



ISSN: 2744-6204
Published 25 Dec 2022

Computational modeling of anoctamin 1 calcium-activated chloride channels as pacemaker channels in interstitial cells of Cajal

Leyla Noroozbabae^{1*}, David P. Nickerson¹, and Peng Du^{1a}

¹Auckland Bioengineering Institute, University of Auckland, New Zealand

^aNominated from the primary publication authors to be a co-author.

ORIGINAL

Abstract

The system of equations and simulation results presented in Lees-Green et al. (2014) are verified and reproduced in the current paper. To demonstrate reproducibility, we here describe the model encoded in CellML and document the differences between our curated model and that published by Lees-Green et al..

Keywords: interstitial cells of Cajal, computational model, OpenCOR, CellML, ionic current

Curated Model Implementation

<http://doi.org/10.36903/physiome.21760691>

Primary Publications

R. Lees-Green, S. J. Gibbons, G. Farrugia, J. Sneyd, and L. K. Cheng. Computational modeling of anoctamin 1 calcium-activated chloride channels as pacemaker channels in interstitial cells of cajal. *American Journal of Physiology-Gastrointestinal and Liver Physiology*, 306(8):G711–G727, 2014. doi: 10.1152/ajpgi.00449.2013.

OPEN ACCESS Reproducible Model

Edited by
Karin Lundengård

Curated by
Weiwei Ai

**Corresponding author*
l.noroozbabae
@auckland.ac.nz

Submitted 10 Mars 2022

Accepted 4 Dec 2022

Citation
Noroozbabae and
Nickerson (2022)
*Computational modeling of
anoctamin 1
calcium-activated chloride
channels as pacemaker
channels in interstitial cells
of Cajal. Physiome.
doi: 10.36903/phys-
iome.21760691*

1 Introduction

Lees-Green et al. (2014) describes a biophysical computational model of anoctamin 1 (Ano 1) calcium-activated chloride channels (pacemaker channels in interstitial cells of Cajal). The current work involves the mathematical curation of the model in the CellML format (Cuellar et al., 2003), which can reproduce the behaviour of each of the model's variations presented in Lees-Green et al. (2014). Figure 2 in Lees-Green et al. (2014) is not included as it does not represent the introduced model's results. Additionally, some inconsistencies have been identified and are discussed in this work. A persistent workspace for this work is available in the Physiome Model Repository at <https://models.physiomeproject.org/workspace/83c>. The specific version of this implementation used to produce the presented results is included in the OMEX Archive (Bergmann et al., 2014).

This implementation includes the Python source code to generate the model simulations in a 'Simulations' directory. The corresponding CellML implementation is available in a 'Experiments' directory containing the encoded mathematical model. This CellML file on its own does not reproduce the figures due to limited solver capabilities at time of testing. As such, we have elected to describe the required simulation experiments in Python and use the OpenCOR (Garny and Hunter, 2015) Python interpreter to perform the simulations and analyses. In this manuscript, we focus on reproducibility and reusability.

MATLAB (Shampine and Reichelt, 1997) code was obtained from the author of Lees-Green et al. (2014) to reproduce the figures of that paper. While there was no exact correspondence

between equations presented in the original article and the provided MATLAB code, all the figures (corresponding to the model results) were produced, and numeric values were matched with some modifications to a few parameter values; for more details, please see [subsection 3.2](#).

2 Model description

2.1 Primary Publication

The model uses a Hodgkin-Huxley type formulation. The cell membrane lipid bilayer is represented as a capacitance (C_m), and the ion channels in the membrane are represented as conductance. The change in the transmembrane potential (V_m) over time depends on the sum of the individual ion currents through each class of ion channel in the cell, see [Equation 1](#),

$$\frac{dV_m}{dt} = -\frac{I_{tot}}{C_m}. \quad (1)$$

There are 12 different ion conductances in this model (to see the components of the model, see [Table 3](#) and the equations below). We noted that the total ionic current and total flux vary according to the different model variations. For the following model variations: High-Cl(NaV: voltage activated Na Channel), High-Cl(NSV: Voltage Activated Non-Selective Channel), High-Cl(NSCC: Ca²⁺ Activated Non-Selective Channel), and Low-Cl(NaV: Voltage Activated Na⁺ Channel); [Equations 2, 3, 4, and 5](#) represent the total ionic current, respectively. [Equation 6](#) is used to calculate the cytosolic Ca²⁺ concentrations for all the above mentioned model variations.

