

RESEARCH ARTICLE

A demonstration of modularity, reuse, reproducibility, portability and scalability for modeling and simulation of cardiac electrophysiology using Kepler Workflows

Pei-Chi Yang¹, Shweta Purawat², Pek U. leong³, Mao-Tsuen Jeng¹, Kevin R. DeMarco¹, Igor Vorobyov¹, Andrew D. McCulloch⁴, Ilkay Altintas², Rommie E. Amaro³, Colleen E. Clancy^{1*}

1 Department of Physiology and Membrane Biology, Department of Pharmacology, School of Medicine, University of California Davis, Davis, California, United States of America, **2** San Diego Supercomputer Center (SDSC), University of California, San Diego, La Jolla, California, United States of America, **3** Department of Chemistry and Biochemistry, National Biomedical Computation Resource, Drug Design Data Resource (D3R), University of California San Diego, La Jolla, California, United States of America, **4** Departments of Bioengineering and Medicine, University of California, San Diego, La Jolla, California, United States of America

☞ These authors contributed equally to this work.

* ceclancy@ucdavis.edu



OPEN ACCESS

Citation: Yang P-C, Purawat S, leong PU, Jeng M-T, DeMarco KR, Vorobyov I, et al. (2019) A demonstration of modularity, reuse, reproducibility, portability and scalability for modeling and simulation of cardiac electrophysiology using Kepler Workflows. *PLoS Comput Biol* 15(3): e1006856. <https://doi.org/10.1371/journal.pcbi.1006856>

Editor: Herbert Sauro, University of Washington, UNITED STATES

Received: August 21, 2018

Accepted: February 8, 2019

Published: March 8, 2019

Copyright: © 2019 Yang et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All data and code are available on the GitHub: https://github.com/ClancyLabUCD/Workflow_Kepler.

Funding: This work was supported by National Institutes of Health (<https://www.nih.gov>) awarded to CEC (grants U01HL126273, R01HL128537, OT2OD026580 and R01HL128170), National Institutes of Health awarded to ADM (grants R01HL137100, R01HL121754, U01HL122199),

Abstract

Multi-scale computational modeling is a major branch of computational biology as evidenced by the US federal interagency Multi-Scale Modeling Consortium and major international projects. It invariably involves specific and detailed sequences of data analysis and simulation, often with multiple tools and datasets, and the community recognizes improved modularity, reuse, reproducibility, portability and scalability as critical unmet needs in this area. Scientific workflows are a well-recognized strategy for addressing these needs in scientific computing. While there are good examples of the use of scientific workflows in bioinformatics, medical informatics, biomedical imaging and data analysis, there are fewer examples in multi-scale computational modeling in general and cardiac electrophysiology in particular. Cardiac electrophysiology simulation is a mature area of multi-scale computational biology that serves as an excellent use case for developing and testing new scientific workflows. In this article, we develop, describe and test a computational workflow that serves as a proof of concept of a platform for the robust integration and implementation of a reusable and reproducible multi-scale cardiac cell and tissue model that is expandable, modular and portable. The workflow described leverages Python and Kepler-Python actor for plotting and pre/post-processing. During all stages of the workflow design, we rely on freely available open-source tools, to make our workflow freely usable by scientists.

and National Institutes of Health awarded to REA (grants P41GM103426, U01GM111528). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Author summary

We present a computational workflow as a proof of concept for integration and implementation of a reusable and reproducible cardiac multi-scale electrophysiology model that is expandable, modular and portable. This framework enables scientists to create intuitive, user-friendly and flexible end-to-end automated scientific workflows using a graphical user interface. Kepler is an advanced open-source platform that supports multiple models of computation. The underlying workflow engine handles scalability, provenance, reproducibility aspects of the code, performs orchestration of data flow, and automates execution on heterogeneous computing resources. One of the main advantages of workflow utilization is the integration of code written in multiple languages. Standardization occurs at the interfaces of the workflow elements and allows for general applications and easy comparison and integration of code from different research groups or even multiple programmers coding in different languages for various purposes from the same group. A workflow driven problem-solving approach enables domain scientists to focus on resolving the core science questions, and delegates the computational and process management burden to the underlying Workflow. The workflow driven approach allows scaling the computational experiment with distributed data-parallel execution on multiple computing platforms, such as, HPC resources, GPU clusters, Cloud etc. The workflow framework tracks software version information along with hardware information to allow users an opportunity to trace any variation in workflow outcome to the system configurations.

Introduction

Computational modeling and simulation has proven to be a powerful approach to reveal fundamental mechanisms of the cardiac rhythm in both normal and pathological conditions. Recent studies have expanded modeling approaches to the domain of predictive pharmacology, utilizing functional *in silico* approaches to *predict* drug efficacy, screen for drug toxicity, as well as suggest disease-specific therapies [1–11]. Modeling and simulation as an approach has distinct advantages over classical experimental methods, including the potential for high throughput prediction, choice of model complexity best suited for a given problem, and investigation of a range of physiological, pathophysiological and pharmacological parameters. Furthermore, computational modeling and simulation allows for the prediction of overall emergent effects of specific parameter perturbations on the simulated system.

As computational cardiac models have become increasingly accepted as predictive tools, there has been a recent movement towards utilizing them in applied venues, especially in the domain of safety pharmacology [12, 13]. This transition has required a deep and objective assessment of the need for well-defined criteria to allow for the verification, validation, and uncertainty quantification (VVUQ) of models and model predictions [13–15]. In the VVUQ paradigm, *verification* ensures the computational model accurately solves the equations underlying the mathematical model, and that model reproducibility is ensured regardless of implementation environment (i.e. different computing hardware, compilers, and code libraries), *validation* serves as a measure of the extent, to which the model is accurate in representing the quantities of interest (that may be experimental data), and *uncertainty quantification* determines the extent to which the model output is sensitive (or uncertain in response) to variation, error and uncertainty in the model input. In concert with VVUQ considerations, there has been a determined effort to address the overlapping issues of reproducibility, repeatability and

replicability across a variety of computational disciplines via the application of standards [16–19] [14, 15, 20, 21].

CellML and related markup languages like SBML have been utilized to provide a standard, software- and programming language-independent description of the model, which can improve consistency and reproducibility of model description and sharing [22]. No single markup language can represent a full cardiac multi-scale model, although the combination of CellML to describe the ionic model, FieldML (<http://physiomeproject.org/software/fieldml/about>) for describing the field equations and geometry, and SEDML (<https://sed-ml.github.io>) [23–26] for describing the protocol of the numerical experiment, could in principle be combined to allow a full description.

Other tools have also been developed, such as CellML API or OpenCor that can automatically implement model representations in markup languages [27, 28]. In this way, it is possible to generate whole cell ODE model equations from a language independent CellML description of the model. There are some examples of integrated frameworks (OpenCMISS [29, 30], Chaste [31–34], CARP [35]) that can solve multi-scale models that are derived from standardized model descriptions and indeed, Chaste and CARP can both be integrated and utilized in Kepler workflows [36]. Some multi-scale simulations do, however, require the use of a variety of solvers and data sets.

Moreover, reproducibility also requires development of standards for simulation and model implementation [20, 23, 25, 26, 37, 38]. SED-ML is a community effort to standardize modeling protocols, but standardized protocols that integrate or connect multiple models represented in standardized model descriptions either requires customized software or a workflow framework [24, 25, 39, 40]. To date, there are a few tools that support SED-ML (Tellurium, JWS Online, SBW Simulation Tool, CellDesigner, COPASI, iBioSim, bioUML, SED-ED) for a limited number of application domains. We tested here whether a workflow platform such as Kepler could provide a reproducible approach for integrating multi-scale models requiring more than one solver, a reproducible protocol for numerical experimentation and provenance tracking. Indeed, none of the tools described are mutually exclusive and workflows such as the one described in this study can be readily expanded to allow inclusion of code generation from CellML, FieldML and SEDML descriptions [16].

In this study, after careful analysis, we decided to utilize the Kepler scientific workflow management system. This framework enables scientists to create intuitive, user-friendly and flexible end-to-end automated scientific workflows using a graphical user interface. Kepler is an advanced open-source platform that supports multiple models of computation [41, 42]. The underlying workflow engine handles scalability, provenance, reproducibility aspects of the code, performs orchestration of data flow, and automates execution on heterogeneous computing resources. A workflow driven problem-solving approach enables domain scientists to focus on resolving the core science questions, and delegates the computational and process management burden to the underlying Kepler Workflow system [43–46]. Further, scientists can parameterize the workflow and perform large-scale search for optimal values in the parameter space. Leveraging the benefits of a workflow driven approach allows scaling the computational experiment with distributed data-parallel execution on multiple computing platforms, such as, HPC resources, GPU clusters, Cloud etc. The framework gives users flexibility to execute the workflows from command-line or GUI. Due to its large open-source developer community, Kepler has a rich library that contains over 350 ready-to-use processing components called 'actors' that can be easily customized.

There have been a number of developments aimed at solving the specific problems of reproducibility, repeatability and replicability. In the context of the work presented here, 'reproducibility' refers to zero-difference in outcome between two executions (say W1 and W2) of same

workflow (W), when both executions of our workflow W have exact same hardware (H), same software (S), and same initial conditions (P). The Kepler workflow system captures provenance during each execution at multiple levels. The workflow records the workflow parameters, workflow outputs, intermediate data tokens and extracts the hardware system (CPU Cores, Cache, Memory etc.) profile as well. All of this information is recorded in the workflow provenance database. The information includes versions of all the programs. The key components recorded are the version information of the operating system, source code compiler, Python, Kepler, Java and associated source code. The workflow stores this information in Kepler provenance database. The detailed capture of hardware and software environment information enables users to completely reproduce and replicate the results. Our aim is to facilitate the user to setup same initial conditions and hardware environment (if required), and reproduce results in similar fashion. Notably, the definition we use for reproducibility has been described as replicability and repeatability in other descriptions, whereas reproducibility has used to describe an independent reconstruction of the model from the model equations and initial conditions [47–49]. Indeed, despite attempts to develop standard definitions, there is, as yet, no full consensus on the definitions of each term [50].

