ModFossa: A Python Library for Ion Channel Modeling

Gareth Ferneyhough
Sergiu M. Dascalu
Corey Thibeault
Frederick C. Harris, Jr.

Computer Science and Engineering
University of Nevada, Reno
Reno, NV, USA

Abstract

The creation and simulation of ion channel models using continuous-time Markov processes is a powerful and well-used tool in the field of electrophysiology and ion channel research. While several software packages exist for the purpose of ion channel modeling, none are available as a Python library. In an attempt to provide an easy-to-use, yet powerful Markov model-based ion channel simulator, we have developed ModFossa, a Python library supporting easy model creation and stimulus definition, complete with a fast numerical solver, and attractive vector graphics plotting.
keywords: Neuroscience, Simulation, ModFossa

1 Introduction

Ion channels are trans-membrane proteins that control the passive flow of ions into, and out of a biological cell. They exist in all cell types and play crucial roles in cellular sensing and communication, maintenance of the membrane potential, and regulation of cell volume.

For example, voltage-gated sodium, potassium, and calcium channels (among others) work together to shape the automatically recurring cardiac action potential that give our hearts their sure (and usually steady) beat. Philosophy aside, the very thoughts passing through the consciousness of a student as he reads this page exist as nothing more than electro-chemical signals in the brain. Specifically, voltage-gated sodium and potassium channels integrated in the cell membrane of all neurons allow electrical signals to propagate rapidly down the axon in the form of an action potential. When the action potential reaches the end of the axon, voltage-gated calcium channels regulate the release of special chemical messengers called neurotransmitters into the synapse (the structure that connects neurons to one another). As the neurotransmitters diffuse across the synapse they bind to ligand-gated ion channels on the post-synaptic neuron. Depending on the type of ion channel and the type of neurotransmitter, the channels may open or close, thereby altering the membrane voltage. If the post-synaptic neuron receives enough excitatory stimulus from the many presynaptic neurons in which it is connected, its membrane potential will depolarize to a point and it too will experience an action potential, thus passing the signal along to other neurons.

As further evidence of their vital importance, ion channels are specifically targeted by several naturally occurring toxins and poisons, including venom from spiders, snakes, scorpions, and fish. It comes as no surprise that the development of new types of drugs which alter the behavior of certain channels is a major motivator behind ion channel research. As in most scientific fields, researchers often develop mathematical models to assist in their understanding of the results. Ion channels are no exception; several classes of ion channel models exist that allow for the study of ion channel behavior under varying conditions. One such generalization involves modeling channel gating using continuous time Markov processes, and solving the underlying ordinary differential equations that arise. A number of commercial as well as open-source software programs exist for this purpose, but it is also common for scientists to implement their own solution using the differential equations solvers in Matlab, for example.

In order to provide an elegant and simple solution for the creation and simulation of Markov model-based ion channels, we present ModFossa, a simulator written in C++ with an easy-to-use Python interface. Several published models of ion channel gating have been reproduced using our simulator; these results, along with performance analysis are presented in the following sections. In Section 2 we present some background concepts related to ion channel modeling. In Section 3, we present the design and implementation of ModFossa. Results are provided in Section 4. Finally, we conclude with a discussion of applications and future work in Section 5.

2 Background

2.1 Biology background

To understand the function and behavior of ion channels, several points need to be discussed. The background comes from a wide disciplinary range, including biology, chemistry, thermodynamics, electricity, probability theory, and differential equations. We will only be able to present a brief overview here. For more information, the reader is referred to Hille's extensive book, *Ion Channels of Excitable Membranes* [9], and Keener and Sneyd's work on the applied mathematics of cellular physiology [12].

Before discussing the biology of cells and ion channels any further, a quick discussion regarding ionic solutions and the *electrochemical gradient* is needed. As any good student should know, ions are atoms or molecules that have a positive or negative charge due to differing numbers of protons and electrons. Specifically, anions have more electrons than protons, so they have a negative charge. Conversely, cations are "missing" electrons, and therefore have a positive charge.