$$I_{tot}(\text{HCl-NaV}) = I_{SOC} + I_{A_{no1}} + I_{CaT} + I_{BK} + I_{BNa} + I_{NaV}, \quad (2)$$

$$I_{tot}(\text{HCl-NSV}) = I_{SOC} + I_{A_{no1}} + I_{CaT} + I_{BK} + I_{BNa} + I_{NSV}, \quad (3)$$

$$I_{tot}(\text{HCl-NSCC}) = I_{SOC} + I_{A_{no1}} + I_{CaT} + I_{BK} + I_{BNa} + I_{NSCC}, \quad (4)$$

$$I_{tot}(\text{LCI-NaV}) = I_{SOC} + I_{A_{no1}} + I_{CaT} + I_{BK} + I_{BNa} + I_{NaV} + I_{KV} + I_{KERG}, \quad (5)$$

$$\frac{d(Ca_i)}{d(\text{time})} = f_c * (J_{IPR} - J_{SERCA} + J_{SOC} + J_{CaT} - J_{PMCA}). \quad (6)$$

The High-Cl(CaV: Voltage-Activated Ca²⁺ channel) model variation incorporates a long-lasting voltage-gated Ca²⁺ channel (I_{CaV}), so the total ionic current across the membrane is following [Equation 7](#). We used [Equation 8](#) to determine the change in cytosolic Ca²⁺.

$$I_{tot}(\text{HCl-CaV}) = I_{SOC} + I_{A_{no1}} + I_{CaT} + I_{BK} + I_{BNa} + I_{CaV}, \quad (7)$$

$$\frac{d(Ca_i)}{d(\text{time})} = f_c * (J_{IPR} - J_{SERCA} + J_{SOC} + J_{CaT} + J_{CaV} - J_{PMCA}). \quad (8)$$

In the case of the Low-Cl(NSCC: Ca²⁺ Activated Non-Selective Channel), [Equation 9](#) and [Equation 10](#) are used to calculate the total ionic current and the change in cytosolic Ca²⁺, respectively.

$$I_{tot}(\text{LCI-NSCC}) = I_{SOC} + I_{A_{no1}} + I_{CaT} + I_{BK} + I_{BNa} + I_{NSCC} + I_{NCX}, \quad (9)$$

$$\frac{d(Ca_i)}{d(\text{time})} = f_c * (J_{IPR} - J_{SERCA} + J_{SOC} + J_{CaT} - J_{PMCA} - J_{NCX}). \quad (10)$$

The model constituents vary according to the different model configurations, in regard to chloride concentrations: high chloride environment ($E_{Cl} = -20.2$ mV, $C_{Cl} = 78$ mM), and low chloride environment ($E_{Cl} = -49.7$ mV, $C_{Cl} = 25.85$ mM). See [Table 3](#) for details on the model configurations.

2.2 Model Implementation

The implementation of the model was performed using CellML (Cuellar et al., 2003). Some simulation values stated in Lees-Green et al. (2014) resulted in different model behaviour. We adjusted them as outlined in [Table 1](#). The simulation experiments presented here were performed using OpenCOR (Snapshot 2021-10-5).

The simulation experiments in [Lees-Green et al. \(2014, Figures 4 & 5\)](#) were produced in various stages corresponding to the model variations, as summarised in [Table 3](#). We implemented the model configuration for each variation using the related parameters input file for each. For example, to produce the data related to the model variation 'H-Cl (NaV)', we load the model configuration file 'HCl-NSV.json' and we choose the CSV files (Fig4-1a & Fig4-1b) to hold the data. For more detailed information, see [subsection 3.3.2](#) and the simulation folder in the mentioned workspace ('ICC-Lees-Green.py'). Then, we executed the Python script to plot the related data corresponding to each figure. Experimental conditions specific to each variations are given in [Table 3](#).

