POMtool: Software tool for efficient population of models creation

Ossi Noita[1], Olli Ylinen[1], Nicole Anderton[1], Antti Ahola[1], Jussi Koivumäki[1], Jari Hyttinen[1]

[1] Tampere University, Tampere, Finland

Abstract

Building virtual databases of computational cell models using the population of models (POM) approach is becoming a widely adopted in silico procedure, to account for the inherent variability of the cells within the cardiovascular system. Despite its popularity, POM implementations have remained as in-house tools, with no open-source tools available for the community. This is not only inefficient, but leaves significant room for mistakes and misunderstandings. To address this, we introduce `POMtool`*: a Python-based open-source tool to generate and handle the entire process of population generation.* `POMtool` *handles the running of any type of model based on a user defining their model's command-line interface, exposing the model parameters, selecting biomarkers, and selecting valid ranges for calibration.* `POMtool` *also supports user-controlled task partitioning for distributed cluster execution. We demonstrate the use of the* `POMtool` *by creating a population of human induced pluripotent stem cell-devived cardiomyocyte models with Long QT syndrome Type 2 based on our earlier* `hiMCES` *MATLAB model. The* `POMtool` *results are reproducible with a simple configuration file, regardless of the underlying model, the model implementation language, or the computational environment.*

1. Introduction

In vitro investigations form the backbone of understanding biological mechanisms. The throughput of those experiments is limited however, owing to experimental equipment, biological material, and human labor [1]. Computational modeling and simulation, or in silico studies have thus emerged as an important complement to in vitro experiments. A single in silico model, such as a model of cardiomyocyte, represents the average behavior that is fitted to the available in vitro data sets, omitting the inherit variability of biological systems. A population of models (POM) approach addresses that major limitation [2, 3], by creating a database of models that are parameter variations of the initial model. Until very recently, POM studies have been carried out with in-house scripts, while the first openly available software with such functionality, like openCARP [4], are just being rolled out. As the use of in-house developed packages can be labor intensive, a common tool for everyone to use would be beneficial.

Producing population of model requires several steps depending on the starting point. Model fitting is done to bridge gaps of measured phenomena and understood phenomena; we use observations, to make our model fit those observations. In a sensitivity analysis (SA), the affect of each parameter on the model output is analyzed. A SA can either be local, or global. In a local SA, the effect of each parameter is observed individually. This might not be enough, as the parameters might affect each other, to detect this, a global sensitivity analysis is required. In a global SA, the combined changes of multiple parameters are analyzed together to determine the contribution of each parameter [5]. In a POM, first the base model, obtained as described above, is expanded by creating sets of varied internal parameters that inform the initial population. Altering its internal parameters, creating an initial population. Parameter ranges for the initial population is based on the e.g. uncertainty of the parameter values and know limits of the system. After this, the initial population is calibrated to fit the acceptable biomarker output range, generating models that are different in function, but that produce responses that are in the biologically expected range.

There is currently no single tool that can handle the entire POM process as briefly described above. Here, we introduce and demonstrate `POMtool` which is a Python-based open-source tool for producing, calibrating, and sharing POMs as well as model fitting and producing SA.

2. Methods and materials

The development and design procedure used in making the `POMtool` is described in section 2.1, followed by the demonstration implementation procedure in section 2.2.

2.1. POMtool components

`POMtool` has been created to bridge the gap between an exposed model and computational methodology.

In a generalized example of `POMtool`, the computational model f is used to describe biomechanical function,

with

$$\mathbf{x} = (x_i)_{i=1}^{N} \xrightarrow{f} \mathbf{y} = (y_i)_{i=1}^{L} \xrightarrow{\phi} \mathbf{z} = (z_i)_{i=1}^{M} \quad (1)$$

where **x** are input parameters, **y** are outputs, and features (**z**) are extracted from those outputs using function ϕ. The function,

$$g = \phi \circ f, \quad (2)$$

which can be used to map the whole process, from parameters to measured features.

User defined configuration control `POMtool` behavior. The user is required to define f, so that it can be called via the command line. In addition, the user defines the parameters **x** that must to be exposed, and how the **y** can be accessed. The `POMtool` has defined standard ϕ for typical [6, 7], and custom ϕ can be defined by the user, to account for more complex features.

`POMtool` handles **x** generation for optimization, sensitivity analysis through a user defined configuration file. If needed, custom **x** can also be used. By default, results are collected to unique directories, and the models can thus be run simultaneously using multiple machines or threads.

Sobol sensitivity analysis is a global sensitivity analysis approach that can be utilized to determine which parameters are meaningful. It describes first order effects, and total order effect by calculating Sobol indices [8].

