[5] For the purposes of this study, entries corresponding to simulations with fewer APs are padded with an arbitrary value.

full details can be found in the online supplement. Firstly, the start of each AP is identified by finding local maxima of the gradient of V. Since array operations process the whole array, in order to determine features of individual APs (rather than of a whole time-course) many steps work on extended arrays with an extra dimension varying over AP number (i.e. its extent is the maximum number of APs recorded in one simulation). Having identified the peak and resting potential for each AP (using local maxima and minima functions), linear interpolation, again a function defined in terms of the basic post-processing operations, is used to locate the first times before and after each peak where the resting potential is reached, and these times are taken as the start and end of each AP; the difference between them is the APD. Finally, the DIs are calculated by taking the element-wise difference between two arrays: the first containing the intervals between AP starts, and the second being a sub-array of the APDs, with the last removed for each run.

We also use the post-processing functionality to perform some filtering of these post-processed results. Firstly, we extract the first DI and the second APD for each S2 interval, since these reflect the AP produced by the S2 stimulus. Secondly we remove entries with a very small or negative DI. A stimulus at a negative DI could induce a full early after depolarization. Alternatively at a small or negative DI the stimulus current itself could form a small depolarization, and cause a subsequent delay in repolarization of the last S1 AP. Neither case is part of a standard experimental S1-S2 protocol, so we remove them from the post-processing.

2.5.2. ICaL Voltage Clamp Protocol

The second test case is based on the generation of L-type calcium channel current–voltage (IV) curves, for different extracellular calcium concentrations. The L-type calcium channel current's reversal potential will depend on the difference between the intra- and extra-cellular calcium concentrations. The channel also inactivates in response to an accumulation of intracellular calcium, changing the cellular action potential duration accordingly (Grandi et al., 2009). The extracellular calcium concentration dependence of the L-type calcium channel therefore has important cell-level effects. The IV curve protocol consists of clamping the voltage to a 'holding potential' (in the literature this is between $-50\,\mathrm{mV}$ and $-40\,\mathrm{mV}$) followed by a step change to a 'testing potential'. We illustrate the principle of this protocol in Figure 3.

The largest magnitude of the L-type calcium current produced as a result

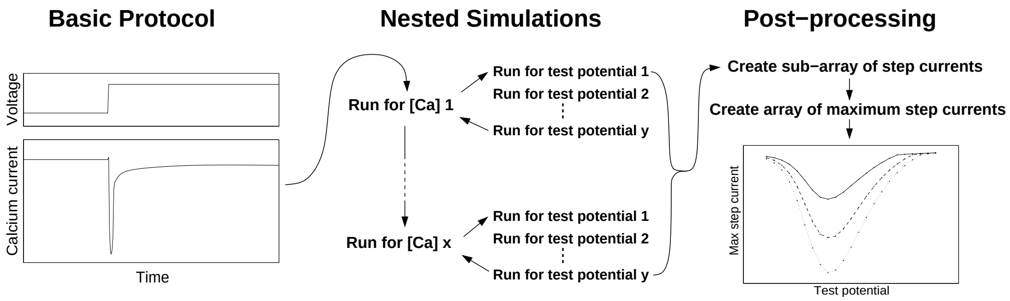

Figure 3: Left: a voltage clamp and the resulting L-type calcium current. Centre: nested simulations formed by repeating the protocol for different extracellular calcium concentrations, and also different test potentials. Right: post-processing steps and plotting.

of the voltage step (the minimum since the current is negative) is plotted for each test potential. The protocol is carried out at varied extracellular calcium concentrations (we have chosen 50%, 100% and 150% of the models' default values).

Since both the test potential and extracellular calcium concentration are varied, this protocol is represented by three nested loops. The outer loop modifies the extracellular calcium concentration, the middle loop modifies the test potential, and the inner loop performs the time-course simulation. A time-dependent modification occurs at $t = 0$ and the holding potential is changed to the test potential (which in this case depends on the state of the middle-loop, see Figure 3).

In the protocol definition the membrane potential is specified as a protocol input parameter, with time and the L-type calcium current set as outputs, to be recorded every 0.05 ms in order to capture the fast transient behaviour of the current. In this case the whole model is not required and the model and protocol are combined to produce a much reduced set of equations for simulation, as described in Section 2.2.