3 Model Modifications

3.1 Mathematical Equations

The equations are the same as reported in the supplementary material of [Lees-Green et al. \(2014\)](#) with the exception of some of the flux definitions. There are no definitions of the fluxes for some conductances: voltage-gated calcium channel (J_{CaV}), T-type calcium channel (J_{CaT}), Soc channel (J_{Soc}), and voltage-gated nonselective channels (J_{NSCC}). Definitions are defined (see [Equation 11](#)) from the references and compared to the similar descriptions of the other fluxes with similar behaviour in the original work and confirmed through the MATLAB updated model of ICC. The following equation defines the total fluxes for the conductances mentioned above:

$$J = -I/ZFV \quad (11)$$

where I and V indicate the current and the cell volume, respectively.

3.2 Parameter Values

The baseline parameter values are provided in the supplementary material of [Lees-Green et al. \(2014\)](#). In an attempt to reproduce the same results, we applied a few modifications as listed below (also see [Table 1](#)):

1. The rescaling factor of time constant for the Ano1 channel (τ_{Ano1}) was not defined in the primary paper; we highlight that the rescaling factor was needed to produce Figure 3 from [Lees-Green et al. \(2014\)](#).
2. In Figure 3D, Lees-Green et al. defined the step changes in calcium concentrations. Applying the mentioned range in the primary paper could not reproduce the same result; for more information, see [subsection 3.3.1](#).
3. To reproduce Figure 5A-5C, the K parameter for voltage-dependent gating equation (F_{NaV}) is changed (from -4.5 mV to 4.5 mV). The K parameters corresponding to the inactivation of the voltage-gated channels are always positive.
4. To reproduce Figure 5B-5D, the conductance value for the 'CaV' channel is needed to be reduced from the original value $g = 4$ nS to $g = 3.72$ nS.
5. No initial conditions were defined in the primary paper. All initial conditions were extracted from the original article using the Engauge Digitizer software ([Mitchell et al., 2020](#)). These values are embedded in the 'json' files regarding each variation; see [subsection 3.3.2](#).

3.3 Computational Simulation

The differential/algebraic equations system is solved using solver CVODE in CellML. with the relative tolerance and time step of $Relol = e - 7$ and $step = 1e - 3$, respectively. In this work, the model results are categorized into two different parts: The Ano1 model ([Figure 1](#)) and the ICC model ([Figure 2](#) and [Figure 3](#)).

Figure	Parameters	Primary paper	provided value [Matlab Code]	Current CellML
3	τ_{scale} [Ano1]	Not defined	1	10
4 & 5	τ_{scale} [Ano1]	Not defined	1	1
3D	Ca_{Ano1}	0.1: 3.3:20	Not defined	15:2.5:30
3	K_s [Ano1]	0.0156	0.156	0.0156
4 & 5	K_s [Ano1]	0.0156	0.156	0.156
5A-5C	K_f [NaV]	-4.5	Not the same Eqs	4.5
5B-5D	g [CaV]	4	3	3.72

Table 1. Reconciliation of parameter values used in different implementations.

3.3.1 Ano1 model

The Ano1 model is defined to study the kinetics of the Ano1 channels ($g = 0.5$ nS, $E_{Cl} = 0$ mV). In the voltage-clamp simulations, the voltage protocol stepped from a holding potential of 0 mV to a membrane potential (V_m) ranging between -100 and +100 mV in 40 mV increments for 800 ms, followed by a hyperpolarizing step -100 mV. Voltage clamp simulations were carried out at three different values of fixed $[Ca^{2+}]$: 0.1 μM (Figure 3A), 1 μM (Figure 3B), and 10 μM (Figure 3C), see Equation 12.

$$V_m = \begin{cases} 0 \text{ mV} & t \leq 1 \text{ s} \\ V_{test} & 1 \text{ s} < t \leq 9 \text{ s} \\ -100 \text{ mV} & \text{otherwise} \end{cases} \quad (12)$$

Figure 3D was reproduced using the Ano1 model by setting the membrane potential (V_m) at a fixed holding potential (ranging from -100 to +100 mV in 40 mV increments) and applying a step-change in Ca_{Ano1} from 15 μM to 30 μM for 800 ms. For more detailed information see Equation 13 and Table 2.

$$Ca_{Ano1} = \begin{cases} Ca_{test} & 1 \text{ s} < \text{time} \leq 9 \text{ s} \\ 0 \mu M & \text{otherwise} \end{cases} \quad (13)$$