Init, two independent matrices A and B are constructed of size $N \times K$

$$\left(A_B^{(i)}\right)_{rj} = \begin{cases} B_{rj}, & j = i, \\ A_{rj}, & j \neq i. \end{cases} \quad (3)$$

For each of the N parameters in the hybrid matrix $\left(A_B^{(i)}\right)_{rj}$ is generated.

This means that each constructed matrix is independently distributed in a similar way, but have a different realization [9]. However, between matrices, only single parameter has been changed. We then calculate variance, and Sobol indexes:

$$S_i = \frac{\frac{1}{K}\sum_{j=1}^{K} g(B)_j \left[g(A_B^{(i)})_j - g(A)_j\right]}{\mathrm{Var}\left(g(A) \cup g(B)\right)} \quad (4)$$

$$S_{T_i} = \frac{\frac{1}{2K}\sum_{j=1}^{K} \left[g(A)_j - g(A_B^{(i)})_j\right]^2}{\mathrm{Var}\left(g(A) \cup g(B)\right)} \quad (5)$$

where S_i describes first order effect, and S_{T_i} tells total effect for x_i [8].

The **Differential evolution** (DE) optimization algorithm is designed for global optimization [10]. As such, it is effective at minimizing the loss function, while avoiding local optimal solutions. DE starts with candidate solutions, and through small iterations evolves the parameters stochastically. Each generation mutates randomly some of the values, and selects either to keep or discard previous sample based on the fitness [10].

Latin hypercube sampling (LHS) is an improvement to random sampling, where selected samples are stratified to fill space completely [11], allowing better coverage of the target space with limited amount of total samples. This can be used it to generate parameters for the population. [11].

Calibration in POM is performed after initial population is created, for example using LHS. User defined biomarker **z** ranges are used get the final POM where each sample is a valid model.

2.2. Demonstration of POMtool

To demonstrate `POMtool` we created a population for mutation Long QT syndrome Type 2 (LQT2) [12,13] using a well established human induced pluripontent cardiomyocyte (hiPSC-Cm) in silico model `hiMCES` [14].

LQT2 is a disease subtype caused by loss-of-function mutation in KCNH2 gene, which, among other things, delays ventricular repolarization [12]. LQT2 is shown to increase action potential duration at 90% repolarization (APD90) by 20.53% [13]. This is a straightforward mutation, chosen to demonstrate the POMtool workflow and capabilities.

Demonstration of the workflow is done in three stages (see Figure 1). Firstly, a sensitivity analysis is performed for the parameters known to be affected by the mutation. This information is then utilized, to select optimization parameters that will be used in fitting the disease model. Lastly, this model is expanded to a POM to have the full LQT2 population.

Our first task is to define our f, which is a MATLAB implementation of `hiMCES` [14], and our ϕ, which is used to acquire our selected biomarker **z** APD90. In addition, the inputs need to be exposed for input modifiers, and the outputs need to be defined to be saved, namely in here , the time and membrane potential that are required to calculate APD90. In this example We expose the Potassium delayer rectifier (I_{Kr}), Sodium channel(I_{Na}), and Sodium Potassium exchanger (I_{NCx}) gains to be modified. The first is chosen for its known affect in the LQT2 [12, 13], and later two are chosen for their direct affect to APD90. We thus expect that in the population, these parameters be altered.

The second task is a SA, which we perform in order to decide between I_{Na} and I_{NCx}. The goal is to pick the parameter for optimization that has a similar effect to the I_{Kr}. We select 128 to be our K, it is large enough for demonstration purposes and quick to generate. For SA, APD90 values corresponding to non-beating simulations are assigned the value 0.

In the third step, base model optimization, we expect to

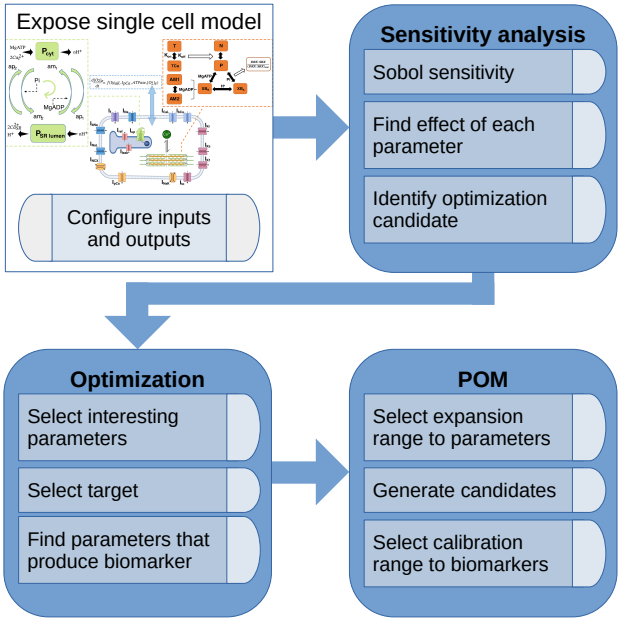

Figure 1. Demonstration workflow: Expose workflow (model as described in [14]), sensitivity analysis, optimization, and population of models