Despite this protocol having an extra level of complexity with an additional nested loop in the simulation definition, the post-processing is much simpler. Firstly a `map` on the time array produces an array containing non-zero values only where the time is greater than zero, i.e. after the test potential step was applied. Applying `find` to this array yields the indices of these entries, which can be used to `index` the L-type calcium current array, giving a sub-array containing just the currents resulting from the voltage steps ("step currents"). A minimum function is created by using a `fold`, applied to the step currents array, and the result allocated to a new array for plotting.

2.5.3. Model Repository

The two test cases are performed on a repository of cardiac cell models representing Purkinje, atrial and ventricular cells, shown in Table 1. These models represent a broad section of available cardiac cell models and provide a comprehensive test bed for evaluating the utility of functional curation. The results of this evaluation appear in the next section.

3. Results

To demonstrate the ability of the protocol language, interfaces and simulation environment to enable functional curation of a repository of models we have applied the two test protocols above to 31 cell models.

In both case studies the differential equations are solved using CVODE (Hindmarsh et al., 2005) with relative and absolute tolerances of 10^{-5} and 10^{-7} respectively.

3.1. Test Case 1: S1-S2 Protocol

In Figure 4 we present S1-S2 restitution curves obtained using the functional curation system, for human ventricular cell models. Despite previous papers stimulating cell models at S1= 1000 ms and comparing with the trace shown in Figure 4 from Morgan et al. (1992), the original paper states that the recording in question was taken with S1= 600 ms at the epicardium, and so we use this protocol (and take the epicardial versions of the models where appropriate).

In Figure 5 we present restitution curves for dog ventricular cell models. In order to facilitate comparison with experimental data the S1 interval for these models was set to 2000 ms.

3.2. Test Case 2: L-type Calcium Channel Properties

IV curves for the L-type calcium current can be seen in Figure 6 for a representative selection of the models. IV curves for the remaining models can be found in the supplementary material, Figures S3 and S4.

Table 1: Cardiac electrophysiology models used in this study.

Model	Species	Cell Type
Aslanidi et al. (2009)	Dog	Purkinje
Beeler and Reuter (1977)	Mammalian	Ventricular
Bondarenko et al. (2004)	Mouse	Ventricular
Corrias et al. (2011)	Rabbit	Purkinje
Courtemanche et al. (1998)	Human	Atrial
Decker et al. (2009)	Dog	Ventricular
Earm and Noble (1990)	Rabbit	Atrial
Espinosa (1998)	Rat	Ventricular
Faber and Rudy (2000)	Mammalian	Ventricular
Fink et al. (2008)	Human	Ventricular
Fox et al. (2002)	Dog	Ventricular
Grandi et al. (2010)	Human	Ventricular
Hilgemann and Noble (1987)	Rabbit	Atrial
Hund and Rudy (2004)	Dog	Ventricular
Iribe et al. (2006)	Guinea-pig	Ventricular
Iyer et al. (2004)	Human	Ventricular
Iyer et al. (2007)	Human	Ventricular
Lindblad et al. (1996)	Rabbit	Atrial
Livshitz and Rudy (2007)	Dog	Ventricular
Luo and Rudy (1991)	Guinea-pig	Ventricular
Mahajan et al. (2008)	Rabbit	Ventricular
Matsuoka et al. (2003)	Guinea-pig	Ventricular
Noble et al. (1991)	Guinea-pig	Ventricular
Pandit et al. (2001)	Rat	Ventricular
Pásek et al. (2008)	Guinea-pig	Ventricular
Priebe and Beuckelmann (1998)	Human	Ventricular
Sakmann et al. (2000)	Guinea-pig	Ventricular
Shannon et al. (2004)	Rabbit	Ventricular
Ten Tusscher et al. (2004)	Human	Ventricular
Ten Tusscher and Panfilov (2006)	Human	Ventricular
Winslow et al. (1999)	Dog	Ventricular