Figure	Holding Voltage [mV]	Baseline Voltage[mV]	Testing voltage [mV]	$Ca_{Ano1} \times 10^{-6}$ [M]
1A	-100	0	-100:40:100	0.1
1B	-100	0	-100:40:100	1
1C	-100	0	-100:40:100	10
	Holding Ca_{Ano1}	Baseline Ca_{Ano1}	Testing voltage	Testing Ca_{Ano1}
1D	0	0	-100:40:100	15:2.5:30

Table 2. Patch clamp protocols

3.3.2 ICC model

The ICC model not only studies the effect of altering E_{Cl} but also investigates and tests whether a variety of different ion currents are plausible candidates for the plateau current that maintains the plateau phase of the slow-wave. Simulations are divided into two distinct categories: high chloride and low chloride environments.

The high chloride environment contains four different variations, as the following:

1. HIGH-CL(NSV): Non-Selective Voltage Activated Channel, to reproduce Figure 4B and F; 'HCl-NSV.json' is loaded to update the model's parameters and initial conditions accordingly. Data is saved in the CVS files: Fig4-2a and Fig4-2b. All the data held in the appropriate string of formats 'ia' and 'ib' where 'i' indicates the figure subplot, 'a' and 'b' indicate the data related to the Wild-type, and Ano1 knockout scenarios, respectively.

2. High-Cl(NaV): Ion current specific to Voltage-gated Na channel, to reproduce Figure 4A and E; the JSON file 'HCl-NaV.json' is loaded.
3. High-Cl(NSCa): Ion currents specific to Ca²⁺-activated non-selective channel, to reproduce Figure 4C and G; 'HCl-NSCC.json' is loaded.
4. High-Cl(CaV): Ion current specific to Voltage-gated Ca channel, to reproduce Figure 4D and H; 'HCl-CaV.json' is loaded.

The low chloride environment consists of two different variations:

1. Low-Cl(NaV): Ion currents specific to Voltage-gated Na channel, to reproduce 5A-5C; 'LCI-NaV.json' is loaded.
2. Low-Cl(NSCa): Ion currents specific to Ca²⁺-activated non-selective channel, to reproduce Figure 5B and 5D; 'LCI-NSCC.json' is loaded.

3.4 Model Results

We employed the Engauge Digitizer software (Mitchell et al., 2020) to extract the data from all figures from Lees-Green et al. (2014) to present alongside the results of the present work. The reproduction of all figures of Lees-Green et al. (2014) is given in Figures 1-3, which show good agreement against the data of the primary paper. Solid lines show the output from this curation effort, and stars show discrete points found by the Engauge Digitizer of the figures in the primary paper.

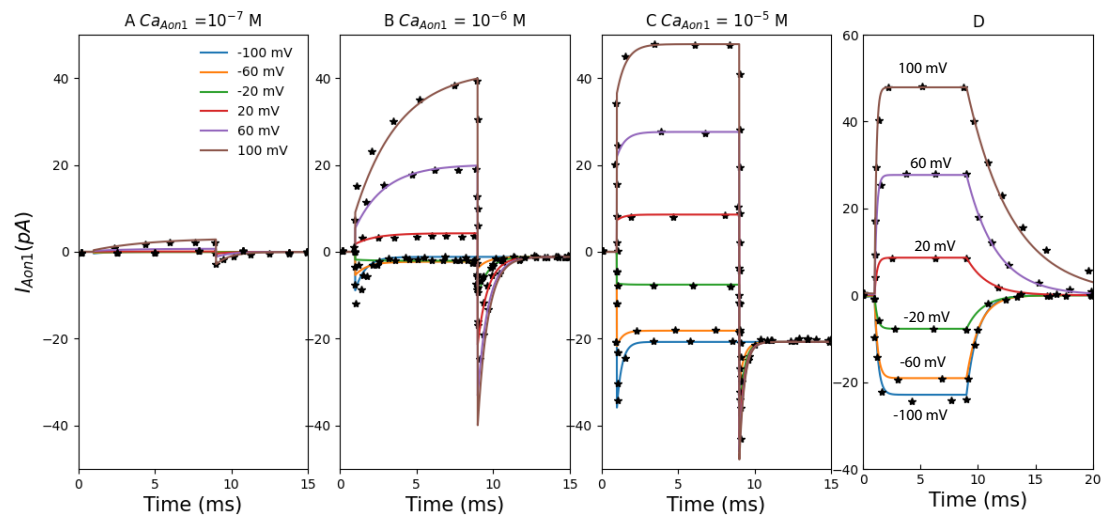


Figure 1. The primary data (*) of (Lees-Green et al., 2014, Figure 3) with our reproduction of all subfigures. To reproduce Lees-Green et al. (2014, Figure 3), use the Python script 'IAno1.py' to produce the simulation results and 'Plot_Fig3_Ano1.py' to plot the results. A–C: simulated voltage-clamp experiments to test the response of the Ano1 model to the voltage protocol.