One of the main advantages of utilization of workflows is that they can integrate code written in multiple languages, allow for variation in application of compilers and can pass information from one code to another. The standardization occurs at the interfaces of the workflow elements (actors) and allows for very general applications and easy comparison and integration of code from different research groups or even multiple programmers coding in different languages for various purposes from the same group.

Kepler workflow elements can be optimized to run on different platforms and compare results (verification), switch code for different models or implementations of the same model and compare results (validation) or run code multiple times with different initial parameters and estimate variation (uncertainty quantification). Also for the reasons above, Kepler workflows are ideally suited for multi-scale modeling due to ability to integrate very different pieces of codes into a workflow and easily parsing input and output parameters between them. Another advantage is that Kepler workflows are easily accessible for non-experts in computational modeling as programming as a detailed knowledge of model inner workings are not needed to run simulations and modify parameters to suit the requirements of the end user.

Here, we present a multi-scale model of cardiac electrophysiology that is executed in the freely available Kepler scientific workflow system [41, 44]. The workflow we present here is a first required step in VVQU by ensuring reproducibility of models through inclusion of provenance information that describes the origin of the model components, referencing to the data, information about any modifications and the associated rationale, as well as the specific components and parameter settings used in each run. We implemented differential equation models of cardiac physiology that automate the execution of simulations with user defined options of outputs from a single cell (0-dimensional), 1 or 2-dimensional tissue, and a pseudo-ECG output, which can be compared to experimental or clinical data.

Many instances of models can be used with varying input parameters, and the models can be linked in the workflows in various ways. For example, single cell models can be linked to an idealized 1-dimensional fiber model, which allows us to compute a signal averaged pseudo ECG that captures temporal and spatial electrical potential gradients of a propagating wave. Another example demonstrates a thousand instances of the single cell model being linked to a 3-dimensional transmural wedge preparation for investigation of ectopic sources. In addition to these multi-modal choices, the framework can also be reused for multispecies comparisons. Users can control a wide range of input parameters from a simplified command-line, or GUI

interface. The workflow is portable and scalable, having the flexibility to run on any platform a user chooses: local workstations, small clusters, or remote HPC resources.

The computational workflow we present here represents a proof of concept of a platform for the robust integration and implementation of a reusable and reproducible cardiac cell and tissue model that is expandable, modular and portable. The detailed checkpointing of version information along with hardware information gives users an opportunity to trace any variation in workflow outcome to the system configurations, when the infrastructure cannot be exactly replicated. In addition to storing in the database, the workflow generates an execution report for each workflow execution that includes the important workflow parameters, input information, software version and hardware system profile.

Methods

Methodological overview of the workflow

A cardiac ventricular electrophysiology modeling and simulation use case:

We present an automated computational workflow ([Fig 1](#)) that can perform simulations to generate user defined instances and configurations of a single-cell cardiac action potential, conduction of a cardiac action potential in a 1-dimensional (1D) or 2-dimensional (2D) tissue representation and generation of a signal average of electrical activity in time and space to represent a pseudo-ECG.

Please access all codes and associated files and attributes via the GitHub link below. The repository contains specific instructions for use of Kepler System with new source codes. The user-manual provides detailed outlines of how to install Kepler, modify workflow parameters, choose the execution platform and get results from the multi-scale cardiac workflow. The user manual can be accessed at the root of the git repository under filename: "UserManual.docx"

https://github.com/ClancyLabUCD/Workflow_Kepler

Here we demonstrate several example scenarios including: (a) Deployment of the workflow for a single-cell simulation to predict a cardiac action potential with a defined set of input parameters, (b) a configuration for a 1-dimensional cardiac tissue simulation, or (c) a 2-dimensional cardiac tissue simulation.

The model formulations for ventricular cells (the Soltis-Saucerman model [51], Morotti-Grandi model [52], or Grandi-Bers model [53] merged with the Soltis-Saucerman model) were implemented in the Kepler workflow. The source code of simulation models has been implemented in C++ and is compiled during the workflow execution using `icc` or `gcc` compiler, depending on the execution platform and compiler availability. Users can use the source code provided by us or attach their custom developed simulation models by editing the workflow parameter "sourceCode." The workflow gives users a choice to select "compilerProgram" parameter. The workflow integrates multistep single-cell (black circle symbol), 1-dimensional (black rectangle symbol) and 2-dimensional (black square symbol) tissue model simulations in a single automated process ([Fig 1](#)). The SingleCell-Sim module includes a sub-workflow that performs single-cell simulation. Likewise, OneD-Sim and TwoD-Sim modules perform 1-dimensional and 2-dimensional tissue model simulations, respectively. The workflow includes user configuration components, simulation components with multiple execution choices and post-processing components for each model.

User configured parameter settings and initial conditions also allow the end user to control simulation constraints for single-cell, 1D and 2D modules (such as Na^+ -blocker concentration; rapid delayed rectifier potassium channel conductance, G_{Kr} , block ratio; ligand (β -blocker isoproterenol) concentration, CaMKII (Ca^{2+} /calmodulin-dependent protein kinase II) activity levels; number of beats and others through workflow parameters. The simulation constraints

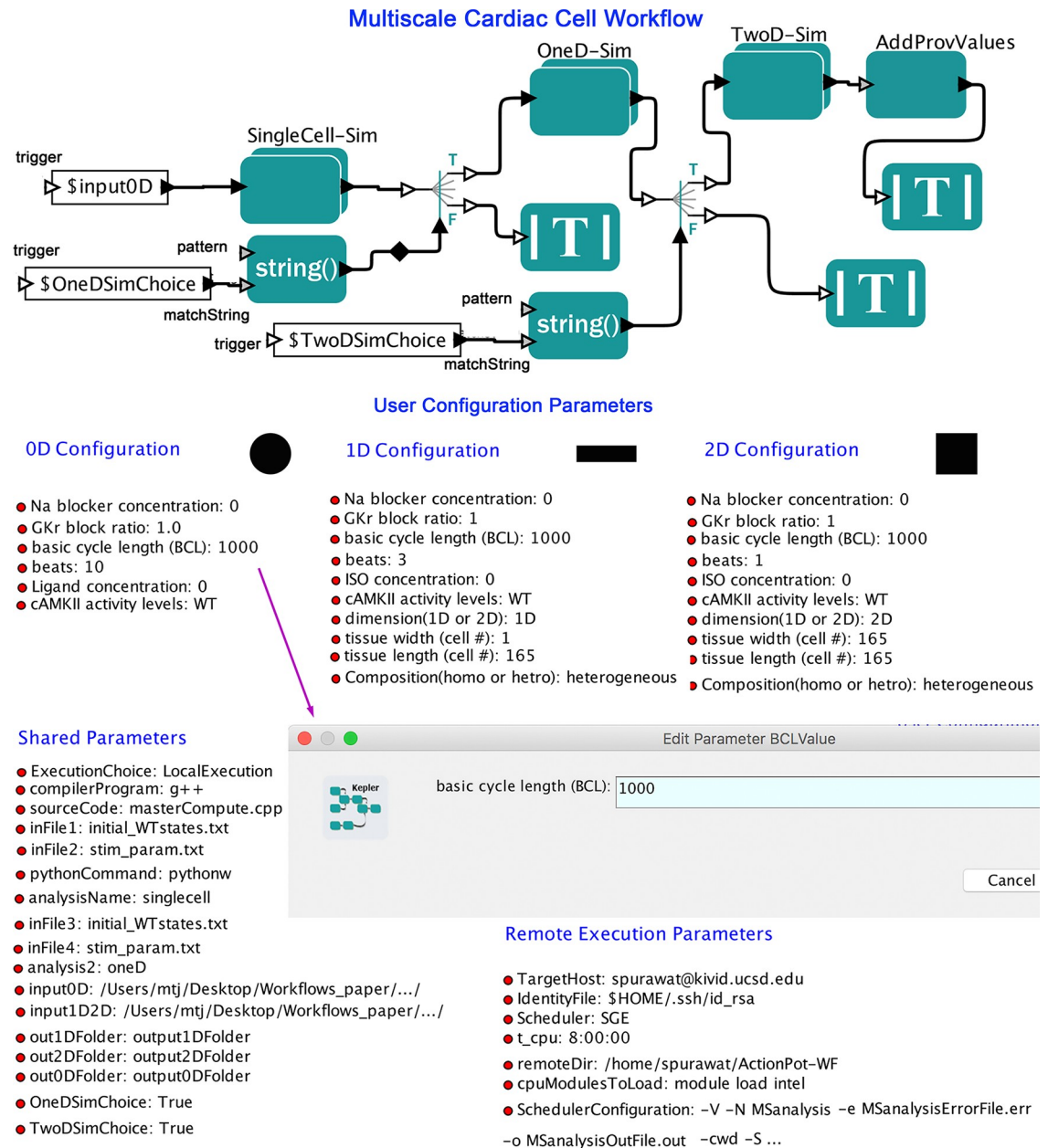


Fig 1. Multiscale cardiac cell workflow. The main interface workflow contains single-cell or zero-dimension (SingleCell-Sim—black circle symbol), one-dimension (OneD-Sim—black rectangle symbol), and two-dimension (TwoD-Sim—black square symbol) tissue simulation modules, as well as user configuration parameters.