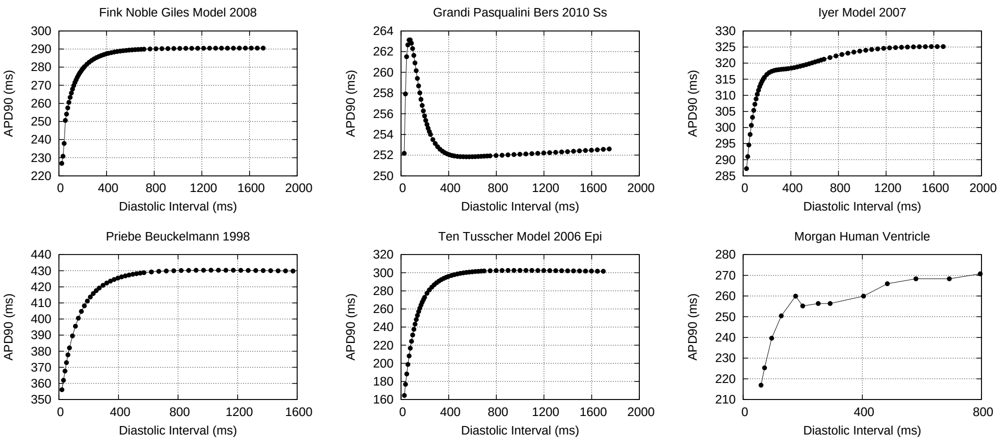

Figure 4: S1-S2 restitution curves for human ventricle models in Table 1, where a choice was available the epicardial model is shown, none of the models showed significant variation when endocardial versions were simulated (S1= 600 ms), experimental data shown in bottom right from Morgan et al. (1992) (for S1= 600 ms).

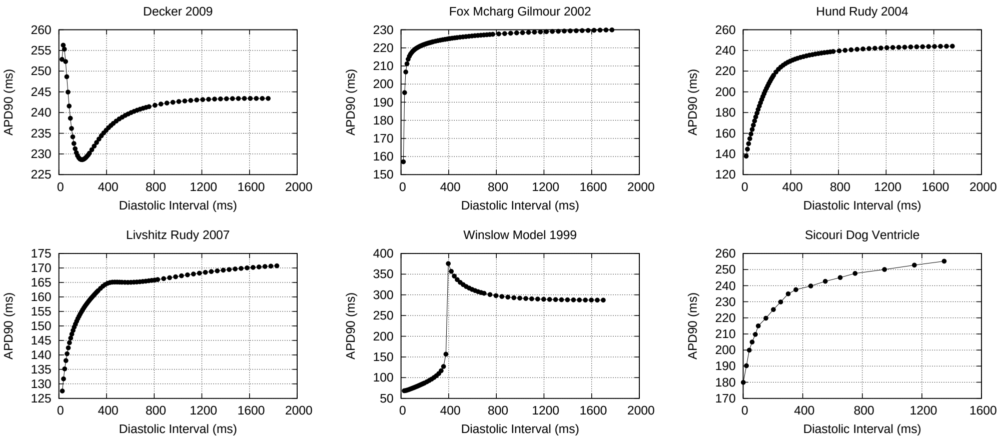

Figure 5: S1-S2 restitution curves for all the canine ventricle models in Table 1, (S1= 2000 ms), experimental data shown in bottom right from Sicouri and Antzelevitch (1991) (for S1= 2000 ms).

