Panama: A tool for ion channel biophysics simulation

Anuj Guruacharya
Department of Biology, University of Oklahoma, Norman,OK
anuj2054@ou.edu
4058898692

I have created an online tool and an R library that simulates biophysics of voltage-gated ion channels. It is made publicly available as an R library called Panama at github.com/anuj2054/panama and as a web app at neuronsimulator.com. A need for such a tool was observed after surveying available software packages. I found that the available packages are either not robust enough to simulate multiple ion channels, too complicated, usable only as desktop software, not optimized for mobile devices, not interactive, lacking intuitive graphical controls, or not appropriate for undergraduate education. My app simulates the physiology of 11 different channels - voltage-gated sodium, potassium, and chloride channels; channels causing A-current, M-current, and After-HyperPolarization (AHP) current; calcium-activated potassium channels; low threshold T type calcium channels and high threshold L type calcium channels; leak sodium and leak potassium channels. It can simulate these channels under both current clamp and voltage clamp conditions. As we change the input values on the app, the output can be instantaneously visualized on the web browser and downloaded as a data table to be further analyzed in a spreadsheet program. The app is a first of its kind, mobile-friendly and touch-screen-friendly online tool that can be used to teach undergraduate neuroscience classes. It can also be used by researchers on their local computers as part of an R library. It has intuitive touch-optimized controls, instantaneous graphical output, and yet is pedagogically robust for education and casual research purposes.

Neuroscience education, ion channel biophysics, Hodgkin-Huxley simulation, web app for neuroscience

1 Introduction

The Hodgkin-Huxley (Hodgkin et.al., 1952) model is one of the fundamental neuronal models. Its mathematical form is a set of differential equations that students in a neuroscience class are taught before moving on to more complex models. Computational simulations using this model strengthen students' concepts of action potentials and ion channels.

Existing simulation programs, such as NEURON (Hines and Carnevale, 1997) and GENESIS (Bower et al., 2007), serve as powerful tools for simulating the response of whole-cell or single-channel parameters to electrical or pharmacological stimuli. Although such software tools are free and could be used for educational or research purposes, they require substantial training and may not be suitable for casual use by researchers. Some effort has been directed towards making educational packages that demonstrate ion channel biophysics that are freely available. These are good tools to teach students about action potentials, ion channel currents and voltages. However, each of these tools has its own disadvantages. I highlight some of them in particular. HHsim (http://www.cs.cmu.edu/ dst/HHsim/) (Touretzky et al., 2003) requires the software to be downloaded and a matching version of Matlab installed on the desktop computer. Neurophysiology Virtual Lab (http://vlab.amrita.edu/?sub=3&brch=43) (Sridharan et al., 2016) requires a signup procedure and is not mobile-friendly. NeuroLab (http://sites.lafayette.edu/schettil/neurolab/) (Schettino, 2014) requires a special software environment called Netlogo. Others such as Phet (https://phet.colorado.edu/en/simulation/neuron) are cartoon reconstructions of ion channel physiology with restricted features. A webapp located at http://myselph.de/hodgkinHuxley.html serves a sufficient purpose in it being an online simulator, but is not touch-optimized, and does not have enough adjustable input controls to be adequate for an undergraduate neurophysiology class. Nerve (http://nerve.bsd.uchicago.edu/) is not touch- or mobile- optimized while others, such as the programs available at http://www.eng.utoledo.edu/ smolitor/download.htm (Molitar et al., 2006), are Matlab packages. Any software package that is dependent on Matlab is not ideal for wide distribution because of the overwhelming cost of Matlab and the requirement to preinstall Matlab. Any software package dependent on Java in the browser is not ideal because of the unavailability of built-in Java support in some modern web browsers.