<https://doi.org/10.1371/journal.pcbi.1006856.g001>

are ported as workflow parameters, which can be modified and passed to the simulation models using the user configuration module. This workflow module is implemented using Kepler Python actor and Python libraries. Users can seamlessly configure the simulation parameters simply by changing workflow parameter values through command line or GUI as shown in Fig 1 (purple arrow). Many instances of these models can be used with varying input parameters, and the models can be linked in the workflows in various ways. The internal structure of the workflow element (actor) is shown in Fig 2. User parameter configurations can also be expanded to include more parameters by modifying a workflow actor.

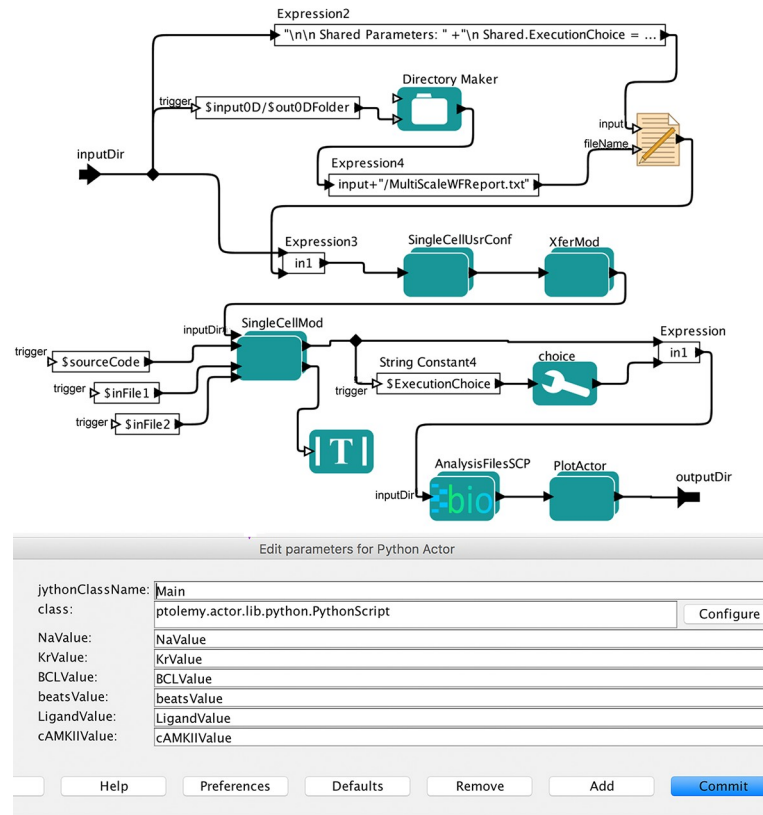


Fig 2. User configuration module. The module “SingleCellUserConf” is the User Configuration Module for single-cell simulations (SingleCell-Sim, see Fig 1). We created the user configuration module to allow users to control simulation constraints such as Na^+ and rapid delayed rectifier K^+ channel (I_{Kr}) blocker concentrations, ligand (β -blocker isoproterenol) concentration, CaMKII activity levels, number of beats, basic cycle length (BCL) i.e. heartbeat duration and other controls for single-cell simulation through workflow parameters.

<https://doi.org/10.1371/journal.pcbi.1006856.g002>

Multiple execution choices

The workflow incorporates flexibility for the end-user’s choice of platform depending on the use case and resource availability. Users can run the workflow on multiple computing platforms such as local, private clusters, and remote HPC clusters by configuring execution choice parameters for individual processes. Kepler allows customization of each execution instance of a workflow with user input parameters. In Fig 3, the Kepler’s Execution Choice actor was created in the Core single-cell module. The Local Execution Options and the Remote execution options are also available in the options menu at the top of the GUI. The capability of multiple execution choice on different hardware platforms is achieved by using the Kepler workflow system. By design, the Kepler framework is capable of automatically creating new jobs for execution. This functionality enables scientists to change execution platforms (local or remote) without any additional user scripting.

Post-processing and visualization module

The post-processing module generates output data files from single-cell, 1D and 2D tissue simulation results. The workflow uses Python libraries and Kepler actors to post process the

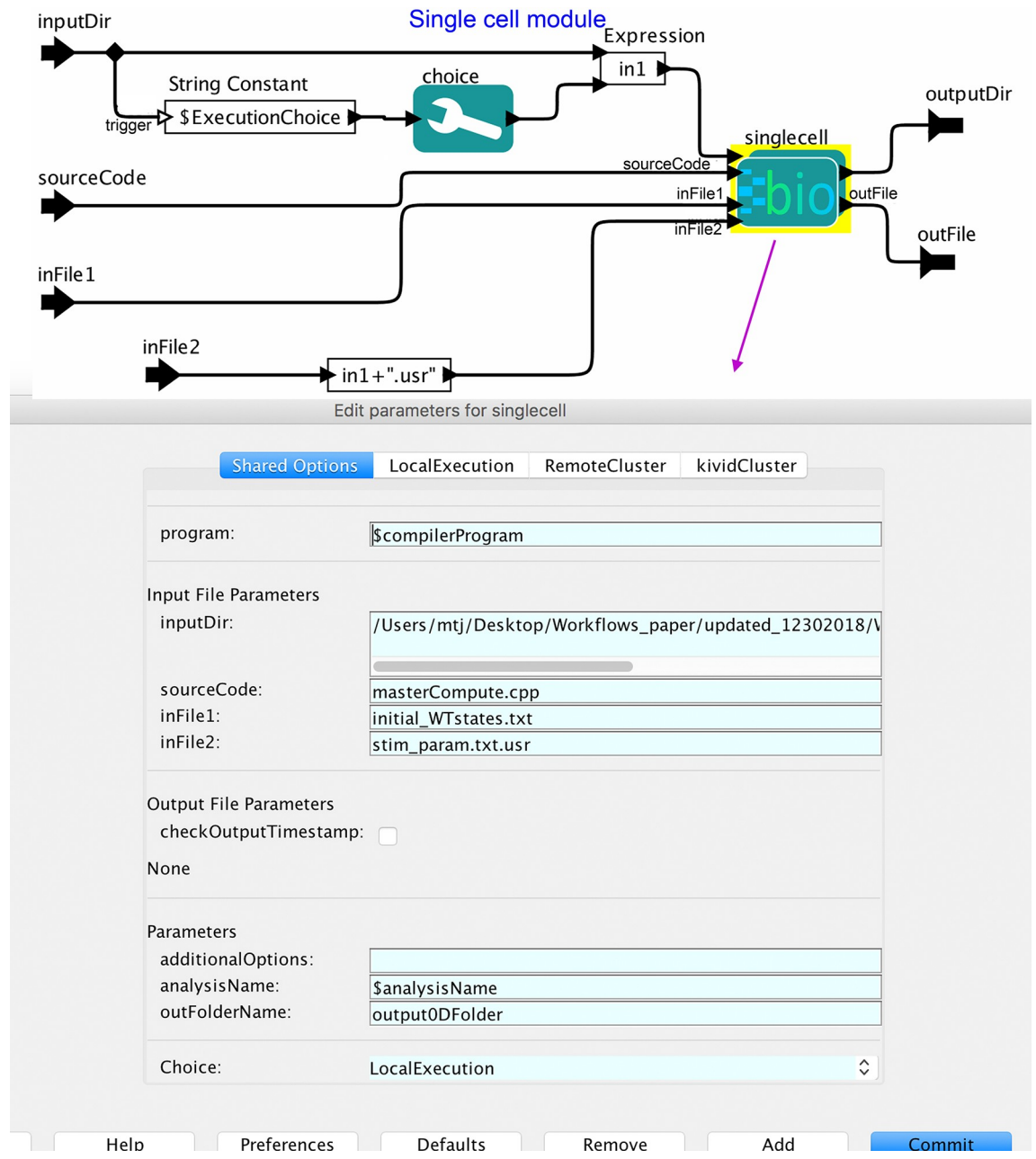


Fig 3. Multiple execution module. The “SingleCellMod” module is the Core Simulation Module for single-cell simulations (SingleCell-Sim, see Fig 1). The Kepler’s Execution Choice actor was created to provide user options of multiple computing platforms based on the use case, data size and the resource availability.

<https://doi.org/10.1371/journal.pcbi.1006856.g003>

simulation results and generate plots for simulated action potentials (AP), main ionic currents (I_{Ca} , I_{Kr} , I_{K1} , I_{NCX} , I_{Na} , I_{to} , I_{Ks}), intracellular (cytosolic) and sarcoplasmic reticulum concentrations of Ca^{2+} and Na^{+} in single cells, pseudo ECG in a 1D-simulation, and snapshots of AP propagation in 2D tissue.

Further, the Kepler workflow automates the provenance collection, execution report generation and reproducibility. For basic execution of the workflow in “as-is” condition, users do

not need expertise in the technologies used, and can execute the workflow using GUI and command line.

Methods for cardiac simulations executed in this study

All simulations of three cardiac myocyte models (the Soltis-Saucerman model [54], Morotti-Grandi model [52], or Grandi-Bers model [53] merged with the Soltis-Saucerman model) were encoded in C/C++, and run using GCC compiler on Mac Pro or Linux computers.