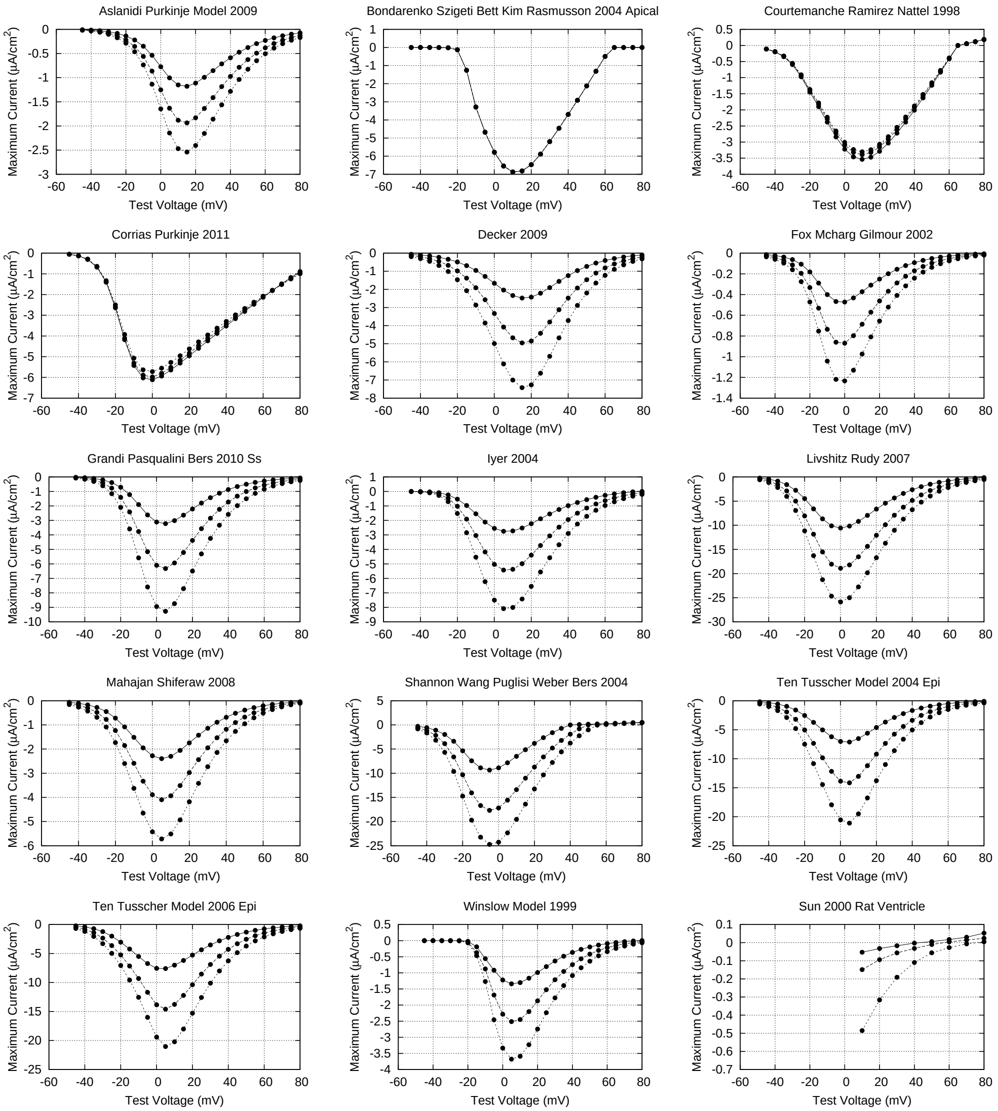

Figure 6: ICaL IV curves for 14 of the models, generated by recording the largest L-type calcium current resulting from a voltage clamp from a holding potential of $-50\,\text{mV}$ to the test voltage plotted. Solid lines 50%, dashed lines 100%, dotted lines 150% of model's default extracellular calcium concentration. Experimental data for a subset of the voltages is shown in the bottom right plot, from Sun et al. (2000, Figure 5(b)) for 0.1 mM, 0.3 mM and 1 mM extracellular calcium (converted into consistent units using a capacitance value of 169pF (Krishna et al., 2010)), most of the models illustrate the same qualitative behaviour in this voltage range.

4. Discussion

In this study we have developed and demonstrated a functional curation system. The system enables the identification of models that exhibit the salient features of an expected response to a protocol. It also allows the rational selection of an existing model for a new study based on the ability of the model to produce a set of expected outputs for a predefined set of protocols. Similarly, the inability of any existing model to replicate a set of protocols motivates the creation of a new or the adaptation of an existing model.

The two case studies evaluated provide specific examples of the utility of functional curation. Specifically, the S1-S2 restitution protocol for Grandi et al. in Figure 4 exhibits strange behaviour, inconsistent with results shown in the original paper (Grandi et al., 2010, Figure 5C). This suggests the possibility that the CellML implementation of the model still requires curation. In Figure 5 the Winslow model appears to yield unusual behaviour inconsistent with experimental restitution curves, which can be attributed to a threshold in the L-type calcium current activation causing the action potential to switch from a "spike" to a "spike and dome" morphology at ∼400 ms. Similarly, in the analysis of ion channel models in Figure 6, the Bondarenko et al. (2004) L-type calcium current model showed no dependence on extracellular calcium, which is due to the Nernst potential appearing as a fixed parameter in the model equations. Aside from these significant differences, subtle variations in model function were also observed for both the S1-S2 restitution and L-type calcium current protocols. By automatically and comprehensively testing multiple models and protocols users can have confidence that the model they have chosen or developed provides a good approximation of the desired physiology.

Functional curation can also facilitate the model development process. By defining collections of protocols and desired outputs new or adapted models can continually be evaluated during the development process to ensure that central functionality is not lost. The publication of the set of protocols and corresponding desired outputs used to develop a model alongside the model equations provides a means of confirming that the model equations are correctly implemented. This also facilitates continuous model development, as new protocols and outputs can be added to the existing set concurrent with the addition of new model functionality.