Table 3. Parameters are categorized according to each different variation. There are six different variations. These parameters are collected from (Lees-Green et al., 2014, Tables 2A and 2B), and some from the appendix of the primary paper.

↓Variation- Current→		I_{Ano1}	I_{CaT}	I_{BK}	I_{BNa}	I_{NaV}	I_{CaV}	I_{NSV}	I_{NCX}	I_{KV}	I_{KERG}	I_{NSCC}	I_{SOC}											
High-Cl (NaV) $V_{PMCA} = 50 \mu M/s$	τ_d	Eqs:24-32 & Eq: 35	0.006	Eq: 37	Eq: 36	Eq:40	0	0	0	0	0	0	Eqs: 33-34											
	V_d^h		-26			-25																		
	K_d		-6			-5.6																		
	a_d		1			1																		
	τ_f		0.04																					
	V_f^h		-66			Eq:39																		
	K_f		6																					
	a_f		1																					
	$g[nS]$		15			3								5	1	6	0	0	0	0	0	0	0.1	
	High-Cl (NSV) $V_{PMCA} = 50 \mu M/s$		τ_d			Eqs:24-32 & Eq: 35								0.003	Eq: 37	Eq: 36	0	0	0.005	0	0	0	0	Eqs: 33-34
V_d^h		-45	-30																					
K_d		-3	-5																					
a_d		1	1																					
τ_f		0.1	0.8																					
V_f^h		-55	-50																					
K_f		5	5																					
a_f		1	1																					
$g[nS]$		20	4	9	2		0	0	30	0	0	0	0	0.1										
High-Cl (NSCa) $V_{PMCA} = 100 \mu M/s$		τ_d	Eqs:24-32 & Eq: 35	0.006	Eq: 37		Eq: 36	0	0	0	0	0	0	Eqs:50-51					Eqs: 33-34					
	V_d^h	-26																						
	K_d	-6																						
	a_d	1																						
	τ_f	0.04																						
	V_f^h	-66																						
	K_f	6																						
	a_f	1																						
	$g[nS]$	20		3.7		9									2	0	0	0		0	0	0	30	0.1
	High-Cl (CaV) $V_{PMCA} = 1500 \mu M/s$	τ_d [s]		Eqs:24-32 & Eq: 35		0.006									Eq: 37	Eq: 36	0	0.006		0	0	0	0	0
V_d^h [mV]		-26	-26																					
K_d [mV]		-6	-6																					
a_d		1	1																					
τ_f [s]		0.04																						
V_f^h [mV]		-66	Eq:44																					
K_f [mV]		6																						
a_f		1																						
g [nS]		15	3		9	2	0	6.2	0	0	0	0	0	0.1										
Low-Cl (NaV) $V_{PMCA} = 50 \mu M$		τ_d	Eqs:24-32 & Eq: 35		0.006	Eq: 37	Eq: 36	Eq: 46	0	0	0	0.003	Eq:49	0				Eqs: 33-34						
	V_d^h	-26		-39	-20			-35																
	K_d	-6		-5.6	-4.6			-7.1																
	a_d	1		1	1			1																
	τ_f	0.04		1	0.045			0.003																
	V_f^h	-66		-55	-45			-42																
	K_f	1		4.5	4.4			4																
	a_f	1		0.53	1			1																
	$g[nS]$	15		3	5			1				10	0		0	0	6		3	0	0.1			
	Low-Cl (NSCa) $V_{PMCA} = 110 \mu M$	τ_d		Eqs:24-32 & Eq: 35	0.003			Eq: 37				Eq: 36	0		0	0	Eq:52		0	0	Eqs:50-51	Eqs: 33-34		
V_d^h		-45																						
K_d		-3																						
a_d		1																						
τ_f		0.1																						
V_f^h		-55																						
K_f		5																						
a_f		1																						
$g[nS]$		10	4		5	1	0		0	0	N/A			0				0					30	0.1