Thus, a need was felt to make a tool that had the following characteristics: 1) mobile-friendly, 2) touch-screen-friendly, 3) pedagogically adequate for an undergraduate neurophysiology class, 4) completely online, 5) not reliant on Matlab or Java software, 6) built with open sourced code, and 7) usable by researchers that want an intuitive way to change ion channel parameters and download the data. To our knowledge, no electrophysiology simulation tool exists that satisfied all these criteria. In this paper, a new webapp for simulating the biophysics of voltage-gated ion channels is described. It has been made publicly available at http://www.neuronsimulator.com and as a downloadable R package called Panama through Github. The design of my simulator overcomes the limitations of previous simulators and satisfies all the criteria listed previously. R (R Core Team, 2014), Shiny package (RStudio Inc., 2014), and Lattice package (Sarkar, 2008) were used to code the software. It has multiple input controls for both voltage clamp and current clamp conditions. It outputs the voltage, current, and conductance values as graphs for each ion channel.

2 Methods

2.1 Numerical design of the simulator

The 11 channels simulated in this app were voltage-gated sodium, potassium, and chloride channels; calcium-activated potassium channels (*KCa*); T-type calcium channels (*CaT*); L-type calcium channels (*CaL*), leak sodium (*NaLeak*), and leak potassium (*KLeak*) channels; A current channels; M current channels; and AHP current channels. Each channel was represented by its maximal conductance or permeability (g_n or p_n where n is the specific ion channel), its ionic current (I_n), its reversal potential (E_n) and its associated gating parameters. Total ionic current (I_{net}) was modeled as the sum of all those individual Hodgkin-Huxley style ionic currents: i_{Na}, i_K, i_{Cl}, i_{NaLeak}, i_{KLeak} ,i_A, i_M, i_{KCa}, i_{AHP}, i_{KCa}, i_M, i_{CaT}, i_{CaL}. The models for these channels were modified from those used in the EOTN software (Huguenard et al.,1996).

Voltage or current across the membrane was held constant depending on the clamping conditions. For the current clamp case, I_{net} was held for the clamp duration at the applied current provided by the user; V_{net} was determined from Kirchoff's current law by solving a differential equation given in Equation 1. The membrane capacitance per area represented by C in Equation 1 is input by the user and set to a default value of 0.01 nFarads.

$$\frac{dV_{net}}{dt} = \frac{1}{C}(I_{net} - i_{Na_{leak}} - i_{K_{leak}} - i_{Na} - i_K - i_{Cl} - i_{CaL} - i_{KCa} - i_{CaT} - i_A - i_{AHP} - i_M) \tag{1}$$

For the voltage clamp case, V_{net} was held for the clamp duration (set to a default of 50 ms) at the applied voltage provided by the user; I_{net} was determined from Kirchhoff's current law as shown in Equation 2.

$$I_{net} = (-i_{K_{leak}} - i_{K_{leak}} - i_{Na} - i_K - i_{Cl} - i_{CaL} - i_{KCa} - i_{CaT} - i_A - i_{AHP} - i_M)$$

The default simulation time was set to 70 ms, with 10 ms being the pre-clamp and 10 ms being the post-clamp duration. This helped avoid data overflow issues. However the user can change the clamp duration, pre-clamp duration and post-clamp duration to increase the total simulation time to 1000 ms.

Equations 3 to 13 described the individual ionic currents for each channel used in the simulator. The electrochemical gradients driving the flow of ions were represented as voltage sources (E_n) whose voltages were determined by Nernst equations that are dependent on the ratio of extra- to intra-cellular concentrations of the ionic species of interest. All gating variables, *m*, *n*, *h*, m_{KCa}, $m1_A$, $m2_A$, $h1_A$, $h2_A$, m_{AHP}, m_M, m_T, m_L were dimensionless numbers, ranging from 0 to 1. *m* denotes the probability of finding a channel in its open/permissive state and 1-*m* denotes the probability of it being closed or in a non-permissive state (current flow will be zero in this state). Other gating variables followed a similar representation. Ca_{in} and Ca_{out} represented the concentration of calcium ions inside the cell and outside the cell in mM. The constant values used in Equations 5 to 16 which cannot be changed by the user are the temperature (*T*) set to be a room temperature of 293.15 Kelvins, gas constant (R) set to 8314 Joules/mole/kelvin, and the Faraday constant (*F*) set to 96485 Ampere-seconds/mole.