find a fitted disease model, that corresponds to a 20.53% increase in APD90 from the control. In our case, that means from controls 469 ms to 564 ms in LQT2. To fit the model, we use gains selected Sobol SA, with differential evolution algorithm.

In POM, we want to expand all of our parameters, from the fitted disease model, to all in our range. We will use LHS to acquire candidate population of size 1000 expand our initial fitted model. In this demonstration, we chose following expansion ranges I_{Kr} 50% - 150%, I_{Na} 75% - 125%, and I_{NCx} 75% - 125%. Prioritization is given to changes in I_{Kr}, with wider range. Calibration is performed with classifications healthy (APD90 $\in$ [422 ms, 516 ms)) and LQT2 (APD90 $\in$ [516 ms, 621 ms])).

3. Results

This section describes our primary result `POMtool`, and results of demonstration.

3.1. POMtool

`POMtool` has been implemented in Python, and is publicly available at `https://github.com/TAU-CBIG/POMtool` with permissive MIT license. The repository itself contains examples on its use, as well as how its results and processes can be shared and reproduced by other users. It has been designed to be modular and expandable, to meet varied needs of researches.

3.2. Demonstration

With **the Sobol SA** (Figure 2) of the demonstration LQT2 case, we determined that I_{Na} has a similar range to that of I_{Kr}. With a total of 640 runs in the SA, 113 of which did not have spontaneous beating. Out of 128 Sobol rows, there were a total of 80 completely valid rows (with all A, and B and intersection being valid).

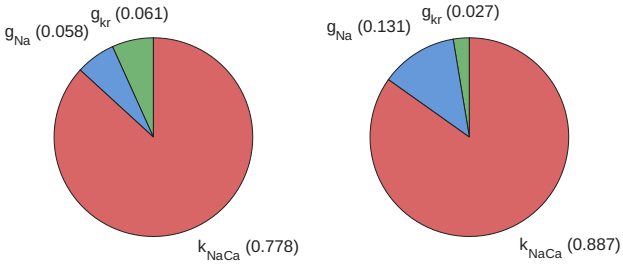

Figure 2. Sobol sensitivity analysis of APD90 sensitivity to parameter changes. S1 (left) shows the first order effects, ST (right) marks the total effect.

For fitting the model to the LQT2 base model, we used channels I_{Kr} and I_{Na}. Changes to I_{NCx} would dominate too greatly, resulting in changes to I_{Kr} being insignificant. Reaching the 7th iteration, we achieved a mean square error 0.0118 with a total of 331 runs. Table 1 shows the result.

Variable	Description	LQT2	POM range
I_{Kr}	K delayer rectifier	0.657	50 % - 150 %
I_{Na}	Na channel	1.001	75 % - 125 %
I_{NCx}	Na K exchanger	1	75 % - 125 %

Table 1. Model parameters with their typical variable name, description, the fitted LQT2 disease model, and POM expansion range

The POM was generated from the LQT2 base model with a LHS of size 1000 x 3 (the ranges for parameters are shown in Table 1). After calibrating the resulting population and displaying both healthy and LQT2 ranges, we obtained 249 models that were within the healthy biomarker range, 471 within the LQT2 biomarker range, and 280 models not representing neither case. The resulting populations are shown in Figure 3.

4. Conclusion and outlook

`POMtool` is available `https://github.com/TAU-CBIG/POMtool` along with the shown demonstration.