The use of functional curation for model development and selection has significant potential and is conceptually simple. However, the practical, automated and general implementation of a functional curation system requires the development of multiple interlinked components. The two test cases in this study evaluated functionality that modellers of arrhythmias or calcium dynamics would wish to perform on a model repository to determine the best model for a particular question. Although basic, these simulations still required advanced features from the functional curation system in order to cope with the variations between models, multiple simulations, and multidimensional post-processing. They thus provide an excellent proof of concept, being beyond the capability of existing standard protocol languages; both currently require purpose-built tools in a general programming language. In the S1-S2 protocol, many corner cases that can influence the results, such as the presence of a notch, early or delayed after depolarisations, or ectopic beats are handled well. Not only does allowing for these require complex processing from the protocol system, but more importantly publishing the protocol used in a standard format means that the algorithm used is known, and hence avoids doubt. The S1-S2 protocol also exercises additional features of the system, including nested simulations (to perform a time-course simulation for each S2 interval), and model modification (to vary the S2 interval across these runs). These features are further exercised by the ICaL protocol, which demonstrates the use of boundary conditions in clamping a state variable, and model reduction to evaluate only the equations required in computing the outputs of interest.

As part of the development of the functional curation system a prototype protocol language, post-processing functions, and cell model component labelling were required. The goal of this project was not to propose new community standards but to demonstrate the requirements and benefits of a functional curation system. Therefore the modular design of the framework allows for the adoption of new simulation platforms, mark-up languages, or ontologies as these develop sufficiently to meet the system requirements. This modularity will allow for the introduction of SED-ML protocol definitions if the functionality defined and developed here is adopted by the SED-ML community, or the use of standardised ontologies once they are developed, formalised, and ratified by the cell modelling community.

5. Conclusion

This study demonstrates the utility of functional curation as a concept, and the capabilities of our implementation. We have provided a system which can run many of the standard electrophysiology protocols on any suitably annotated CellML file, and can readily be extended to alternate mark-up languages and cell types. This tool will improve the selection of models for reuse, enabling the identification of models that exhibit particular experimentally observed phenomena. It can also be used to ensure that incremental development of models to investigate new scenarios does not remove their ability to replicate desirable experimental results.

Acknowledgements

The authors would like to thank all other members of the Chaste development team for fruitful discussions on the topics considered in this manuscript, and Martin Fink for his comments.

Role of the funding source

JC is partially supported by the European Commission DG-INFSO under grant numbers 223920 (VPH-NoE) and 224381 (preDiCT). GRM is also supported under grant number 224381 (preDiCT). SAN is supported by United Kingdom Engineering and Physical Sciences Research Council through grants EP/F043929/1 and EP/F059361/1. The funding bodies had no other role in this work.

Editors' note

Please see also related communications in this issue by Atia et al. (2011) and Winslow and Greenstein (2011).

References

Aslanidi, O.V., Stewart, P., Boyett, M.R., Zhang, H., 2009. Optimal velocity and safety of discontinuous conduction through the heterogeneous Purkinje-ventricular junction. Biophysical Kournal 97, 20–39.

Atia, J., Benson, A.P., van den Berg, H.A., Blanks, A.M., Choi, C., 2011. Towards a computational reconstruction of the electrodynamics of premature and full term human labour. Progress in Biophysics and Molecular Biology NUMBER, [note to publisher: please update before print].

Bassingthwaighte, J.B., 2000. Strategies for the Physiome Project. Ann Biomed Eng 28, 1043–1058.

Beard, D.A., Britten, R., Cooling, M.T., Garny, A., Halstead, M.D.B., Hunter, P.J., Lawson, J., Lloyd, C.M., Marsh, J., Miller, A., Nickerson, D.P., Nielsen, P.M.F., Nomura, T., Subramanium, S., Wimalaratne, S.M., Yu, T., 2009. CellML metadata standards, associated tools and repositories. Phil Trans R Soc A 367, 1845–1867.

Beeler, G.W., Reuter, H., 1977. Reconstruction of the action potential of ventricular myocardial fibres. J Physiol 268, 177.

Bergmann, F.T., 2010. A simple nested simulation for SED-ML. Nature Precedings .

Bondarenko, V.E., Szigeti, G.P., Bett, G.C.L., Kim, S.J., Rasmusson, R.L., 2004. Computer model of action potential of mouse ventricular myocytes. American Journal of Physiology — Heart and Circulatory Physiology 287, H1378.