$$i_{Na_{leak}} = g_{Na_{leak}}(V_{net} - E_{Na_{leak}})$$

$$i_{K_{leak}} = g_{K_{leak}}(V_{net} - E_{K_{leak}})$$

$$i_{Na} = g_{Na}(V_{net} - E_{Na})hm^3$$

$$i_K = g_K(V_{net} - E_K)n^4$$

$$i_{Cl} = g_{Cl}(V_{net} - E_{Cl})m$$

$$i_{Kca} = g_{KCa}(V_{net} - E_k) * m_{Kca}$$

$$i_A = g_A(V_{net} - E_K)(0.6m1_A^4 h1_A + 0.4m2_A^4 h2_A)$$

$$i_{AHP} = g_{AHP}(V_{net} - E_K)m_{AHP}^2$$

$$i_M = g_M(V_{net} - E_K)m_M$$

$$i_{CaT} = \frac{4p_{CaT}m_{CaT}^2 h_{CaT}V_{net}F^2}{RT}(CaIn - CaOut)\frac{e^{-2FV_{net}/RT}}{1 - e^{-2FV_{net}/RT}}$$

$$i_{CaL} = \frac{4 p_{CaL} m_{CaL}^2 V_{net} F^2}{RT} (CaIn - CaOut) \frac{e^{-2FV_{net}/RT}}{1 - e^{-2FV_{net}/RT}}$$

Differential equations in Equations 14 to 27 were used to model the activation and deactivation of the gating variables. The differential equations were discretized using an explicit forward Euler algorithm with a default time step of 0.025 ms. The time step can be changed by the user, however it is not recommended to turn it lower than the default value since it can slow down the simulation. The equations were assumed to be initial value problems and solved for each value of voltage across the membrane. So these differential equations were solved with different values of V_{net} across the membrane.

$$\frac{dm}{dt} = \alpha_m V_{net} (1 - m) - \beta_m V_{net} m \tag{14}$$

$$\frac{dh}{dt} = \alpha_h V_{net} (1 - h) - \beta_h V_{net} h \tag{15}$$

$$\frac{dn}{dt} = \alpha_n V_{net} (1 - n) - \beta_n V_{net} n \tag{16}$$

$$\frac{dm_{AHP}}{dt} = \alpha_{m_{AHP}} (1 - m_{AHP}) - \beta_{m_{AHP}} m_{AHP}$$

$$\frac{dm_{KCa}}{dt} = \alpha_{m_{Kca}} V_{net} (1 - m_{Kca}) - \beta_{m_{Kca}} V_{net} m_{Kca}$$

$$\frac{dm_M}{dt} = \frac{m_{\inf\ M} V_{net} - m_M}{m_{tauM} V_{net}} \tag{19}$$

$$\frac{dm1_A}{dt} = \frac{m1_{infA} V_{net} - m1_A}{m1_{tauA} V_{net}} \tag{20}$$

$$\frac{dh1_A}{dt} = \frac{h1_{infA} V_{net} - h1_A}{h1_{tauA} V_{net}} \tag{21}$$

$$\frac{dm2_A}{dt} = \frac{m2_{infA} V_{net} - m2_A}{m2_{tauA} V_{net}} \tag{22}$$

$$\frac{dh2_A}{dt} = \frac{h2_{infA} V_{net} - h2_A}{h2_{tauA} V_{net}} \tag{23}$$

$$\frac{dm_{CaT}}{dt} = \frac{m_{infCaT} V_{net} - m_{CaT}}{m_{tauCaT} V_{net}} \tag{24}$$

$$\frac{dh_{CaT}}{dt} = \frac{h_{infCaT} V_{net} - h_{CaT}}{h_{tauCaT} V_{net}} \tag{25}$$

$$\frac{dm_{CaL}}{dt} = \alpha_{m_{CaL}} V_{net} (1 - m_{CaL}) - \beta_{mCaL} V_{net} m_{CaL}$$

In all models, voltage was measured in *mVolt*, current in *nAmp*, time in *msec*, conductance in *µSiemens*, and capacitance in *nFarads*. The model parameters for sodium channels were derived from data of Huguenard et al. (1988). The potassium channel model used was of a general delayed rectifier. T-type calcium channel (i_{CaT}) was modelled using the constant field equation. L-type calcium channel (i_{CaL}) was also modelled using constant field equation as in i_{CaT}, except that it was considered to not inactivate. i_{CaL} was based upon the data of Kay and Wong (1987) from isolated hippocampal pyramidal cells. Calcium activated potassim channel (i_{KCa}) was modelled according to the procedure used by Yamada et al. (1989). The A current was modelled to inactivate with two time constants. The first component comprising of *m1_A*