4 Discussion

We demonstrated that most results of Lees-Green et al. (2014) can be reproduced given some added equations (subsection 3.1) and adjustments (Table 1) made to model parameter values. We implemented the model (Lees-Green et al., 2014) using CellML, replicated all the model behaviors reported in the primary paper, and summarized detailed experiment settings for simulation in subsection 3.3. While we are generally able to reproduce the simulation results there are some cases where the results presented here are not identical. The reason for this is the lack of information about the initial conditions from the primary paper leading to slight phase shifts in the oscillatory behaviour as the simulations are not starting from the same initial state. These differences do not impact their interpretation and the conclusions drawn in the primary

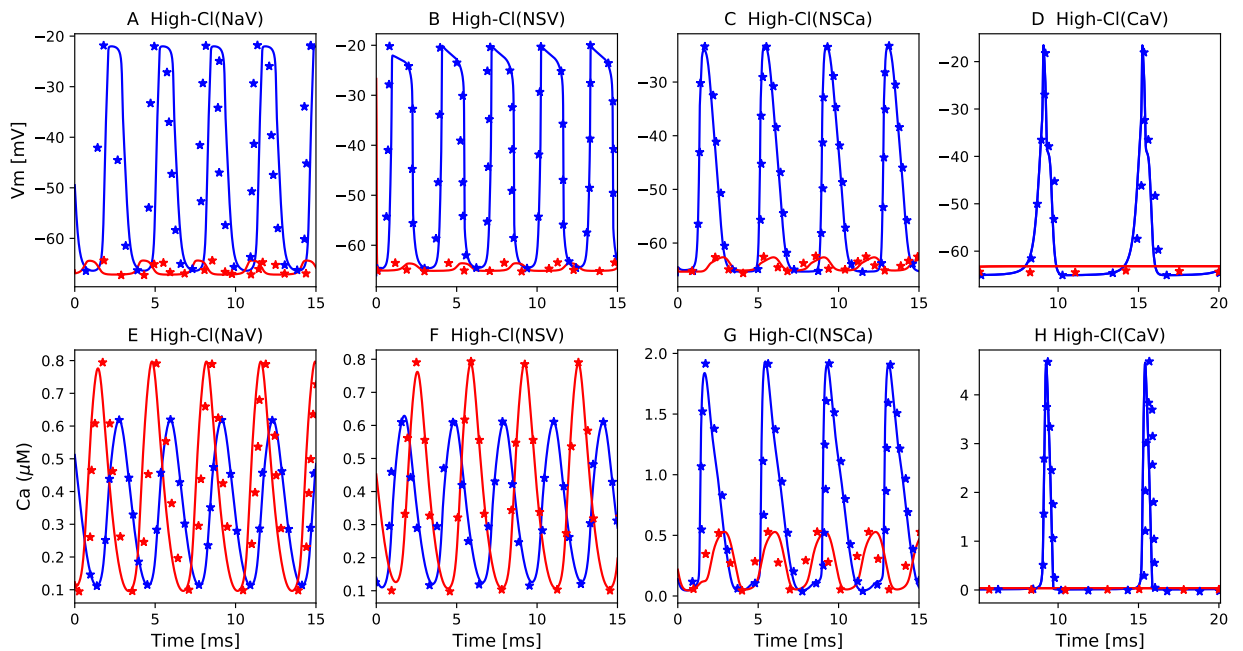


Figure 2. The primary data (*) of (Lees-Green et al., 2014, Figure 4) with our reproduction of all subfigures. To reproduce (Lees-Green et al., 2014, Figure 4), use the Python script 'ICC_Lees_Green.py' to produce the simulation results and 'Plot_Fig4_ICC.py' to plot the results. A-H: membrane potentials were simulated using the wild-type (WT) and Ano1 knockout (KO) scenarios (blue and red lines, respectively).

paper of Lees-Green et al. (2014). Python code was needed in some cases and can be found at <https://models.physiomeproject.org/workspace/762>. As this paper follows the tenets of FAIR (Wilkinson et al., 2016), further credibility is lent to the primary models.