Burke, C., 1996. J and APL. Iverson Software Inc.. Toronto, Canada.

Cherry, E., Fenton, F., 2007. A tale of two dogs: analyzing two models of canine ventricular electrophysiology. American Journal of Physiology — Heart and Circulatory Physiology 292, H43.

Clayton, R.H., Bernus, O., Cherry, E.M., Dierckx, H., Fenton, F.H., Mirabella, L., Panfilov, A.V., Sachse, F.B., Seemann, G., Zhang, H., 2010. Models of cardiac tissue electrophysiology: progress, challenges and open questions. Progress in Biophysics and Molecular Biology 104, 22–48.

Cooper, J., Cervenansky, F., Fabritiis, G.D., Fenner, J., Friboulet, D., Giorgino, T., Manos, S., Martelli, Y., Villà-Freixa, J., Zasada, S., Lloyd, S., McCormack, K., Coveney, P.V., 2010. The Virtual Physiological Human ToolKit. Phil Trans R Soc A 368, 3925–3936.

Cooper, J., Corrias, A., Gavaghan, D., Noble, D., 2011, this issue. Considerations for the use of cellular electrophysiology models within cardiac tissue simulations. Progress in Biophysics and Molecular Biology NUMBER, [note to publisher: please update before print].

Cooper, J., McKeever, S., 2008. A model-driven approach to automatic conversion of physical units. Softw Pract Exper 38, 337–359.

Corrias, A., Giles, W., Rodriguez, B., 2011. Ionic mechanisms of electrophysiological properties and repolarization abnormalities in rabbit Purkinje fibers. American Journal of Physiology — Heart and Circulatory Physiology , in press.

Courtemanche, M., Ramirez, R.J., Nattel, S., 1998. Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. American Journal of Physiology — Heart and Circulatory Physiology 275, H301.

Dada, J.O., Spasić, I., Paton, N.W., Mendes, P., 2010. SBRML: a markup language for associating systems biology data with models. Bioinformatics 26, 932–938.

Decker, K.F., Heijman, J., Silva, J.R., Hund, T.J., Rudy, Y., 2009. Properties and ionic mechanisms of action potential adaptation, restitution, and accommodation in canine epicardium. American Journal of Physiology — Heart and Circulatory Physiology 296, H1017.

Earm, Y.E., Noble, D., 1990. A model of the single atrial cell: relation between calcium current and calcium release. Proc Roy Soc B 240, 83–96.

Espinosa, L.E., 1998. L'échange Na+/Ca2+ dans l'hypertrophie ventriculaire d'altitude chez le rat: Etude electrophysiologique et utilisation du modele "oxsoft heart". Ph.D. thesis. Leon.

Faber, G.M., Rudy, Y., 2000. Action potential and contractility changes in [Na+]i overloaded cardiac myocytes: a simulation study. Biophysical Journal 78, 2392–2404.

Fink, M., Noble, D., Virag, L., Varro, A., Giles, W.R., 2008. Contributions of HERG K+ current to repolarization of the human ventricular action potential. Progress in Biophysics and Molecular Biology 96, 357–376.

Fox, J.J., McHarg, J.L., Gilmour Jr, R.F., 2002. Ionic mechanism of electrical alternans. American Journal of Physiology — Heart and Circulatory Physiology 282, H516.

Garny, A., Kohl, P., Noble, D., 2003. Cellular Open Resource (COR): a public CellML based environment for modelling biological function. Int J Bif Chaos 13, 3579–3590.

Garny, A., Nickerson, D., Cooper, J., dos Santos, R.W., McKeever, S., Nielsen, P., Hunter, P., 2008. CellML and associated tools and techniques. Phil Trans R Soc A 366, 3017–3043.

Grandi, E., Pasqualini, F., Pes, C., Corsi, C., Zaza, A., Severi, S., 2009. Theoretical investigation of action potential duration dependence on extracellular Ca2+ in human cardiomyocytes. Journal of Molecular and Cellular Cardiology 46, 332–342.

Grandi, E., Pasqualini, F.S., Bers, D.M., 2010. A novel computational model of the human ventricular action potential and ca transient. J Mol Cellular Cardiology 48, 112–121.

Hilgemann, D.W., Noble, D., 1987. Excitation-contraction coupling and extracellular calcium transients in rabbit atrium: reconstruction of basic cellular mechanisms. Proc Roy Soc B 230, 163.