$m1_A$ and $h1_A$ contributes 60% to the total value of the gating variables. The second component comprising of $m2_A$ and $h2_A$ contributes 40% to the total value of the gating variables. The M current model was adapted from Adams et al. (1982). The AHP current was modelled according to the model in bullfrog sympathetic cells by Yamada et al. (1989).

2.2 Software design of the simulator

The webapp was created using the R programming language. After an initial survey of different languages and packages available in each language, the R language was chosen for its availability of Shiny and Lattice packages which are both excellent packages for web development and graphics development. R was also chosen because of its widespread use by biologists and undergraduates. The differential equations were coded into the script using only R, without using any external differential equation solver packages such as deSolve.

The Shiny package was used to serve the webpages. The Twitter bootstrap toolkit was used as the theme for user interface controls. Sliders from the bootstrap UI toolkit were used to make the input controls touch-friendly so that users do not have to type the values in a textbox.

The Lattice package was used to create the graphs that were embedded into the webpage. The output of the webapp is a set of voltage, current and conductance graphs for the channels. These can be visualized instantaneously while changing the input values on the app after pressing the update button, or they can be downloaded as CSV tables and analyzed in a spreadsheet software.

The code is open-sourced and deposited at http://www.github.com/anuj2054/panama. The front end of the software is coded in a file *ui.R* and the backend is coded in a file *server.R* as shown in Figure I. The app is hosted on a Shiny server located the the High Performance Computing Center facilities at Oklahoma State University. The computations for the equations all occur on the server's side, so that there is no load on the user's computer.

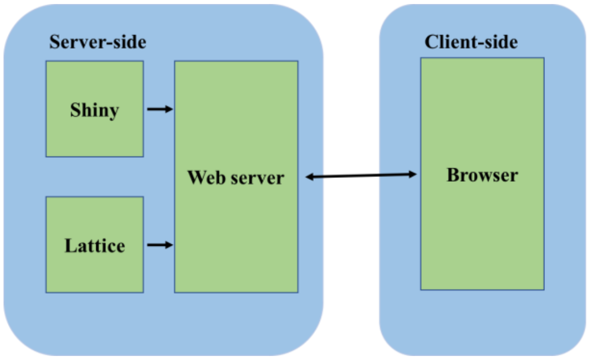
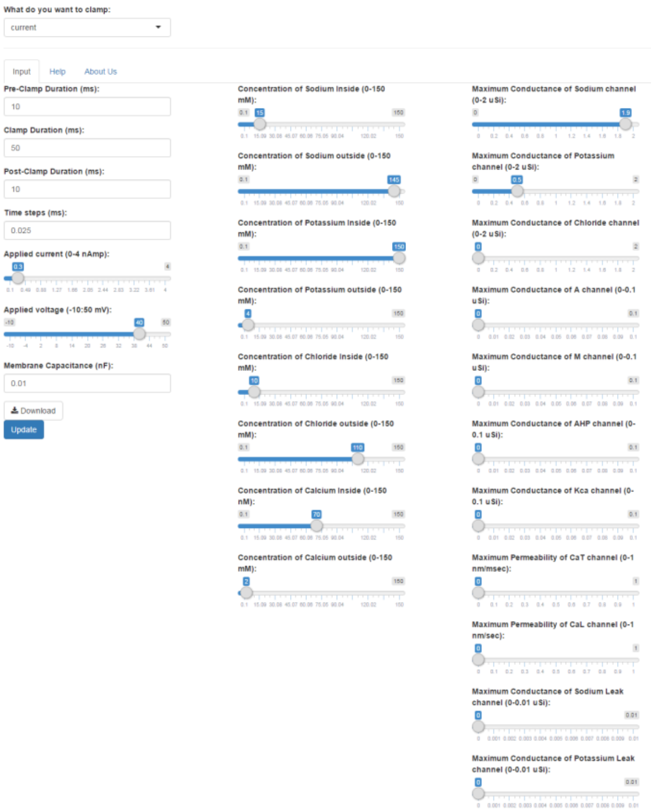


Figure I: Software architecture of the simulator. The server side consists of the Shiny package and the Lattice package. The Shiny package is used for server side computations. The Lattice package is used for graphical display. The computations and the graphs are served to the client browser through the webserver built into the Shiny package.