5 Acknowledgements

The authors gratefully acknowledge the support of the Ministry of Business, Innovation and Employment's Catalyst Strategic Fund (12 Labours).

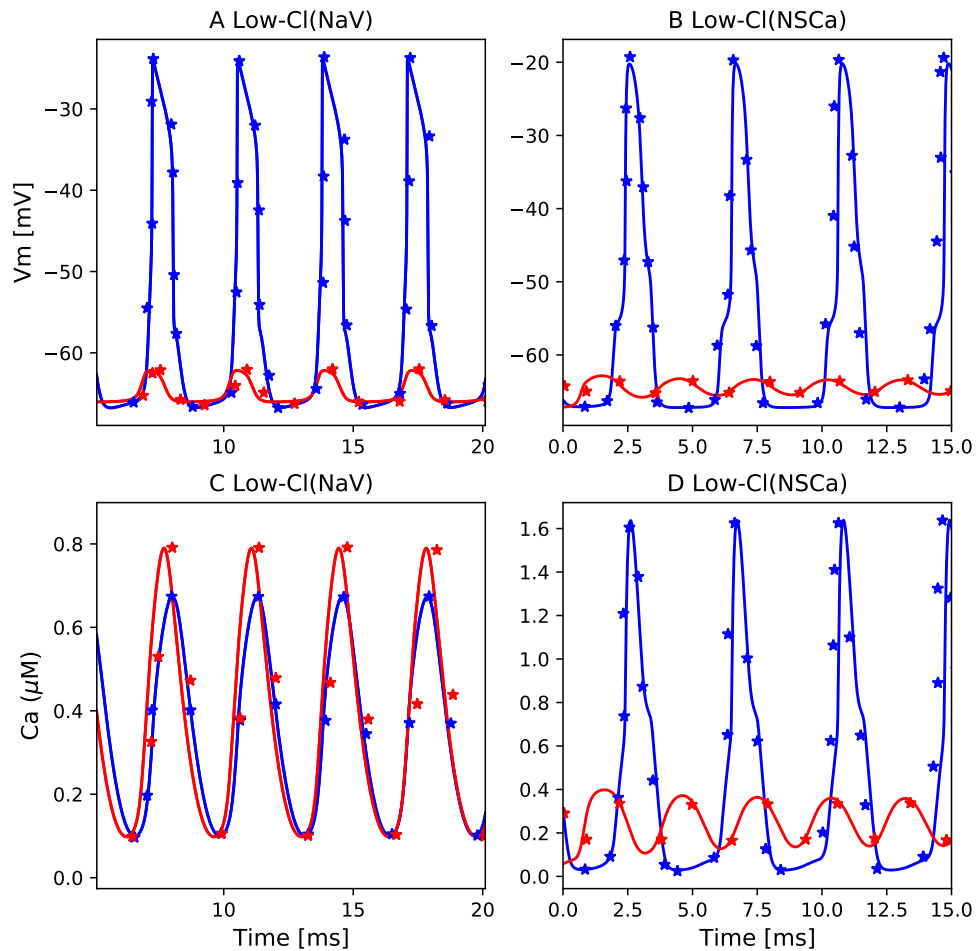


Figure 3. The primary data (*) of (Lees-Green et al., 2014, Figure 5) with our reproduction of all subfigures. To reproduce (Lees-Green et al., 2014, Figure 5), use the Python script 'ICC_Lees_Green.py' to produce the simulation results and 'Plot Fig5_ICC.py' to plot the results. A-D: membrane potentials simulated using the wild-type (WT) and Ano1 knockout (KO) scenarios (blue and red lines, respectively).

References

- F. T. Bergmann, R. Adams, S. Moodie, J. Cooper, M. Glont, M. Golebiewski, M. Hucka, C. Laibe, A. K. Miller, D. P. Nickerson, B. G. Olivier, N. Rodriguez, H. M. Sauro, M. Scharm, S. Soiland-Reyes, D. Waltemath, F. Yvon, and N. L. Novère. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics*, 15(1):369, Dec. 2014. ISSN 1471-2105. doi: 10.1186/s12859-014-0369-z. URL <http://www.biomedcentral.com/1471-2105/15/369/abstract>.
- A. A. Cuellar, C. M. Lloyd, P. F. Nielsen, D. P. Bullivant, D. P. Nickerson, and P. J. Hunter. An overview of cellml 1.1, a biological model description language. *SIMULATION*, 79(12):740–747, 2003. doi: 10.1177/0037549703040939. URL <https://doi.org/10.1177/0037549703040939>.
- A. Garny and P. J. Hunter. OpenCOR: a modular and interoperable approach to computational biology. *Computational Physiology and Medicine*, 6:26, 2015. doi: 10.3389/fphys.2015.00026.