The numerical method used for updating the voltage was forward Euler. Single cell action potentials (APs) and selected ionic currents were recorded. For higher dimension simulations, we simulated a transmural fiber composed of 165 ventricular cells ($\Delta x = \Delta y = 100 \mu\text{m}$) connected by resistances to simulate gap junctions [55]. The transmural fiber contains an endocardial region and epicardial region with a linear decreased in APD as indicated by experimental data [56, 57]. G_{Kr} was used as the index value of endocardium in the cell #1, and the index value of epicardium in cell #165. We can simulate a heterogeneous 2D cardiac tissue composed of 165 by 165 cells with $\Delta x = \Delta y = 100 \mu\text{m}$. The tissue contains an endocardial region and epicardial region with a linear decreased in APD as indicated by experimental data [56, 57]. Channel conductance and gap-junction parameters are same as in the one-dimensional simulations. Current flow is described by the following equation:

$$\frac{\partial V(x, y, t)}{\partial t} = D_x \frac{\partial^2 V(x, y, t)}{\partial x^2} + D_y \frac{\partial^2 V(x, y, t)}{\partial y^2} - \frac{I_{ion} - I_{stim}}{C_m}$$

Where V is the membrane potential, x and y are distances in the longitudinal and transverse directions, respectively, D_x and D_y are diffusion coefficients in the x and y directions, C_m is membrane capacitance ($C_m = 1$). I_{stim} is 180 mA/cm^2 for the first 0.5 ms. We also incorporated anisotropic effects by setting D_x and D_y such that the ratio of conduction velocities is 1:2 [58].

Pseudo-ECG computation. Extracellular unipolar potentials (Φ_e) generated by the fiber in an extensive medium of conductivity σ_e , were computed from the transmembrane potential V_m using the integral expression as in Gima and Rudy [59]:

$$\Phi_e(\mathbf{x}') = \frac{a^2 \sigma_i}{4\sigma_e} \int (-\nabla V_m) \cdot \left[\nabla \frac{1}{r} \right] d\mathbf{x}$$

$$r = [(x - x')^2 + (y - y')^2]^{1/2}$$

Numerical results were visualized using Matplotlib from Python. The workflow requires users to install Kepler 2.5, bioKepler 1.2, Matplotlib, GCC version 4.2.1. Please see Action Potential Workflow User Manual for details.

Results

Modeling and simulation in the workflow

One of the key added advantages of using Kepler Workflow system is the ability to deploy new source code easily. To facilitate execution of new cardiac cell models, the path to C++ source code file is parametrized in the Kepler Workflow. If a scientist wants to use their customized cardiac cell model, she/he can edit the Kepler Workflow parameter called 'sourceCode' under the category of 'SharedParameters', to point to the directory where the desired C++ source code resides. Further, the parameters unique to a given source code can be defined in a file called 'stim_param.txt'. The last step in parametrization is to add a placeholder in the Kepler user interface using the 'Parameter' option under the 'Workflow Input' menu.

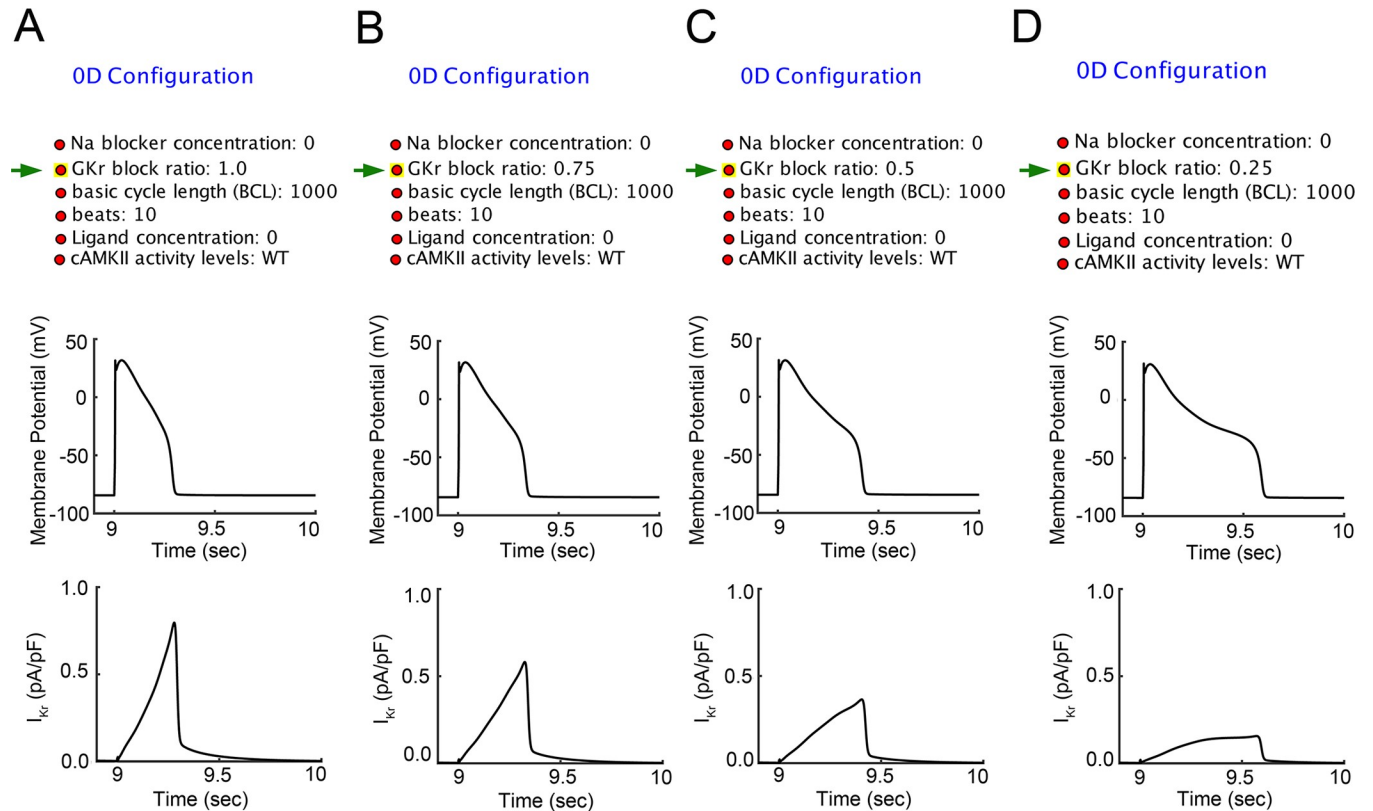


Fig 4. Simulated human ventricular single-cell model via Kepler workflow with or without I_{Kr} reduction. A. User configurations are shown in the left panels. Simulated single-cell action potential (AP—middle panel), and the time course of original I_{Kr} (control) during the AP (bottom). B–D Simulated single-cell action potentials (middle) with reduced I_{Kr} (bottom). G_{Kr} block ratio was set to 0.75 (B), 0.5 (C) and 0.25 (D) in 0D configuration setting (green arrows—left panels), corresponding to 25, 50 or 75% I_{Kr} block, respectively.

<https://doi.org/10.1371/journal.pcbi.1006856.g004>

We first demonstrate the potential for the Kepler workflow environment to be used to run batch simulations for a simulated human ventricular single-cell model for varying degrees of I_{Kr} reduction (Fig 4). The workflow allows users vary rapid delayed rectifier potassium channel conductance, G_{Kr} , in the simulations. Fig 4 illustrates single-cell APs and the time-course of I_{Kr} through the Kepler workflow with varying G_{Kr} . In the top of panels A–D, various end user configurations for input parameters are shown for each simulation instance. In the middle row, simulated single-cell action potentials are shown. In the bottom row, the time-course of I_{Kr} during the AP is shown. The G_{Kr} was reduced via the indicated (green arrows—top panels) block ratios of 1 (used as control, panel A), 0.75 (B), 0.50 (C) and 0.25 (D), corresponding to I_{Kr} block of 0, 25, 50 and 75%, respectively.

In Fig 5, we demonstrate expansion of the workflow beyond single-cell simulation to user defined 1D and 2D-simulations. The workflow generates a single cell cardiac ventricular action potential (Fig 5A), as well as a one-dimensional simulation and a pseudo-ECG (Fig 5B) and then ingests steady-state results from 1D simulations to seed 2D simulations shown in Fig 5C. In this example, the single cell was simulated at a pacing rate of 1 Hz and 10 action potentials were generated. The last AP (10th beat) is shown in Fig 5A (bottom panel). In the tissue simulations, we simulated a heterogeneous fiber (with a linear decrease in AP duration from endocardial to epicardial region [57] (i.e. from the innermost to the outer layer of the cardiac tissue) composed of 165 ventricular cells (parameter tissue length = 165 cells in Fig 5B, top) for three beats. The pseudo-ECG is shown in Fig 5B (bottom panel). In the panel C, we

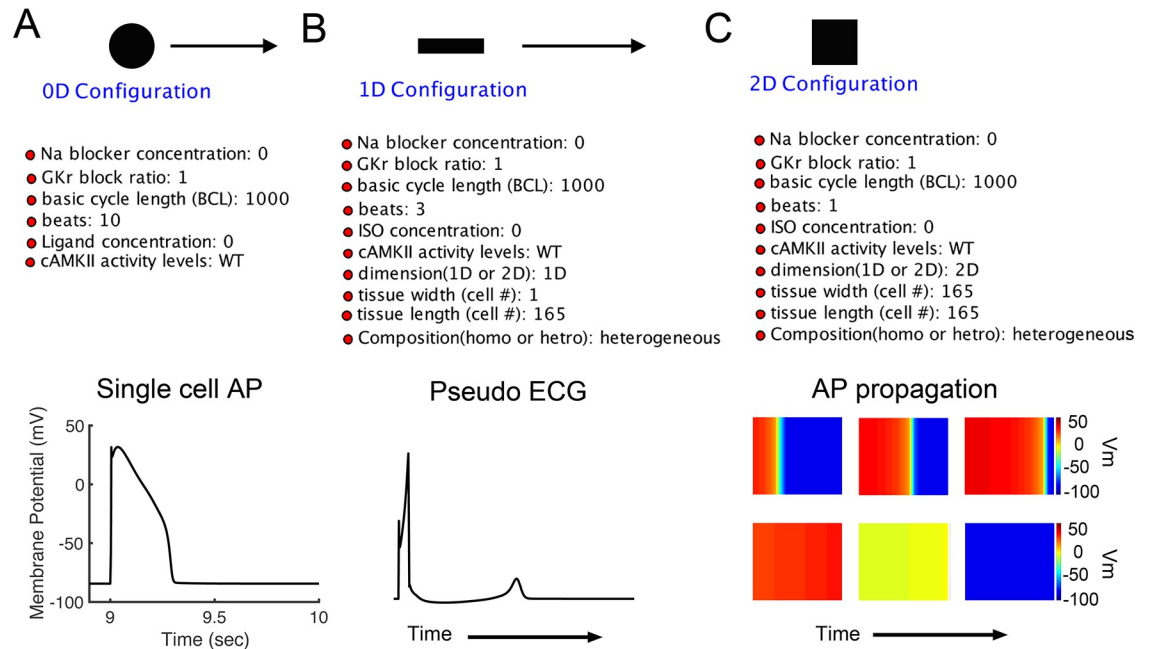


Fig 5. The workflow integrates multistep 0D (single-cell), 1D and 2D model simulations in a single automated process. A. Simulated results of single-cell action potential (bottom panel). **B.** The pseudo ECG (bottom panel), and **C.** Six distinct time snapshots of 2D action potential wave propagation (bottom panel) from endocardial region (left) to epicardial region (right), with a linear decrease in APDs.