Ions arise from a variety of natural processes, including the dissolution of salts in water. For example, when table salt, or more officially, sodium chloride ($NaCl$) dissolves in water, sodium cations with one positive charge (Na^+) and chloride anions with one negative charge (Cl^-) become dissociated and free to move around independently. Similarly, the salts potassium chloride (KCl) and calcium chloride ($CaCl_2$) dissolve into solutions of free K^+, Cl^-, and Ca^{2+} ions.

Biological cells are divided into two main groups: prokaryotic cells which include bacteria, and eukaryotic cells such as plant and animal cells. Prokaryotic and eukaryotic cells are both enclosed in a selectively permeable biological membrane known as the cell (or plasma) membrane. Eukaryotic cells also have a number of internal membranes enclosing their nucleus, mitochondria, and several other organelles. Prokaryotes on the other hand do not have a nucleus or any other membrane-bound organelles.

The cell membrane of both prokaryotes and eukaryotes, as well as the membranes of certain organelles all contain ion channels. From this point on, however, the focus will be on the cell membrane of eukaryotic cells.

The cell membrane separates the intracellular contents from the extracellular environment and is composed of a thin double layer of lipids about 7.5 nM thick [12].

An important characteristic of the cell membrane is that it is selectively permeable, meaning that it allows certain molecules to pass though in either direction in controlled quantities. For example, waste products are allowed to leave the cell, and certain sugars, acids, and other molecules can enter. Also affected by the membrane's selective permeability are ions, such as Na^+ (sodium), K^+ (potassium), Cl^- (chloride), and Ca^{2+} (calcium).

The transportation of molecules across the membrane can either be passive (requiring no net energy), or active (requires energy, such as ATP). An important active transport mechanism is the $Na^+/K^+ - ATPase$, which is also called the sodium-potassium pump for the sake of readability. This pump is a trans-membrane protein that uses the energy stored in ATP to pump Na^+ out of the cell and K^+ into the cell [12]. In addition to pumps, integral proteins called exchangers also facilitate active transport across the cell membrane. The $Na^+ - Ca^{2+}$ exchanger uses the energy gained from allowing Na^+ entry into the cell to remove Ca^{2+}. One may question why allowing Na^+ into the cell provides energy, while removing Ca^{2+} requires energy. The reason this is so is due to the electrochemical gradients arising from the ionic concentration levels that the cell maintains through active transport.

At the cell's resting potential, the concentration of Na^+ is much higher outside the cell than inside the cell, so Na^+ wants to enter because the electrochemical gradient is driving it into the cell. Ca^{2+} also a higher extracellular concentration, so it takes energy to remove it from the cell because it must be moved up, or *against* its electrochemical gradient.

The last transport mechanism that deserves mention are ion channels, trans-membrane proteins which facilitate the diffusion of select ion species down their electrochemical gradient. Ion channels are therefore referred to as passive transport mechanisms. As a simplification, an individual channel can be thought of as either open, or closed. In the open configuration, channels allow ions to pass through at speeds of 10^8 ions per second, near diffusion speed [3].

From the perspective of the whole cell, the large number ion channels together can be said to affect the *permeability* of the membrane to a certain ion species.

In summary, the cell membrane uses ion channels, pumps, and exchangers to maintain a certain concentration gradient for a number of ion species. The membrane's permeability to an ion species depends on the types of ion channels in that cell type, as well as their current states (open, or closed).

2.2 Modeling ion channels

The modeling of ion channels dates back over 100 years [14]. As is with many mathematical models, the motivations driving the development of ion channel models include: (1) striving to infer fundamental knowledge of the underlying physical processes, (2) matching experimental data to the model's predictions, (3) using the model to predict behaviors which are difficult to observe, and (4) developing common terminology and knowledge among researchers.

The interpretation of ion channels models during the pre-molecular era provided the primary source of information about channel structure [14]. However, current technology such as molecular biology and x-ray diffusion are beginning to prove atomic-resolution detail of bacterial as well as some mammalian channels, though the fraction of channels with known structure remains low [3, 14].

Hodgkin and Huxley's landmark model describing the ionic mechanisms behind the action potential of a squid giant axon provided the foundation for many modern practices in electrophysiology and ion channel modeling [10, 12]. Their model describes the voltage-dependent and time-dependent behavior of Na^+ and K^+ conductances using a coupled set of ordinary differential equations. While the Hodgkin-Huxley model is still used today due to its simplicity and low number of parameters, it does exhibit several shortcomings. These include, for example, a lack of connectivity between activating and inactivating gates in the Na^+ channel, as well as the premise that the inactivation gate can only close after the activation gate opens [8].