URL <http://journal.frontiersin.org/article/10.3389/fphys.2015.00026/abstract>.

- R. Lees-Green, S. J. Gibbons, G. Farrugia, J. Sneyd, and L. K. Cheng. Computational modeling of anoctamin 1 calcium-activated chloride channels as pacemaker channels in interstitial cells of cajal. *American Journal of Physiology-Gastrointestinal and Liver Physiology*, 306(8):G711–G727, 2014. doi: 10.1152/ajpgi.00449.2013.
- M. Mitchell, B. Muftakhidinov, T. Winchen, A. Wilms, B. van Schaik, badshah400, Mo-Gul, T. G. Badger, Z. Jędrzejewski-Szmek, kensington, and kylesower. markumitchell/engauge-digitizer: Nonrelease. July 2020. doi: 10.5281/zenodo.3941227. URL <https://doi.org/10.5281/zenodo.3941227>.
- L. F. Shampine and M. W. Reichelt. The matlab ode suite. *SIAM journal on scientific computing*, 18(1):1–22, 1997.
- M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.

Reproducibility report for: Computational modeling of anoctamin 1 calcium-activated chloride channels as pacemaker channels in interstitial cells of Cajal.

Submitted to: *Physiome*

Manuscript identifier: S000015

Curation outcome summary: Successfully reproduced all the figures presented in this manuscript.

Box 1: Criteria for repeatability and reproducibility

■ Model source code provided:

■ Source code: a standard procedural language is used (e.g. MATLAB, Python, C)

- There are details/documentation on how the source code was compiled
- There are details on how to run the code in the provided documentation
- The initial conditions are provided for each of the simulations
- Details for creating reported graphical results from the simulation results

■ Source code: a declarative language is used (e.g. SBML, CellML, NeuroML)

- The algorithms used are defined or cited in previous articles
- The algorithm parameters are defined
- Post-processing of the results are described in sufficient detail

Executable model provided:

- The model is executable without source (e.g. desktop application, compiled code, online service)
- There are sufficient details to repeat the required simulation experiments

■ The model is described mathematically in the article(s):

- Equations representing the biological system
- There are tables or lists of parameter values
- There are tables or lists of initial conditions
- Machine-readable tables of parameter values
- Machine-readable tables of initial conditions

■ The simulation experiments using the model are described mathematically in the article:

- Integration algorithms used are defined
- Stochastic algorithms used are defined
- Random number generator algorithms used are defined
- Parameter fitting algorithms are defined
- The paper indicates how the algorithms yield the desired output



Box 2: Criteria for accessibility

- Model/source code is available at a public repository or researcher's web site
 - Prohibitive license provided
 - Permissive license provided
 - Open-source license provided
- All initial conditions and parameters are provided
- All simulation experiments are fully defined (events listed, collection times and measurements specified, algorithms provided, simulator specified, etc.)

Box 3: Rules for Credible practice of Modeling and Simulation^a

^aModel credibility is assessed using the Interagency Modeling and Analysis Group conformance rubric:
<https://www.imagwiki.nibib.nih.gov/content/10-simple-rules-conformance-rubric>

- Define context clearly: Extensive
- Use appropriate data: Extensive
- Evaluate within context: Extensive
- List limitations explicitly: Insufficient
- Use version control: Adequate
- Document adequately: Extensive
- Conform to standards: Extensive

Box 4: Evaluation

- Model and its simulations could be repeated using provided declarative or procedural code
- Model and its simulations could be reproduced

Summary comments: Model and source code are available in the associated OMEX archive. This was used in our attempt to reproduce the results presented in the paper. We successfully ran the Python, CellML and SED-ML files provided to reproduce Figure 1 - Figure 3 as presented in this manuscript.



Anand K. Rampadarath¹, PhD
Curator at Center for Reproducible Biomedical Modeling

¹Contact: info@reproduciblebiomodels.org