<https://doi.org/10.1371/journal.pcbi.1006856.g005>

demonstrated 2D AP wave propagation in response to one stimulus (a planar wave). The workflow simulated a heterogeneous 2D cardiac tissue composed of an array of 165 cells by 165 cells (1.65 cm x 1.65 cm) [57].

In the example shown in Fig 6, we tested three different species models and performed simulations to generate propagation of an action potential in one dimension using the topology shown in Fig 6 (top). Pseudo-ECGs are shown in response to seven stimuli at 1 Hz in Human (Fig 6A—orange), Rabbit (Fig 6B—purple) and Mouse (Fig 6C—green). This example demonstrates how the workflow cyberinfrastructure can also be re-used as a multi-species simulator by utilizing single cell cardiac ventricular computer models as inputs into the higher dimensional models. The cell model of choice can be linked to an idealized one-dimensional fiber model, which can be used to compute signal averaged pseudo ECG traces (Fig 6A–6C). They capture temporal and spatial gradients of electric potential during a simulation that tracks conduction and repolarization of a propagating wave.

Discussion

There has been a tremendous increase in both the number of cardiac models in existence, and in model complexity over the last several decades, correlating with both an increase in computational power, and dramatically reduced computational cost. These developments have created the potential for cardiac cell models and their mathematical and/or agent-based model components to be reused and coupled with one another, creating flexible, modular, portable and potentially scalable models that can account for a range of attributes [60]. The potential for linking models together in new ways also suggests construction of multi-scale models from existing models at various temporal and spatial scales.

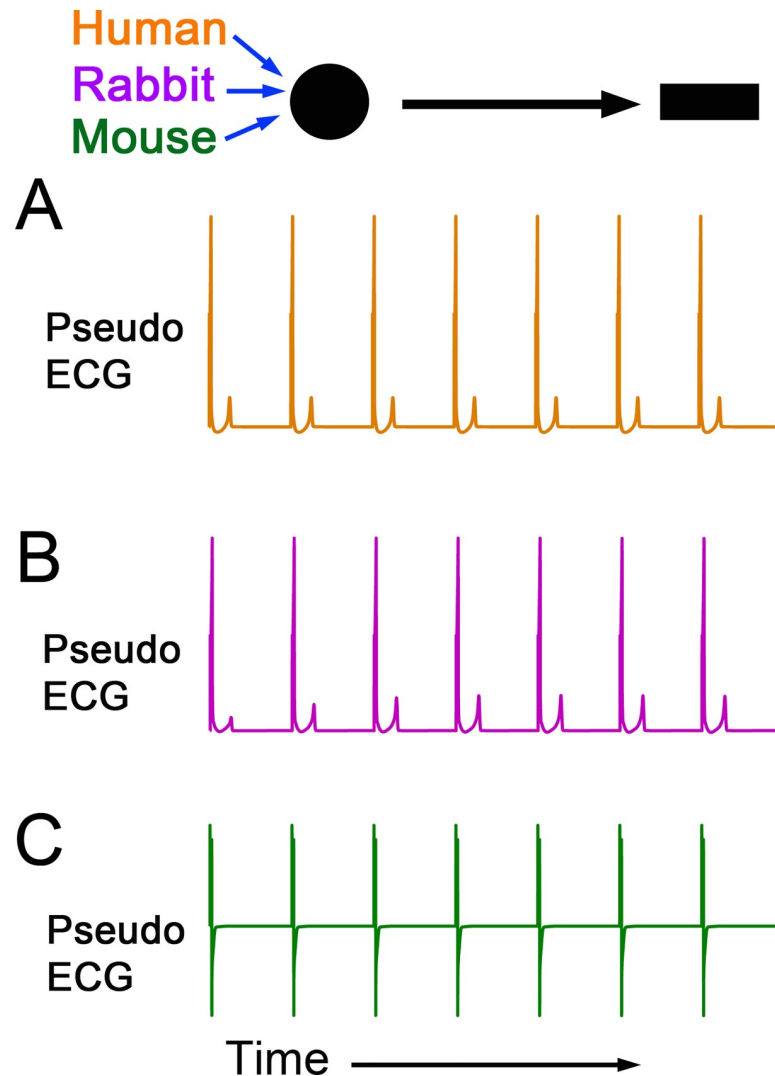


Fig 6. Simulated multispecies ventricular single-cell (0D) and 1D simulations depending on user choice. The multiscale cardiac cell modeling workflow presents from 0-Dimension (circle) to 1-Dimension (rectangle). A–C. Pseudo-ECGs were computed in the Human (orange), Rabbit (purple) and Mouse (green) ventricular cell models.

<https://doi.org/10.1371/journal.pcbi.1006856.g006>

To ideally enable model modularity, reuse, reproducibility, portability and scalability, a model execution platform should be able to provide the reuse of code, reproduction of reported *in silico* predictions, as well as a way to run simulations in an efficient, expandable, modular and portable manner. Scientific workflow tools allow exactly these elements and can provide a user interface and potential for automation and optimization of software and hardware elements of the model execution. Workflows derive from the concept of directed graphs with individual nodes that represent discrete computational components that can be optimized to execute on distinct hardware architecture [61–64]. A scientific workflow is conceptualized as a set of tasks performed on a collection of datasets. The workflow-based design enables scientists to break large computational tasks into smaller manageable and reusable modules (nodes). The data flows through these modules (nodes) and gets transformed. The scientists can collaborate effectively on a large-scale problem by bringing their expertise to different modules in a workflow. Data and results flow between the individual nodes.

The computational overhead involved with workflow implementation is during the start of the Kepler Workflow Engine, and the added cost of building the workflow graph in the Kepler GUI. However, this is a one-time cost during a single execution. Once the Kepler Workflow Engine is up and running, the real advantage comes from automated execution, automated provenance collection, and parameterization driven extensibility benefits. In essence, the end user will get these benefits at least, and these can be enhanced by creation of a wrapper mechanism using the Kepler system, that enhances modularity, shareability and extensibility of their work. Wrapping with Kepler Workflow system enhances the portability of source code—it can work on local machine, or on a distributed cluster—so users are not required to modify or write any script for change in execution platforms. Moreover, diversity of parameters can be handled at two levels, when designing the wrapper workflow. The parameters common to an application area, can be abstracted away and customized at the workflow level (indicated by purple arrow in the Fig 1). The parameters unique to a given source code can be defined in a file called ‘stim_param.txt’ and the user needs to add a placeholder (Fig 1 –User configuration Parameters) in the workflow definition using the ‘Parameter’ option under the ‘Workflow Input’ menu.

The added cost of Kepler workflow system can be hedged by exploiting potential to parallelize codes across distributed systems, for problems involving large-scale computation and large datasets. The Kepler system has inbuilt mechanisms to quickly divide and conquer large computations in batch parallel computations. For cases involving specific cost benefit analysis, since the computational overhead of Kepler is dependent on each workflow, we suggest performing case-specific measurement of ‘Kepler + SourceCode + Parallelization Director’ against isolated run of ‘SourceCode’. We are happy to provide support for such efforts, using our support team for open source users of Kepler.

One critical feature of the Kepler that was decisive in selecting this engine, is the “provenance module”. This module archives workflow execution history, parameters, software and hardware signatures. Workflows Provenance can help preserve evidence and data from experiments to achieve reproducibility [43, 65, 66]. The Kepler reporting module generates informative and detailed summaries of the execution that include user configuration parameters used during the execution in various simulation steps, version of respective software tools, and system hardware information on which the workflow is executed. This “execution-signature” can drastically reduce time required to write reports or methods and material section in scientific publications, enabling domain experts to focus their energy on problem solving [43, 45, 65, 66]. Use of Kepler enabled us to delegate these critical components to the framework, and allowed us to focus on the science behind the problem.

It is important to note that workflow frameworks are not an alternative to markup languages for model description or simulation experimentation or an alternative to specialized packages that can integrate more than one kind or scale of model, but rather than efficient and reproducible approach to multi-scale modeling using multiple component models, software tools and data sets that facilitates usability, sharing and provenance tracking.

Here, we demonstrated the application of the freely-available Kepler scientific workflow system to execute a multi-scale model of cardiac electrophysiology. The workflow allows for modularity, scalability and flexibility in a deployable framework that can be configured by the end-user for maximum flexibility. Like most computational scientists, we have long shared concerns about the reproducibility and reuse of models. Versioning and provenance information can be included in Kepler workflow approach as well as the origin of the model components and user defined components and parameter settings used in each run. In this demonstration, we utilized Kepler to develop a workflow containing differential equation models of cardiac physiology that automate the execution of simulations with user defined

options of outputs from a single cell (0-dimensional), 1 or 2-dimensional tissue, and a pseudo-ECG output, which can be compared to experimental or clinical data.