Markov models: Markov models for ion channels are an extension of the Hodgkin-Huxley formalism, and are good for modeling single channel data, gating currents, and drug interactions. Unlike Hodgkin-Huxley models, Markov models show the state dependence of activation and inactivation [7].

Theory: Ion channels can be modeled as continuous time Markov processes, where a simplification of the channels functional physical shape is represented by states in the Markov chain. A model can have several open, closed, deactivated, and inactivated states, as well as states which represent the binding of ligands. The rates connecting the various states are time independent kinetic rates which can be constant, or dependent on the membrane voltage, or intra- and extra-cellular ionic concentrations.

Criticism: Markov models are not without critics. In a one page perspective titled *Are rate constants constant?* Jones challenges the community to examine the benefits of time-dependent rate constants over Markov models [11]. Jones cites the findings by Uebachs [21] who reports that the rate of recovery from inactivation in a type of Ca^{+2} channel depends on the length of depolarization, which goes against the basic principle of chemical kinetics that rate constants remain constant in a constant environment [11].

Although Markov models are memoryless by definition, the time-dependent behavior described by Uebachs can be reproduced by having a long chain of open or closed states [11, 16]. Nevertheless, Jones concludes that time-dependent rate constants as used by fractal channel models, would provide a more explicit memory than multistate Markov models.

Fractal Models: Liebovitch has been the main proponent of fractal ion channel models for over 15 years [16, 17, 15]. One of his main arguments questioning the use of Markov models is founded in the body of evidence depicting ion channel proteins as complex structures with a continuum of states, exhibiting large and small motions over varying time scales. Fractal models as proposed by Liebovitch contain a large number of conformational states with kinetic rates connecting them. The rates, however, are linked and not independent as in Markov models. This has the effect of giving the channel "memory" [16].

An additional concern raised by Liebovitch is the use of exponential rate constants. He provides a friendly reminder to his colleagues that the fitting of experimental data to sums of exponentials is famously ill-conditioned [16, 1]. As confirmation he provides a quote from a numerical methods text which states that those who try to determine the parameters of such equations from experimental data "must be spanked or counseled. At the very least, keep them from obstructing Progress and the computer!" [1].

In their defense, other researchers have argued that Markov models with exponential rate constants and a small number of states fit the experimental data better than fractal models [18]. Nevertheless, fractal models continue as the number one challenger to Markov models.

3 Design Overview

In this section we present ModFossa, a fast and easy-to-use Python library for creating and simulating ion channel kinetics using continuous time Markov processes. A brief discussion of ModFossa's features, components, are provided here. For more details regarding the design of ModFossa the reader is referred to [5] and for the source code the reader is referred to [6].

For those interested, the name ModFossa was formed from two Latin words — mod, and fossa, which can be interpreted as *open channel* when written together.

Before we begin the design overview, Table 1 compares the basic features of each simulator, including our solution, ModFossa.

Table 1: Feature comparison of several ion channel simulators which use Markov models.

Name	Model	GUI	Scriptable	Plotting	Plot format	Simulation type	OS	License
IonChannelLab[4]	Markov	yes	no	yes, interactive	bitmap	numerical integration, Monte Carlo, Q-Matrix	Win.	Free, no source available
ChannelLab[19]	Markov	yes	no	yes	bitmap	numerical integration, Monte Carlo	Win.	Proprietary
QUB	HMM	yes	yes	yes, interactive	bitmap	numerical integration	Win., Mac, Linux	GPL
ModFossa	Markov	no	yes, from Python	yes	vector	numerical integration	Linux	GPL

3.1 Features

Python interface: ModFossa provides easy, yet versatile and powerful ion channel modeling through its novel Python interface. Python is well-used in the scientific and academic community; however, we are unaware of any Python libraries which facilitate the creation and simulation of ion channel kinetics. The popularity of Python as a research tool stems from its simple syntax and extensive standard library, and the availability of many high-quality open source numeric and scientific libraries. Additional qualities of Python include easy third party library installation; strong, dynamic typing; and an interactive interpreter.