3 Results and Discussion

There are three ways to access Panama. The first and the easiest way to access it is at http://www.neuronsimulator.com/. A second way that does not require a constant internet connection to work with the software is by using the command runUrl('https://github.com/anuj2054/panama/archive/master.zip') on the R terminal. The command downloads the required files into an R folder and executes the software from the user's local computer. A third way to access the app is by using the command shiny::runGitHub('panama', 'anuj2054') on the R terminal given that the user has preinstalled Shiny. In the second and third methods, once the required command is run and the code is automatically downloaded to the local computer, access to internet is not required anymore.

On a desktop web browser, the input controls appear as in Figure II. However, on a mobile device, the three columns of the input controls are merged into one column for easy scrolling. On a mobile device, the use of sliders eases the process of entering values for individual parameters of the model. However precision is not sacrificed. The user can change the input parameter values by hovering above the slider and using the keyboard to fine tune the exact number they want to three significant decimal places. The default values on the app can be reloaded by refreshing the web browser.



What do you want to clamp: current

Pre-Clamp Duration (ms): 10

Clamp Duration (ms): 50

Post-Clamp Duration (ms): 10

Time steps (ms): 0.025

Applied current (0-4 nAmp): 0.3

Applied voltage (-10:50 mV): 40

Membrane Capacitance (nF): 0.01

Concentration of Sodium inside (0-150 mM): 15

Concentration of Sodium outside (0-150 mM): 145

Concentration of Potassium inside (0-150 mM): 150

Concentration of Potassium outside (0-150 mM): 4

Concentration of Chloride inside (0-150 mM): 10

Concentration of Chloride outside (0-150 mM): 110

Concentration of Calcium inside (0-150 nM): 70

Concentration of Calcium outside (0-150 mM): 2

Maximum Conductance of Sodium channel (0-2 uSi): 1.9

Maximum Conductance of Potassium channel (0-2 uSi): 0.5

Maximum Conductance of Chloride channel (0-2 uSi): 0

Maximum Conductance of A channel (0-0.1 uSi): 0

Maximum Conductance of M channel (0-0.1 uSi): 0

Maximum Conductance of AHP channel (0-0.1 uSi): 0

Maximum Conductance of Kca channel (0-0.1 uSi): 0

Maximum Permeability of CaT channel (0-1 nm/msec): 0

Maximum Permeability of CaL channel (0-1 nm/sec): 0

Maximum Conductance of Sodium Leak channel (0-0.01 uSi): 0

Maximum Conductance of Potassium Leak channel (0-0.01 uSi): 0

Figure II: The input controls of the app on a desktop. The user first has to select the current clamp or voltage clamp conditions from the drop down menu. After selecting the clamping conditions, various parameters can be changed using the scrollbar. Pressing the *Update* button will update the output graphs depending on the values input. The user has to click on *Update* each time they change the parameters. Clicking on the *Download* button will download the time series voltage, current and conductances of the ion channels in a CSV format.

The app outputs conductance, voltage, and current data as both a graphical display and as downloadable CSV tables. The ability to download the output data as CSV tables enables the user to use their own spreadsheet software such as Excel to further analyze the data or embedd the graphs in their own documents. On a desktop web browser, the output graphs appear as shown in Figure III. Each of the lines inside the graphs are color coded and described with the name of the channel inside its respective colored rectangle. During the current clamp step, the current injection can be made more noticeable in the current graph by increasing the applied current and observing the steep red line that appears after the pre-clamp duration. The user can increase the clamp duration to see numerous action potentials that resemble neural spikes.