The workflow as presented could be readily adopted and expanded for applied use in the safety pharmacology domain. In both clinical and experimental settings, prolongation of the QT interval of the ECG and related proarrhythmia have been so strongly associated, that a prolonged QT interval is largely accepted as surrogate marker for proarrhythmia. Here we demonstrate how the workflow can be applied to an investigation of the impact of perturbation of the key repolarizing potassium current in the heart, the rapidly activating component of the delayed rectifier potassium current, I_{Kr} . Mutations in the potassium channel gene encoding I_{Kr} or drug-induced inhibition of I_{Kr} can lead to inherited or acquired long QT syndrome. The QT interval is a phase of the cardiac cycle that corresponds to action potential duration (APD) including cellular repolarization (T-wave). Our single-cell examples demonstrate that reduction of I_{Kr} caused AP prolongation (Fig 4). In Fig 5, the workflow can be used to predict QT intervals in the setting of 1-dimensional tissue or further investigate repolarization phases on 2D AP propagation maps by modifying I_{Kr} . Finally, in Fig 6 our Kepler workflows allow to easily demonstrate that cardiac electrical signal propagation varies in different species used in experimental studies. And using this approach we can relate findings from animal model studies and correlate them to clinical human studies as well.

While considerable attention has been given to the prospects of computational modelling and simulation as a platform for prediction of cardiac drug safety, electro-toxicity and proarrhythmia risk assessment, less scrutiny over the choice of model and the impact of model choice on predicted effects has been given. Here we also show how the Kepler multi-scale workflow can be applied to multispecies to allow users to perform preliminary assessments in models for which predetermined selections of validation experiments can be performed.

The Kepler cyberinfrastructure enables biomedical scientists to (1) understand and catalog accuracy for assembly and linking of models through rigorous uncertainty quantification (UQ) and sensitivity analysis, (2) define a common practice and methodology for linking together (big) data and high-throughput, multi-spatial, multi-temporal, and complex models through reusable workflow definitions, execution, and tools, (3) develop a user interface building toolkit, and (4) develop new methods for deployment and distribution of highly scalable, portable, expandable and robust software and platforms. An additional benefit of this approach is that it allows for individual workflow elements to be optimized for hardware to maximize efficient parallel computing. Various processes of the workflow can be distributed to execute on optimized systems and then pass data through linkage between the workflow elements.

In the near future, our next steps will include the development of an online training course package with lecture material, videos and hands-on on this Multi-scale Cardiac Workflow tool on the e-learning platform called Biomedical Big Data Training and Collaborative (BBDTC) as our educational and community outreach efforts. The BBDTC (<https://biobigdata.ucsd.edu>) is a community-oriented platform that encourages collaborative efforts on training and education to ensure high-quality knowledge dissemination to biomedical big data scientific community. The BBDTC provides easy and intuitive interface to create, launch and share open training materials and tools for biomedical community [42, 67].

Future plans also include goals to integrate this workflow with our Machine Learning based performance prediction module to efficiently schedule different components of the workflow on available computing hardware in a way to gain performance and resource optimization [68–70]. We will couple the workflow with our provenance-based fault tolerance framework to automatically detect failure point and re-start the execution of the workflow from point of failure to save time and resources [71].

In summary, we have developed a Kepler based workflow for multi-scale cardiac electrophysiology that can be utilized and expanded for any number of predictions as defined by the end user. The approach brings us closer to the increasingly shared goal of computational scientists to enable model modularity, reuse, reproducibility, portability and scalability. The workflow concept also allows a model execution platform that allows the reuse of code, reproduction of reported *in silico* predictions, as well as a way to run simulations in an efficient, expandable, modular and portable manner. We have demonstrated an application of the approach by linking models together for construction of multispecies multiscale models from existing models at various temporal and spatial scales.

Author Contributions

Conceptualization: Pei-Chi Yang, Pek U. Jeong, Mao-Tsuen Jeng, Igor Vorobyov, Ilkay Altintas, Rommie E. Amaro, Colleen E. Clancy.

Data curation: Pei-Chi Yang, Pek U. Jeong.

Formal analysis: Pei-Chi Yang, Shweta Purawat, Pek U. Jeong, Igor Vorobyov, Andrew D. McCulloch, Ilkay Altintas, Rommie E. Amaro, Colleen E. Clancy.

Funding acquisition: Andrew D. McCulloch, Ilkay Altintas, Rommie E. Amaro, Colleen E. Clancy.

Investigation: Pei-Chi Yang, Pek U. Jeong, Mao-Tsuen Jeng, Ilkay Altintas, Colleen E. Clancy.

Methodology: Pei-Chi Yang, Shweta Purawat, Pek U. Jeong, Mao-Tsuen Jeng, Kevin R. DeMarco, Igor Vorobyov, Ilkay Altintas, Colleen E. Clancy.

Project administration: Pei-Chi Yang, Pek U. Jeong, Andrew D. McCulloch, Ilkay Altintas, Rommie E. Amaro, Colleen E. Clancy.

Resources: Andrew D. McCulloch, Ilkay Altintas, Rommie E. Amaro, Colleen E. Clancy.

Software: Pei-Chi Yang, Shweta Purawat, Mao-Tsuen Jeng, Ilkay Altintas.

Supervision: Ilkay Altintas, Rommie E. Amaro, Colleen E. Clancy.

Validation: Pei-Chi Yang, Shweta Purawat, Mao-Tsuen Jeng, Andrew D. McCulloch.

Visualization: Pei-Chi Yang, Mao-Tsuen Jeng.

Writing – original draft: Pei-Chi Yang, Shweta Purawat, Kevin R. DeMarco, Igor Vorobyov, Andrew D. McCulloch, Rommie E. Amaro, Colleen E. Clancy.

Writing – review & editing: Pei-Chi Yang, Shweta Purawat, Andrew D. McCulloch, Rommie E. Amaro, Colleen E. Clancy.

References

1. Di Veroli GY, Davies MR, Zhang H, Abi-Gerges N, Boyett MR. High-throughput screening of drug-binding dynamics to HERG improves early drug safety assessment. *Am J Physiol Heart Circ Physiol*. 2013; 304(1):H104–17. Epub 2012/10/30. <https://doi.org/10.1152/ajpheart.00511.2012> PMID: 23103500.
2. Sahli Costabal F, Yao J, Kuhl E. Predicting the cardiac toxicity of drugs using a novel multiscale exposure-response simulator. *Comput Methods Biomech Biomed Engin*. 2018; 21(3):232–46. Epub 2018/03/02. <https://doi.org/10.1080/10255842.2018.1439479> PMID: 29493299.
3. Yang PC, El-Bizri N, Romero L, Giles WR, Rajamani S, Belardinelli L, et al. A computational model predicts adjunctive pharmacotherapy for cardiac safety via selective inhibition of the late cardiac Na current. *J Mol Cell Cardiol*. 2016; 99:151–61. Epub 2016/08/23. <https://doi.org/10.1016/j.yjmcc.2016.08.011> PMID: 27545042; PubMed Central PMCID: PMC45453509.