ModFossa can also be used directly as a C++ library, allowing for more flexibility.

Easy model creation: ModFossa allows the user to define an ion channel's Markov model using states, rates, and connections. States and rates are referred to by their user-supplied names. This allows the user to create the model in any order. For example, connections can be defined before the definition of the states and rate constants.

States: States represent the various channel states as defined by a continuous time Markov chain. All states are referred to by unique name which is assigned by the user during creation. States are created as either non-conducting or conducting. Additionally, the gating of conducting states can by specified during construction. The gating parameter is used to simulate states that are partially conducting. For example, assigning a gating value of 0.5 to a state in a particular model may be used to simulate channel blocking by 50 percent.

Rates: Rates represent the kinetic rate constants which define the rate of transition between two states. Rates are referred to by unique name. ModFossa includes the following rate constant types: constant, Boltzmann voltage dependence, exponential voltage dependence, and ligand-gated. The equations for each rate constant are presented below.

Constant rate constant are defined by a single parameter, k, like so:

$$rate = k. \tag{1}$$

Exponential voltage dependent rate constants depend exponentially on the membrane voltage:

$$rate(V) = a * exp(k * V), \tag{2}$$

where V is the membrane voltage, exp is the exponential function, and a and k are parameters.

Boltzmann voltage dependence is defined using the sigmoid-like Boltzmann equation:

$$rate(V) = \frac{a}{1 + exp[(V - V_{0.5})/k]}, \tag{3}$$

where V is the membrane voltage, $V_{0.5}$ is the half-maximal activation voltage (given as a parameter), exp is the exponential function, and a and k are parameters. Taken from Angermann et al. [2].

Ligand-gated rate constants depend on the concentration of a particular ionic species, S, like so:

$$rate([S]) = k * [S]^n, \tag{4}$$

where $[S]$ is the concentration of ligand S, n is the ligand power, and k is a parameter.

Connections: Connections define a transition from one unique state to another using a specified rate constant. States can have multiple ingoing and outgoing connections. Rate constants may also be used in multiple connections.

Experiment definition: An experiment consists of a channel Markov model, a voltage protocol, and an optional concentration protocol.

Voltage protocol: Voltage protocols are a fundamental technique in electrophysiology used to measure the current response of ion channels. In the laboratory, a small electrode is inserted into the cell, through the cell membrane. The electrode, and therefore the cell membrane, is held constant at a desired voltage for a length of time, as defined by the voltage protocol. A typical voltage protocol is shown in Figure 1. As shown in the figure, the voltage protocol begins with a *hold stage* at -50 mV for 100 ms. Next, a 1 second long *stepped stage* is defined, which steps the voltage from -100 mV to 140 mV in 20 mV increments. The voltage protocol ends with another hold stage at -80 mV.

The voltage protocol shown in Figure 1 is easily defined in ModFossa by adding two hold stages, and one stepped stage. A ModFossa voltage protocol can have any number of hold stages, but only one stepped stage.

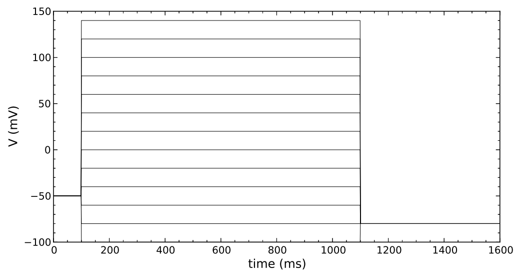

Figure 1: Example of a voltage protocol plot generated using ModFossa. This protocol starts with a hold stage at -50 mV for 100 ms. Next, a 1 second long *stepped stage* is defined, which steps the voltage from -100 mV to 140 mV in 20 mV increments. The voltage protocol ends with another hold stage at -80 mV.

Concentration protocol: A concentration protocol defines the concentration values (extracellular or intracellular) of a certain ligand. Like a voltage protocol, ModFossa's concentration protocols are defined using *hold stages*. However, there is no support for stepped concentration protocol stages. During simulation, the simulator will run the model through the entire voltage protocol using a single stage from the concentration protocol. Next, the concentration protocol will advance to the next stage, and the voltage protocol will be run using the new concentration value. This continues until there are no stages left in the concentration protocol.