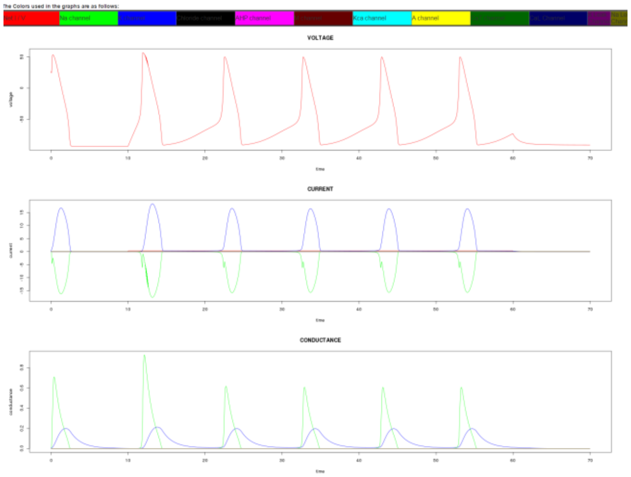


Figure III: Output graphs of the app on a desktop. The simulator gives as output the time series representation of voltage, current, and conductance. The lines in the graph are color coded to represent different ion channels. In this particular graph, only sodium and potassium channels have been activated. The output graphs can also be replicated by downloading the CSV file and using a spreadsheet program to draw the graphs.

Users and students are encouraged to use the app in conjunction with a textbook about ion channels and their electrical properties. They are encouraged to change the default ion concentrations and channel conductances to those for different types of cells such as squid axon cells and observe its effect on the current and voltage of the cell. The default values used in the app are for a mammalian cell at room temperature (Lodish, 2008). During student interaction, it has been particularly helpful in pointing out to them the reasons for the particular changes they observe in the action potentials when different types of channels are activated. Changing the default capacitance and conductance values demonstrates to students how different conditions of a cell membrane affect the electrical properties of the cell. This practise gives students a hands on approach towards learning neurophysiology that they would otherwise only get from a textbook or from an expensive electrophysiology rig.

The numerical output of the simulator was tested against NEURON with similar parameters. Both programs returned equivalent results. The app was also tested under different operating systems (Windows, Android, iOS, Mac, and Linux) and under different browsers (Chrome, Firefox, and Internet Explorer). It was found to operate consistently across all platforms. The app has been well received by practicing and learning neurobiologists who have been exposed to its early versions. User behavior was tested on 3 volunteers. Based on their suggestions, the output graphs were color-coded and a simpler web address, neuronsimulator.com, was bought for the app. Future versions of the app will have phase space graphs to help users better understand membrane dynamics. It will also model synaptic currents where the chloride channels would play an important role. A better help section, tutorials for undergraduate students, and an even cleaner user interface is also being planned.

Panama is a first of its kind, a touch-friendly, mobile-friendly online tool that models electrophysiology of 11 different types of ion channels using Hodgkin-Huxley style differential equations. It requires no user training for installation and no infrastructure for download, making it suitable for casual research use or classroom environments. The webapp or the R package can be used by teachers to teach undergraduate neuroscience classes and also by researchers for casual use.

Conflict of Interest Statement

This project, including the annual cost of the internet domain name, is not directly funded by any organization or grant. However, my salary during the initiation of this project was paid for by National Science Foundation grants IOS1257580 and IOS1350753 to Michael R. Markham.

Information Sharing Statement

The Panama software is available at www.neuronsimulator.com. Its associated scripts are available at https://github.com/anuj2054/panama. R software is available at https://www.r-project.org/. Shiny Server is available at https://www.rstudio.com/products/shiny/. Lattice software is available at https://cran.r-project.org/web/packages/lattice/index.html.

Acknowledgments

This work was also facilitated utilizing the High Performance Computing Center facilities of Oklahoma State University (OSU) in Stillwater, Oklahoma. The author would like to thank Ari Berkowitz, Alex Bernard, Binita Rajbanshi, and

Sharmistha Shayamal for their helpful comments on the manuscript. The author would also like to thank Jesse Schafer and Evan Linde for their support of computing facilities at OSU.