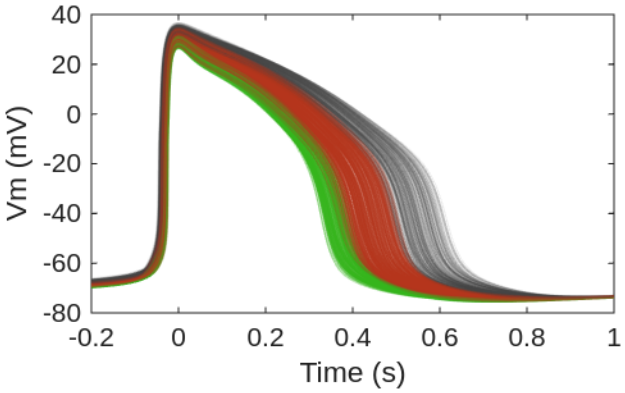

Figure 3. Candidate in silico cells characterized by the action potential in the generated population. Calibrated subpopulations are color coded: green = models with healthy biomarkers, red = LQT2, and gray = discarded as being out of physiological range.

The tool enables researches to run their model with different parameters sets, for the purposes of creating a POM, SA, or fitting their models. It selects parameters, calculates biomarkers, and prunes the populations. `POMtool` additionally enables researchers to sharing and reproduction of work more easily.

`POMtool` will be in active development with permissive open-source license, and welcomes pull requests, issues, and feedback. Usability issues will take priority, as the goal is to make tool efficient to use with as little as possible overhead. It is our hope, that the `POMtool` will become an integral tool on for running models, whether those models are implemented with MATLAB, openCARP, or any custom in-house tool. `POMtool` has integrated all needs for creating a POM and offers standardized format to describe studies where biological variability is introduced to the models.

Acknowledgments

This work is part of the HORIZON-HLTH-2023-TOOL-05 SMASH-HCM project funded by the European Union under Grant 101137115.

References

[1] Macarron R, Banks MN, Bojanic D, Burns DJ, Cirovic DA, Garyantes T, Green DV, Hertzberg RP, Janzen WP, Paslay JW, et al. Impact of high-throughput screening in biomedical research. Nature reviews Drug discovery 2011; 10(3):188–195.

[2] Prinz AA, Billimoria CP, Marder E. Alternative to hand-tuning conductance-based models: construction and analysis of databases of model neurons. Journal of neurophysiology 2003;.

[3] Britton OJ, Bueno-Orovio A, Van Ammel K, Lu HR, Towart R, Gallacher DJ, Rodriguez B. Experimentally calibrated population of models predicts and explains intersubject variability in cardiac cellular electrophysiology. Proceedings of the National Academy of Sciences 2013; 110(23):E2098–E2105.

[4] openCARP consortium, Augustin C, Boyle PM, Chegini F, Colin R, Huppé A, Gerach T, Gsell M, Houillon M, Huang YLC, et al. openCARP, 2026. URL `https://git.opencarp.org/openCARP/openCARP`.

[5] Read MN, Alden K, Timmis J, Andrews PS. Strategies for calibrating models of biology. Briefings in Bioinformatics 2020;21(1):24–35.

[6] Forouzandehmehr M, Koivumäki JT, Hyttinen J, Paci M. A mathematical model of hiPSC cardiomyocytes electromechanics. Physiological reports 2021;9(22):e15124.

[7] Paci M, Passini E, Klimas A, Severi S, Hyttinen J, Rodriguez B, Entcheva E. All-optical electrophysiology refines populations of in silico human iPSC-CMs for drug evaluation. Biophysical journal 2020;118(10):2596–2611.

[8] Saltelli A, Annoni P, Azzini I, Campolongo F, Ratto M, Tarantola S. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. Computer physics communications 2010;181(2):259–270.

[9] Sobol IM. Sensitivity estimates for nonlinear mathematical models. Mathematical Modelling and Computational Experiments 1993;4:407–414.

[10] Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 1997;11(4):341–359.

[11] McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 1979;21(2):239–245.

[12] Ono M, Burgess DE, Schroder EA, Elayi CS, Anderson CL, January CT, Sun B, Immadisetty K, Kekenes-Huskey PM, Delisle BP. Long QT syndrome type 2: emerging strategies for correcting class 2 KCNH2 (hERG) mutations and identifying new patients. Biomolecules 2020;10(8):1144.

[13] Shah D, Prajapati C, Penttinen K, Cherian RM, Koivumäki JT, Alexanova A, Hyttinen J, Aalto-Setälä K. hiPSC-derived cardiomyocyte model of LQT2 syndrome derived from asymptomatic and symptomatic mutation carriers reproduces clinical differences in aggregates but not in single cells. Cells 2020;9(5):1153.

[14] Forouzandehmehr M, Paci M, Hyttinen J, Koivumäki JT. In silico study of the mechanisms of hypoxia and contractile dysfunction during ischemia and reperfusion of hiPSC cardiomyocytes. Disease models mechanisms 2024; 17(4):dmm050365.

Address for correspondence:

Ossi Noita
ossi.noita@tuni.fi