Data analysis and plotting: ModFossa supports the creation of a number of useful plots using the Python plotting library, PyPlot. Unlike other simulators, the plots can be saved easily in a vector graphics format. Visual examples of each plot type are found in Section 4.

Model and experiment validation: To assist the user in creating a valid simulation, ModFossa provides validation methods which return detailed messages regarding any problems with the model and experiment definitions. Specific error conditions include: *no connections defined*, *max conductance not defined*, *rate constant not defined*, *state not defined*, *ligand not defined*, *no voltage protocols defined*, and *no experiment sweeps defined*.

3.2 Implementation

Development tools: C++ code was developed under Ubuntu Linux using Eclipse CDT. CMake was chosen as the build system. Unit tests for the C++ code were written using the Google Test framework. Having unit tests available aided the development process by instilling confidence in the correctness of the code, especially during refactoring. Google Test is included in the source tree, and is built by CMake during the build process. This removes the need for the user to have Google Test installed on the target computer.

Doxygen was used to generate documentation for the C++ code, while the Python documentation was generated using Sphinx. Generated documentation can be found in the Appendix.

Dependencies: Sundials (SUite of Nonlinear and DIfferential/ALgebraic equation Solvers) is a C library developed by Lawrence Livermore National Laboratory. ModFossa uses Sundials to solve the system of ordinary differential equations governing the ion channel state probabilities. Armadillo, a C++ linear algebra library provides easy-to-use matrix structures and linear solvers, and is also used by ModFossa. However, it would be ideal to remove the dependency on Armadillo in order to remove complications during ModFossas installation.

In order to run ModFossa's Python interface, two Python libraries are required: MatplotLib and Numpy.

4 Results

Several example models of varying complexity have been created and simulated using ModFossa. In this section we will present one such example. Visual results are provided in the form of plots generated by the code. Finally, an examination of ModFossa's computational performance is provided.

4.1 Model Description

In [2], Angermann et al. attempt to determine how phosphatase activity influences calcium-activated chloride channels in rabbit pulmonary artery myocytes (smooth muscle cells).

To simulate the behavior of the test and control groups, Angermann et al. created a Markov chain kinetic model based on the work by Kuruma and

Hartzell [13]. Slightly different parameters were chosen for the test and control groups in an attempt to make clear the difference in behavior between the two. The model, shown in Figure 2, consists of four closed states and three open states. Calcium binding occurs in the closed states with rates directly proportional to $[Ca]$. State C_1 is representative of the channel's state when no Ca^{2+} ions are bound to the receptor sites, while state C_4 represents the channel with all three Ca^{2+} binding sites occupied. The closed states with at least one binding site occupied can transition into open states with constant channel opening rates. Closed states with more occupied binding sites have higher values for the channel opening rates, giving the model its concentration-dependent behavior. Voltage dependent channel closing rates take the model from open to closed states.

In [2], the behavior of the channel in the presence of AMP-PNP was simulated by setting the gating variables of all three of the channel's open states to 1. Conversely, to simulate the channel in the presence of ATP, only channel O_1 was assigned a gating variable of 1, while the rest of the open states were given a value of zero. This reproduces the blocking effect by phosphorylation that the authors hypothesize as a possible explanation for the inhibition of I_{ClCa} while the channel is subjected to ATP. In addition to reducing the channel gating variables to model the effect of ATP, the authors also increased the magnitude of the channel closing rates and shifted the voltage dependence towards more positive potentials.

By altering the Markov model's parameters in an insightful manner, Angermann et al. were able to simulate the behavior of Ca^{2+}-activated Cl^- currents in the presence of both AMP-PNP and ATP.

4.2 Model implementation in ModFossa

The calcium activated chloride channel Markov model shown in Figure 2 was implemented in ModFossa. The parameter values were chosen to match the behavior of I_{ClCa} in the presence of ATP, and were modified slightly from Thibeault et al. [20].