Hindmarsh, A., Brown, P., Grant, K., Lee, S., Serban, R., Shumaker, D., Woodward, C., 2005. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. ACM Trans Math Software (TOMS) 31, 363–396.

Hucka, M., Finney, A., Bornstein, B., Keating, S., Shapiro, B., Matthews, J., Kovitz, B., Schilstra, M., Funahashi, A., Doyle, J., Kitano, H., 2004. Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup Language (SBML) project. Systems Biology 1, 41–53.

Hund, T.J., Rudy, Y., 2004. Rate dependence and regulation of action potential and calcium transient in a canine cardiac ventricular cell model. Circ 110, 3168–3174.

Iribe, G., Kohl, P., Noble, D., 2006. Modulatory effect of calmodulin-dependent kinase II (CaMKII) on sarcoplasmic reticulum Ca2+ handling and interval–force relations: a modelling study. Phil Trans R Soc A 364, 1107.

Iverson, K., 1962. A Programming Language. Wiley, New York.

Iyer, V., Hajjar, R.J., Armoundas, A.A., 2007. Mechanisms of abnormal calcium homeostasis in mutations responsible for catecholaminergic polymorphic ventricular tachycardia. Circulation Research 100, 10–22.

Iyer, V., Mazhari, R., Winslow, R.L., 2004. A computational model of the human left-ventricular epicardial myocyte. Biophysical Journal 87, 1507–1525.

Jenkins, M., Jenkins, W., 1993. The Q'Nial language and reference manuals. Nial Systems Ltd.. Ottawa, Canada.

Köhn, D., Le Novère, N., 2008. SED-ML — an XML format for the implementation of the MIASE guidelines, in: Heiner, M., Uhrmacher, A. (Eds.), Computational Methods in Systems Biology. Springer Berlin / Heidelberg. volume 5307 of *Lecture Notes in Computer Science*, pp. 176–190.

Krishna, A., Sun, L., Valderrábano, M., Palade, P., Clark, J., 2010. Modeling CICR in rat ventricular myocytes: voltage clamp studies. Theoretical Biology and Medical Modelling 7, 43.

Le Novère, N., Finney, A., 2005. A simple scheme for annotating SBML with references to controlled vocabularies and database entries. Online.

Li, C., Donizelli, M., Rodriguez, N., Dharuri, H., Endler, L., Chelliah, V., Li, L., He, E., Henry, A., Stefan, M.I., Snoep, J.L., Hucka, M., Le Novère, N., Laibe, C., 2010. BioModels Database: an enhanced, curated and annotated resource for published quantitative kinetic models. BMC Systems Biology 4, 92.

Lindblad, D.S., Murphey, C.R., Clark, J.W., Giles, W.R., 1996. A model of the action potential and underlying membrane currents in a rabbit atrial cell. American Journal of Physiology — Heart and Circulatory Physiology 271, H1666.

Livshitz, L.M., Rudy, Y., 2007. Regulation of Ca2+ and electrical alternans in cardiac myocytes: role of CAMKII and repolarizing currents. Am J Physiol — Heart Circ Physiol 292, H2854.

Lloyd, C.M., Halstead, M.D., Nielsen, P.F., 2004. CellML: its future, present and past. Progress in Biophysics and Molecular Biology 85, 433–450.

Lloyd, C.M., Lawson, J.R., Hunter, P.J., Nielsen, P.F., 2008. The CellML model repository. Bioinformatics 24, 2122–2123.

Luo, C.H., Rudy, Y., 1991. A model of the ventricular cardiac action potential. depolarization, repolarization, and their interaction. Circ Res 68, 1501.

Mahajan, A., Shiferaw, Y., Sato, D., Baher, A., Olcese, R., Xie, L.H., Yang, M.J., Chen, P.S., Restrepo, J.G., Karma, A., Garfinkel, A., Qu, Z., Weiss, J.N., 2008. A rabbit ventricular action potential model replicating cardiac dynamics at rapid heart rates. Biophys J 94, 392–410.

Matsuoka, S., Sarai, N., Kuratomi, S., Ono, K., Noma, A., 2003. Role of individual ionic current systems in ventricular cells hypothesized by a model study. The Japanese Journal of Physiology 53, 105–123.