References

1. Adams, P. R., Brown, D. A., & Constanti, A. (1982). M-currents and other potassium currents in bullfrog sympathetic neurones. *The Journal of Physiology*, *330*(1), 537-572.
2. Bower, J. M., Beeman, D., & Hucka, M. (2003). The genesis simulation system.
3. Diwakar, S., Parasuram, H., Medini, C., Raman, R., Nedungadi, P., Wiertelak, E., & Nair, B. (2014). Complementing neurophysiology education for developing countries via cost-effective virtual labs: case studies and classroom scenarios. *Journal of undergraduate neuroscience education*, *12*(2), A130.
4. Bezanilla F. (2017). Nerve. Retrieved 6-16-2017, from http://nerve.bsd.uchicago.edu/nerve1.html.
5. Hines, M. L., & Nicholas T. C. (1997). The NEURON simulation environment. *Neural computation* 9.6, 1179-1209.
6. Hodgkin, A. L., Huxley, A. F., & Katz, B. (1952). Measurement of current-voltage relations in the membrane of the giant axon of Loligo. *The Journal of physiology*, *116*(4), 424-448.
7. Huguenard, J. R., Hamill, O. P., & Prince, D. A. (1988). Developmental changes in Na+ conductances in rat neocortical neurons: appearance of a slowly inactivating component. *Journal of Neurophysiology*, *59*(3), 778-795.
8. Huguenard, J., McCormick, D. A., & Campbell G. (1996). Electrophysiology of the Neuron, A Companion to GM Shepherd's Neurobiology, An Interactive Tutorial. *Trends in Neurosciences*, 19.4, 155.
9. Kay, A. R., & Wong, R. K. (1987). Calcium current activation kinetics in isolated pyramidal neurones of the Ca1 region of the mature guinea-pig hippocampus. *The Journal of Physiology*, *392*, 603.
10. Lodish, H. (2008). *Molecular cell biology*. Macmillan.
11. McCormick, D. A., & Pape, H. C. (1990). Properties of a hyperpolarization-activated cation current and its role in rhythmic oscillation in thalamic relay neurones. *The Journal of physiology*, *431*, 291.
12. Mohagheghi-Nejad, M. R., & Mahnam, A. (2012, August). "Stim-eLab": A simulation tool to enhance education of bioelectrical mechanisms of electrical stimulation. In *Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference* (Vol. 2012, pp. 4591-4594).
13. Molitor, S. C., Tong, M., & Vora, D. (2006). MATLAB-based simulation of whole-cell and single-channel currents. *J Undergrad Neurosci Educ*, *4*, A74-A82.
14. R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria
15. RStudio Inc. (2014). Shiny: Web Application Framework for R. R package version 0.10.1.
16. RStudio Team (2015). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA
17. Sarkar, D. (2008). *Lattice: multivariate data visualization with R*. Springer Science & Business Media.
18. Schettino, L. F. (2014). NeuroLab: a set of graphical computer simulations to support neuroscience instruction at the high school and undergraduate level. *Journal of Undergraduate Neuroscience Education*, *12*(2), A123.
19. Sridharan, A., Sasidharakurup, H., Kumar, D., Nizar, N., Nair, B., Achuthan, K., & Diwakar, S. (2016). Implementing a Web-Based Simulator with Explicit Neuron and Synapse Models to Aid Experimental Neuroscience and Theoretical Biophysics Education. In *Proceedings of the International Conference on Signal, Networks, Computing, and Systems* (pp. 57-66). Springer India.
20. Touretzky, D., Ladsariya, A., Albert, M., Johnson, J., & Daw, N. (2003). HHsim: An open source, real-time, graphical Hodgkin-Huxley simulator. In *Society for Neuroscience Abstracts* (Vol. 29, No. 24.13).
21. Yamada, W. M., Koch, C., & Adams, P. R. (1989, August). Multiple channels and calcium dynamics. In *Methods in neuronal modeling* (pp. 97-133). MIT press.

Anuj Guruacharya He did his bachelors in electronics and communication engineering from Nepal and went on to do his masters in biophysics at Louisiana Tech University, USA. He is graduating with a Ph.D. in computational biology in November,2017. Anuj works on integrating modelling biological phenomenon using non-linear dynamics. He is currently employed by the University of Oklahoma and is a student member of Society of Molecular Biology and Evolution.