4. Yang PC, Moreno JD, Miyake CY, Vaughn-Behrens SB, Jeng MT, Grandi E, et al. In silico prediction of drug therapy in catecholaminergic polymorphic ventricular tachycardia. *J Physiol*. 2016; 594(3):567–93. Epub 2015/10/31. <https://doi.org/10.1113/JP271282> PMID: 26515697; PubMed Central PMCID: PMC4784170.
5. Lancaster MC, Sobie EA. Improved Prediction of Drug-Induced Torsades de Pointes Through Simulations of Dynamics and Machine Learning Algorithms. *Clin Pharmacol Ther*. 2016; 100(4):371–9. Epub 2016/03/08. <https://doi.org/10.1002/cpt.367> PMID: 26950176.
6. Shim JV, Chun B, van Hasselt JGC, Birtwistle MR, Saucerman JJ, Sobie EA. Mechanistic Systems Modeling to Improve Understanding and Prediction of Cardiotoxicity Caused by Targeted Cancer Therapeutics. *Front Physiol*. 2017; 8:651. Epub 2017/09/28. <https://doi.org/10.3389/fphys.2017.00651> PMID: 28951721; PubMed Central PMCID: PMC5599787.
7. Gong JQX, Sobie EA. Population-based mechanistic modeling allows for quantitative predictions of drug responses across cell types. *NPJ Syst Biol Appl*. 2018; 4:11. Epub 2018/03/07. <https://doi.org/10.1038/s41540-018-0047-2> PMID: 29507757; PubMed Central PMCID: PMC5825396.
8. Ortega FA, Grandi E, Krogh-Madsen T, Christini DJ. Applications of Dynamic Clamp to Cardiac Arrhythmia Research: Role in Drug Target Discovery and Safety Pharmacology Testing. *Front Physiol*. 2017; 8:1099. Epub 2018/01/23. <https://doi.org/10.3389/fphys.2017.01099> PMID: 29354069; PubMed Central PMCID: PMC5758594.
9. Ellinwood N, Dobrev D, Morotti S, Grandi E. In Silico Assessment of Efficacy and Safety of IKur Inhibitors in Chronic Atrial Fibrillation: Role of Kinetics and State-Dependence of Drug Binding. *Front Pharmacol*. 2017; 8:799. Epub 2017/11/23. <https://doi.org/10.3389/fphar.2017.00799> PMID: 29163179; PubMed Central PMCID: PMC5681918.
10. Gomez JF, Cardona K, Romero L, Ferrero JM, Trenor B. Electrophysiological and Structural Remodeling in Heart Failure Modulate Arrhythmogenesis. 1D Simulation Study. *PLoS One*. 2014; 9(9):e106602. Epub 2014/09/06. <https://doi.org/10.1371/journal.pone.0106602> PMID: 25191998; PubMed Central PMCID: PMC4156355.
11. Passini E, Britton OJ, Lu HR, Rohrbacher J, Hermans AN, Gallacher DJ, et al. Human In Silico Drug Trials Demonstrate Higher Accuracy than Animal Models in Predicting Clinical Pro-Arrhythmic Cardiotoxicity. *Front Physiol*. 2017; 8:668. Epub 2017/09/29. <https://doi.org/10.3389/fphys.2017.00668> PMID: 28955244; PubMed Central PMCID: PMC5601077.
12. Mirams GR, Davies MR, Cui Y, Kohl P, Noble D. Application of cardiac electrophysiology simulations to pro-arrhythmic safety testing. *Br J Pharmacol*. 2012; 167(5):932–45. Epub 2012/05/10. <https://doi.org/10.1111/j.1476-5381.2012.02020.x> PMID: 22568589; PubMed Central PMCID: PMC3492977.
13. Mirams GR, Pathmanathan P, Gray RA, Challenor P, Clayton RH. Uncertainty and variability in computational and mathematical models of cardiac physiology. *J Physiol*. 2016; 594(23):6833–47. Epub 2016/03/19. <https://doi.org/10.1113/JP271671> PMID: 26990229; PubMed Central PMCID: PMC45134370.
14. Pathmanathan P, Gray RA. Ensuring reliability of safety-critical clinical applications of computational cardiac models. *Front Physiol*. 2013; 4:358. Epub 2014/01/01. <https://doi.org/10.3389/fphys.2013.00358> PMID: 24376423; PubMed Central PMCID: PMC3858646.
15. Pathmanathan P, Gray RA. Verification of computational models of cardiac electro-physiology. *Int J Numer Method Biomed Eng*. 2014; 30(5):525–44. Epub 2013/11/22. <https://doi.org/10.1002/cnm.2615> PMID: 24259465.
16. de Bono B, Safaei S, Grenon P, Nickerson DP, Alexander S, Helvensteijn M, et al. The Open Physiology workflow: modeling processes over physiology circuitboards of interoperable tissue units. *Front Physiol*. 2015; 6:24. Epub 2015/03/12. <https://doi.org/10.3389/fphys.2015.00024> PMID: 25759670; PubMed Central PMCID: PMC4338662.
17. Beaulieu-Jones BK, Greene CS. Reproducibility of computational workflows is automated using continuous analysis. *Nat Biotechnol*. 2017; 35(4):342–6. Epub 2017/03/14. <https://doi.org/10.1038/nbt.3780> PMID: 28288103; PubMed Central PMCID: PMC6103790.
18. Vanhaelen Q, Mamoshina P, Aliper AM, Artemov A, Lezhnina K, Ozerov I, et al. Design of efficient computational workflows for in silico drug repurposing. *Drug Discov Today*. 2017; 22(2):210–22. <https://doi.org/10.1016/j.drudis.2016.09.019> WOS:000395223800004. PMID: 27693712
19. Abramson D, Bethwaite B, Enticott C, Garic S, Peachey T, Michailova A, et al. Embedding optimization in computational science workflows. *J Comput Sci-Neth*. 2010; 1(1):41–7. <https://doi.org/10.1016/j.jocs.2010.04.002> WOS:000208807700008.
20. Daly AC, Clerx M, Beattie KA, Cooper J, Gavaghan DJ, Mirams GR. Reproducible model development in the cardiac electrophysiology Web Lab. *Prog Biophys Mol Biol*. 2018; 139:3–14. Epub 2018/05/31. <https://doi.org/10.1016/j.pbiomolbio.2018.05.011> PMID: 29842853.

21. Krishnamoorthi S, Perotti LE, Borgstrom NP, Ajjola OA, Frid A, Ponnaluri AV, et al. Simulation Methods and Validation Criteria for Modeling Cardiac Ventricular Electrophysiology. *PLoS One*. 2014; 9(12): e114494. Epub 2014/12/11. <https://doi.org/10.1371/journal.pone.0114494> PMID: 25493967; PubMed Central PMCID: PMC4262432.
22. Miller AK, Britten RD, Nielsen PM. Declarative representation of uncertainty in mathematical models. *PLoS One*. 2012; 7(7):e39721. Epub 2012/07/18. <https://doi.org/10.1371/journal.pone.0039721> PMID: 22802941; PubMed Central PMCID: PMC3389025.
23. Bergmann FT, Cooper J, Konig M, Moraru I, Nickerson D, Le Novere N, et al. Simulation Experiment Description Markup Language (SED-ML) Level 1 Version 3 (L1V3). *J Integr Bioinform*. 2018; 15(1). ARTN 20170086 <https://doi.org/10.1515/jib-2017-0086> WOS:000431020600009. PMID: 29550789
24. Bergmann FT, Nickerson D, Waltemath D, Scharm M. SED-ML web tools: generate, modify and export standard-compliant simulation studies. *Bioinformatics*. 2017; 33(8):1253–4. Epub 2017/01/04. <https://doi.org/10.1093/bioinformatics/btw812> PMID: 28049131; PubMed Central PMCID: PMC5860579.
25. Waltemath D, Adams R, Beard DA, Bergmann FT, Bhalla US, Britten R, et al. Minimum Information About a Simulation Experiment (MIASE). *Plos Computational Biology*. 2011; 7(4). ARTN e1001122 <https://doi.org/10.1371/journal.pcbi.1001122> WOS:000289973600006. PMID: 21552546
26. Waltemath D, Adams R, Bergmann FT, Hucka M, Kolpakov F, Miller AK, et al. Reproducible computational biology experiments with SED-ML—The Simulation Experiment Description Markup Language. *Bmc Systems Biology*. 2011; 5. Artn 198 <https://doi.org/10.1186/1752-0509-5-198> WOS:000301740900001. PMID: 22172142
27. Garry A, Hunter PJ. OpenCOR: a modular and interoperable approach to computational biology. *Front Physiol*. 2015; 6:26. Epub 2015/02/24. <https://doi.org/10.3389/fphys.2015.00026> PMID: 25705192; PubMed Central PMCID: PMC4319394.
28. Yu T, Lloyd CM, Nickerson DP, Cooling MT, Miller AK, Garry A, et al. The Physiome Model Repository 2. *Bioinformatics*. 2011; 27(5):743–4. Epub 2011/01/11. <https://doi.org/10.1093/bioinformatics/btq723> PMID: 21216774.
29. Nickerson DP, Ladd D, Hussan JR, Safaei S, Suresh V, Hunter PJ, et al. Using CellML with OpenCMISS to Simulate Multi-Scale Physiology. *Front Bioeng Biotechnol*. 2014; 2:79. Epub 2015/01/21. <https://doi.org/10.3389/fbioe.2014.00079> PMID: 25601911; PubMed Central PMCID: PMC4283644.
30. Safaei S, Bradley CP, Suresh V, Mithraratne K, Muller A, Ho H, et al. Roadmap for cardiovascular circulation model. *J Physiol*. 2016; 594(23):6909–28. Epub 2016/08/11. <https://doi.org/10.1113/JP272660> PMID: 27506597; PubMed Central PMCID: PMC45134416.
31. Cooper J, Scharm M, Mirams GR. The Cardiac Electrophysiology Web Lab. *Biophys J*. 2016; 110(2):292–300. Epub 2016/01/21. <https://doi.org/10.1016/j.bpj.2015.12.012> PMID: 26789753; PubMed Central PMCID: PMC4724653.
32. Bernabeu MO, Wallman M, Rodriguez B. Shock-induced arrhythmogenesis in the human heart: A computational modelling study. *Conf Proc IEEE Eng Med Biol Soc*. 2010; 2010:760–3. Epub 2010/11/26. <https://doi.org/10.1109/IEMBS.2010.5626338> PMID: 21095904.
33. Bernabeu MO, Bordas R, Pathmanathan P, Pitt-Francis J, Cooper J, Garry A, et al. CHASTE: incorporating a novel multi-scale spatial and temporal algorithm into a large-scale open source library. *Philos T R Soc A*. 2009; 367(1895):1907–30. <https://doi.org/10.1098/rsta.2008.0309> WOS:000265282200006. PMID: 19380318
34. Pitt-Francis J, Bernabeu MO, Cooper J, Garry A, Momtahan L, Osborne J, et al. Chaste: using agile programming techniques to develop computational biology software. *Philos T R Soc A*. 2008; 366(1878):3111–36. <https://doi.org/10.1098/rsta.2008.0096> WOS:000257921900008. PMID: 18565813
35. Vigmond EJ, Hughes M, Plank G, Leon LJ. Computational tools for modeling electrical activity in cardiac tissue. *J Electrocardiol*. 2003; 36 Suppl:69–74. Epub 2004/01/13. PMID: 14716595.
36. Abramson D, Bernabeu MO, Bethwaite B, Burrage K, Corrias A, Enticott C, et al. High-throughput cardiac science on the Grid. *Philos Trans A Math Phys Eng Sci*. 2010; 368(1925):3907–23. Epub 2010/07/21. <https://doi.org/10.1098/rsta.2010.0170> PMID: 20643684.
37. Bergmann FT, Cooper J, Le Novere N, Nickerson D, Waltemath D. Simulation Experiment Description Markup Language (SED-ML) Level 1 Version 2. *J Integr Bioinform*. 2015; 12(2). ARTN 262 <https://doi.org/10.2390/biecoll-jib-2015-262> WOS:000383955300005. PMID: 26528560
38. Henkel R, Wolkenhauer O, Waltemath D. Combining computational models, semantic annotations and simulation experiments in a graph database. *Database (Oxford)*. 2015; 2015. Epub 2015/03/11. <https://doi.org/10.1093/database/bau130> PMID: 25754863; PubMed Central PMCID: PMC4352687.
39. Adams RR. SED-ED, a workflow editor for computational biology experiments written in SED-ML. *Bioinformatics*. 2012; 28(8):1180–1. Epub 2012/03/01. <https://doi.org/10.1093/bioinformatics/bts101> PMID: 22368254.