McGraw, J., Skedzielewski, S., Allan, S., Oldehoeft, R., et al., 1985. Sisal: Streams and Iteration in a Single Assignment Language: Reference Manual Version 1.2. Lawrence Livermore National Laboratory, LLNL. Livermore, California.

Morgan, J.M., Cunningham, D., Rowland, E., 1992. Dispersion of monophasic action potential duration: demonstrable in humans after premature ventricular extrastimulation but not in steady state. JACC 19, 1244–53.

Nickerson, D.P., Buist, M.L., 2009. A physiome standards-based model publication paradigm. Phil Trans R Soc A 367, 1823–1844.

Niederer, S.A., Fink, M., Noble, D., Smith, N.P., 2009. A meta-analysis of cardiac electrophysiology computational models. Experimental Physiology 94, 486–495.

Noble, D., Noble, S.J., Bett, G.C.L., Earm, Y.E., Ho, W.K., So, I.K., 1991. The role of sodium-calcium exchange during the cardiac action potential. Annals of the New York Academy of Sciences 639, 334–353.

Nolasco, J.B., Dahlen, R.W., 1968. A graphic method for the study of alternation in cardiac action potentials. Journal of Applied Physiology 25, 191.

Oliphant, T.E., 2007. Python for scientific computing. Computing in Science and Engineering 9, 10–20.

Pandit, S.V., Clark, R.B., Giles, W.R., Demir, S.S., 2001. A mathematical model of action potential heterogeneity in adult rat left ventricular myocytes. Biophysical Journal 81, 3029–3051.

Pásek, M., Simurda, J., Orchard, C.H., Christé, G., 2008. A model of the guinea-pig ventricular cardiac myocyte incorporating a transverse-axial tubular system. Progress in Biophysics and Molecular Biology 96, 258–280.

Pitt-Francis, J., Pathmanathan, P., Bernabeu, M.O., Bordas, R., Cooper, J., Fletcher, A.G., Mirams, G.R., Murray, P., Osbourne, J.M., Walter, A., Chapman, J., Garny, A., van Leeuwen, I.M.M., Maini, P.K., Rodriguez, B., Waters, S.L., Whiteley, J.P., Byrne, H.M., Gavaghan, D.J., 2009. Chaste: a test-driven approach to software development for biological modelling. Computer Physics Communications 180, 2452–2471.

Priebe, L., Beuckelmann, D.J., 1998. Simulation study of cellular electric properties in heart failure. Circ Res 82, 1206.

Sakmann, B.F.A.S., Spindler, A.J., Bryant, S.M., Linz, K.W., Noble, D., 2000. Distribution of a persistent sodium current across the ventricular wall in guinea pigs. Circulation Research 87, 910.

Scholz, S.B., 2003. Single Assignment C: efficient support for high-level array operations in a functional setting. Journal of Functional Programming 13, 1005–1059.

Shannon, T.R., Wang, F., Puglisi, J., Weber, C., Bers, D.M., 2004. A mathematical treatment of integrated Ca dynamics within the ventricular myocyte. Biophys J 87, 3351–3371.

Sicouri, S., Antzelevitch, C., 1991. A subpopulation of cells with unique electrophysiological properties in the deep subepicardium of the canine ventricle. the M cell. Circulation Research 68, 1729.

Smith, N.P., Crampin, E.J., Niederer, S.A., Bassingthwaighte, J.B., Beard, D.A., 2007. Computational biology of cardiac myocytes: proposed standards for the physiome. J Exp Biol 210, 1576–1583.

Sun, L., Fan, J., Clark, J., Palade, P., 2000. A model of the L-type Ca2+ channel in rat ventricular myocytes: ion selectivity and inactivation mechanisms. The Journal of Physiology 529, 139–158.

Ten Tusscher, K., Noble, D., Noble, P.J., Panfilov, A.V., 2004. A model for human ventricular tissue. Am J Physiol Heart Circ Physiol 286, 1573–1589.

Ten Tusscher, K., Panfilov, A.V., 2006. Alternans and spiral breakup in a human ventricular tissue model. Am J Physiol Heart Circ Physiol 291, 1088–1100.

Winslow, R.L., Greenstein, J.L., 2011. Cardiac myocytes and local signaling in nano-domains. Progress in Biophysics and Molecular Biology NUMBER, [note to publisher: please update before print].

Winslow, R.L., Rice, J., Jafri, S., Marban, E., O'Rourke, B., 1999. Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, II: model studies. Circulation Research 84, 571.