4.3 ModFossa output

A chord conductance plot was generated using ModFossa (Figure 3). This plot shows channel conductance as a function of voltage, for 6 different concentration values. A chord conductance plot of Angermann et. al's measured experimental data is shown in Figure 4. As shown, the simulated data (Figure 3) match the experimental data (Figure 4) well.

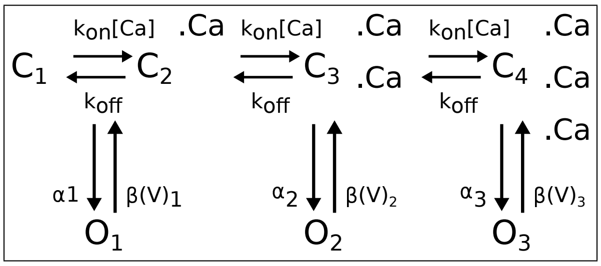

Figure 2: Angermann CaCl Markov model. The model has four closed states and three open states. State C2 represents the channel with a single Ca^{2+} binding site occupied, while state C4 represents the channel with all three binding sites occupied. Note that the model can only transition to an open state in the presence of at least one bound Ca^{2+}.

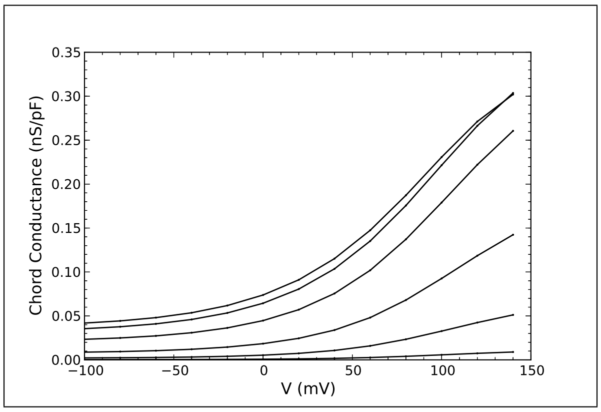

Figure 3: Simulated Angermann CaCl conductance at varying concentrations, as a function of membrane voltage. This figure was generated using ModFossa's *plotGvsV* method.

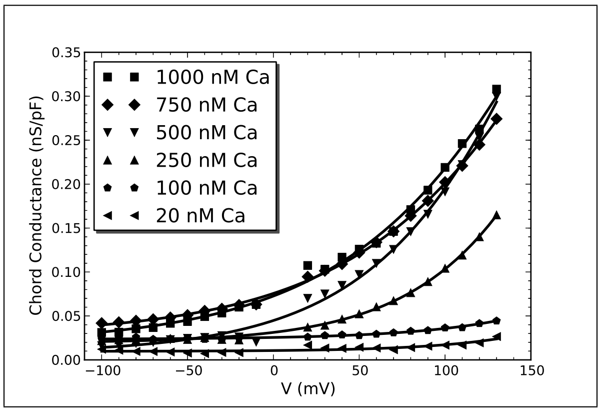

Figure 4: Measured data from [2] showing conductance as a function of voltage and concentrations. The data were fit to a Boltzmann function using the least squares method.

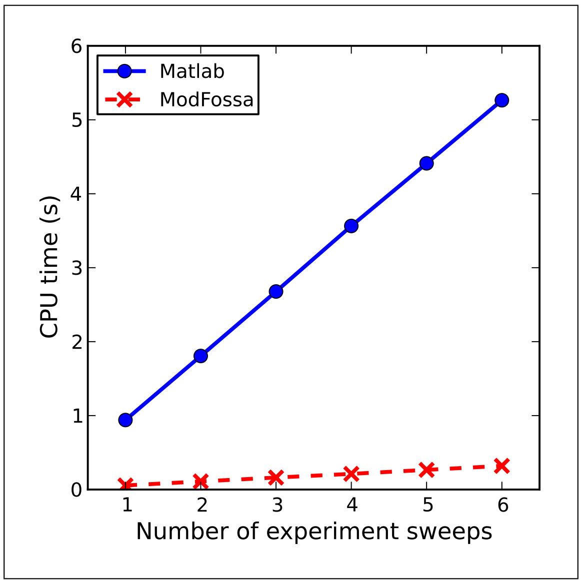

Figure 5: ModFossa runtime versus a Matlab implementation for the example model. The x-axis corresponds to the number of experiment sweeps performed at each data point.