40. Kohn D, Le Novere N. SED-ML—An XML Format for the Implementation of the MIASE Guidelines. *Lect N Bioinformat*. 2008; 5307:176–+. WOS:000260924200012.
41. Ludascher B, Altintas I, Berkley C, Higgins D, Jaeger E, Jones M, et al. Scientific workflow management and the Kepler system. *Concurr Comp-Pract E*. 2006; 18(10):1039–65. <https://doi.org/10.1002/cpe.994> WOS:000239804900003.
42. The Kepler project website 2017. Available from: <http://kepler-project.org>.
43. Altintas I, Barney O, Jaeger-Frank E. Provenance collection support in the Kepler Scientific Workflow System. *Provenance and Annotation of Data*. 2006; 4145:118–32. WOS:000241462000014.
44. Altintas I, Berkley C, Jaeger E, Jones M, Ludascher B, Mock S. Kepler: An extensible system for design and execution of scientific workflows. 16th International Conference on Scientific and Statistical Database Management, Proceedings. 2004:423–4. WOS:000222968600054.
45. Crawl D, Singh A, Altintas I. Kepler webview: A lightweight, portable framework for constructing real-time web interfaces of scientific workflows. *international Conference on Computational Science* 6–8 June 2016; San Diego, California 2016. p. 673–9.
46. Wang JW, Crawl D, Altintas I, Li WZ. Big Data Applications Using Workflows for Data Parallel Computing. *Comput Sci Eng*. 2014; 16(4):11–21. <https://doi.org/10.1109/Mcse.2014.50> WOS:000340025700004.
47. Nickerson D, Hunter PJ, editors. *Introducing the Physiome Journal: Improving Reproducibility, Reuse, and Discovery of Computational Models*. P IEEE Int C E-Sci; 2017 24–27 Oct. 2017; Auckland, New Zealand: IEEE.
48. Crook S. M. DAP, Plesser H. E. Learning from the past: approaches for reproducibility in computational neuroscience. In: M. BJ editor. *20 Years of Computational Neuroscience*. 9. New York, NY: Springer; 2013. p. 73–102.
49. McDougal RA, Bulanova AS, Lytton WW. Reproducibility in Computational Neuroscience Models and Simulations. *IEEE Trans Biomed Eng*. 2016; 63(10):2021–35. Epub 2016/04/06. <https://doi.org/10.1109/TBME.2016.2539602> PMID: 27046845; PubMed Central PMCID: PMCPMC5016202.
50. Plesser HE. Reproducibility vs. Replicability: A Brief History of a Confused Terminology. *Front Neuroinform*. 2017; 11:76. Epub 2018/02/07. <https://doi.org/10.3389/fninf.2017.00076> PMID: 29403370; PubMed Central PMCID: PMCPMC5778115.
51. Soltis AR, Saucerman JJ. Synergy between CaMKII substrates and beta-adrenergic signaling in regulation of cardiac myocyte Ca(2+) handling. *Biophysical journal*. 2010; 99(7):2038–47. Epub 2010/10/07. <https://doi.org/10.1016/j.bpj.2010.08.016> PMID: 20923637; PubMed Central PMCID: PMC3042590.
52. Morotti S, Edwards AG, McCulloch AD, Bers DM, Grandi E. A novel computational model of mouse myocyte electrophysiology to assess the synergy between Na+ loading and CaMKII. *J Physiol*. 2014; 592(6):1181–97. Epub 2014/01/15. <https://doi.org/10.1113/jphysiol.2013.266676> PMID: 24421356; PubMed Central PMCID: PMCPMC3961080.
53. Grandi E, Pasqualini FS, Bers DM. A novel computational model of the human ventricular action potential and Ca transient. *J Mol Cell Cardiol*. 2010; 48(1):112–21. Epub 2009/10/20. <https://doi.org/10.1016/j.yjmcc.2009.09.019> PMID: 19835882; PubMed Central PMCID: PMCPMC2813400.
54. Soltis AR, Saucerman JJ. Synergy between CaMKII substrates and beta-adrenergic signaling in regulation of cardiac myocyte Ca(2+) handling. *Biophys J*. 2010; 99(7):2038–47. Epub 2010/10/07. <https://doi.org/10.1016/j.bpj.2010.08.016> PMID: 20923637; PubMed Central PMCID: PMCPMC3042590.
55. Faber GM, Rudy Y. Action potential and contractility changes in [Na(+)](i) overloaded cardiac myocytes: a simulation study. *Biophys J*. 2000; 78(5):2392–404. Epub 2000/04/25. [https://doi.org/10.1016/S0006-3495\(00\)76783-X](https://doi.org/10.1016/S0006-3495(00)76783-X) PMID: 10777735.
56. Lou Q, Fedorov VV, Glukhov AV, Moazami N, Fast VG, Efimov IR. Transmural heterogeneity and remodeling of ventricular excitation-contraction coupling in human heart failure. *Circulation*. 2011; 123(17):1881–90. Epub 2011/04/20. <https://doi.org/10.1161/CIRCULATIONAHA.110.989707> PMID: 21502574; PubMed Central PMCID: PMC3100201.
57. Glukhov AV, Fedorov VV, Lou Q, Ravikumar VK, Kalish PW, Schuessler RB, et al. Transmural dispersion of repolarization in failing and nonfailing human ventricle. *Circ Res*. 2010; 106(5):981–91. Epub 2010/01/23. <https://doi.org/10.1161/CIRCRESAHA.109.204891> PMID: 20093630; PubMed Central PMCID: PMCPMC2842469.
58. Young RJ, Panfilov AV. Anisotropy of wave propagation in the heart can be modeled by a Riemannian electrophysiological metric. *Proceedings of the National Academy of Sciences of the United States of America*. 2010; 107(34):15063–8. <https://doi.org/10.1073/pnas.1008837107> PMID: 20696934; PubMed Central PMCID: PMC2930580.
59. Gima K, Rudy Y. Ionic current basis of electrocardiographic waveforms: a model study. *Circ Res*. 2002; 90(8):889–96. Epub 2002/05/04. PMID: 11988490.

60. Roberts BN, Yang PC, Behrens SB, Moreno JD, Clancy CE. Computational approaches to understand cardiac electrophysiology and arrhythmias. *Am J Physiol Heart Circ Physiol*. 2012; 303(7):H766–83. Epub 2012/08/14. <https://doi.org/10.1152/ajpheart.01081.2011> PMID: 22886409; PubMed Central PMCID: PMC3774200.
61. Deelman E, Gannon D, Shields M, Taylor I. Workflows and e-Science: An overview of workflow system features and capabilities. *Future Gener Comp Sy*. 2009; 25(5):528–40. <https://doi.org/10.1016/j.future.2008.06.012> WOS:000264322900005.
62. Cheng Y, Thalhauser CJ, Smithline S, Pagidala J, Miladinov M, Vezina HE, et al. QSP Toolbox: Computational Implementation of Integrated Workflow Components for Deploying Multi-Scale Mechanistic Models. *AAPS J*. 2017; 19(4):1002–16. <https://doi.org/10.1208/s12248-017-0100-x> PMID: 28540623.
63. Clewley R. Hybrid models and biological model reduction with PyDSTool. *PLoS Comput Biol*. 2012; 8(8):e1002628. <https://doi.org/10.1371/journal.pcbi.1002628> PMID: 22912566; PubMed Central PMCID: PMC3415397.
64. Boekel J, Chilton JM, Cooke IR, Horvatovich PL, Jagtap PD, Kall L, et al. Multi-omic data analysis using Galaxy. *Nat Biotechnol*. 2015; 33(2):137–9. <https://doi.org/10.1038/nbt.3134> WOS:000349198800014. PMID: 25658277
65. Crawl D., Wang J., I A, editors. Provenance for MapReduce-based Data-Intensive Workflows. Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science (WORKS11) at Supercomputing 2011 (SC2011) Conference; 2011: ACM.
66. Wang JW, Crawl D, Purawat S, Nguyen M, Altintas I. Big Data Provenance: Challenges, State of the Art and Opportunities. Proceedings 2015 Ieee International Conference on Big Data. 2015:2509–16. WOS:000380404600315.
67. Purawat S, Cowart C, Amaro RE, Altintas I. Biomedical Big Data Training Collaborative (BBDTc): An effort to bridge the talent gap in biomedical science and research. *J Comput Sci-Neth*. 2017; 20:205–14. <https://doi.org/10.1016/j.jocs.2017.03.010> WOS:000403123400022. PMID: 29104704
68. Singh A, Rao A, Purawat S, Altintas I, editors. A Machine Learning Approach for Modular Workflow Performance Prediction. Proceedings of the 12th Workshop on Workflows in Support of Large-Scale Science; 2017; New York, NY.
69. Singh A, Stephan E, Schram M, Altintas I. Deep Learning on Operational Facility Data Related to Large-Scale Distributed Area Scientific Workflows. IEEE 13th International Conference on e-Science (e-Science). 2017:586–91.
70. Singh A, Stephan E, Elsethagen T, MacDuff M, Raju B, Schram M, et al. Leveraging Large Sensor Streams for Robust Cloud Control. 2016 Ieee International Conference on Big Data (Big Data). 2016:2115–20. WOS:000399115002023.
71. Crawl D, Altintas I. A Provenance-Based Fault Tolerance Mechanism for Scientific Workflows. Provenance and Annotation of Data and Processes. 2008; 5272:152–9. WOS:000262977000015.