The runtimes of ModFossa versus a similar Matlab implementation are provided in Figure 5. ModFossa is approximately 17 times faster.

5 Discussion and Future Work

5.1 Summary

We have identified and addressed a gap in ion channel simulation software by developing ModFossa, a Python library dedicated to the construction and simulation of ion channels based on continuous time Markov processes. While several recent software tools have been released for this purpose, none are Python libraries. Python is a popular tool among research scientists, therefore we hypothesize that a versatile and easy-to-use ion channel simulation library has the potential for providing a valuable service to those developing and testing Markov models of ion channels. In summary, ModFossa allows for the creation of channel models using simple Python syntax. Several rate constant types are supported, including exponential and Boltzmann voltage dependent rates, as well as ligand-gated rates. Voltage and current protocols are defined easily. Some common plots including IV curves, state probabilities, conductance versus voltage, and conductance versus ligand concentration are generated using a single function call. Finally, ModFossas core is implemented in C++ using the Sundials differential equation solver, providing the benefit of fast execution times.

5.2 Applications

ModFossa can provide rapid model development and testing for researchers with varying programming skill. A common task in ion channel model development is parameter searching, which requires the modeler to find the rate constant parameters which best fit the experimental data. Because ModFossa is usable from Python, existing machine learning libraries such as scikit-learn can be leveraged to perform parameter searches automatically. This situation is where ModFossas fast execution speed provides a great benefit, as it is not uncommon for modelers to leave parameter searches running for several days, or weeks.

5.3 Future work

Several ideas for additional development of ModFossa are presented here. First, some minor enhancements such as the addition of more sample models, improved customization of plotting functions (plot size, legend, colors, data range, etc.), and standardization of the data structures used in the results module would add value to the software by ensuring a smooth experience for new users. Additionally, creating and distributing a Debian package for ModFossa would ease installation significantly.

Several preliminary users have expressed interest in user-defined rate contant equations. A custom rate equation could be written by the user in Python using either a decorator function, or by inheriting from a base class. During simulation, the simulator would call the user's custom rate equation, passing in the current state of the simulation (membrane voltagte, concentrations, time, etc.), and use the returned value as the rate for whichever transitions the user specified. The performance hit due to calling a Python function from C++ during the simulation would have to be quantified, but it is not expected to be significant.

Lastly, the idea of offering ModFossa as a web service with a rich graphical user interface has been explored. Doing so would encourage use from a wider audience, particularly from those with limited or no programming experience. Such a web service could be implemented using a Python web framework such as Flask. A JavaScript visualization package, such as d3.js could provide rich graph visualizations of the model, as well as the stimuli definitions and simulation results. Finally, a web site would allow like-minded scientists to collaborate and easily share results with the community.

References

[1] Forman S Acton. *Numerical Methods That (Usually) Work*. Harper and Row, NY, 1970.

[2] Jeff E Angermann, Amy R Sanguinetti, James L Kenyon, Normand Leblanc, and Iain A Greenwood. Mechanism of the inhibition of Ca2+-activated Cl- currents by phosphorylation in pulmonary arterial smooth muscle cells. *The Journal of general physiology*, 128(1):73–87, 2006.

[3] Charlotte E Capener, Hyun Ji Kim, Yalini Arinaminpathy, and Mark SP Sansom. Ion channels: structural bioinformatics and modelling. *Human molecular genetics*, 11(20):2425–2433, 2002.

[4] Jose A De Santiago-Castillo, Manuel Covarrubias, Jorge E Sánchez-Rodríguez, Patricia Perez-Cornejo, and Jorge Arreola. Simulating complex ion channel kinetics with ionchannellab. *Channels*, 4(5):422–428, 2010.

[5] Gareth B. Ferneyhough. A python library for ion channel modeling. Master's thesis, Dept. of Computer Science and Engineering, University of Nevada, August 2013.

[6] Gareth B. Ferneyhough. A python library for ion channel modeling. `http://www.cse.unr.edu/brain/ionchannelmodeling`, 2013.

[7] Martin Fink and Denis Noble. Markov models for ion channels: versatility versus identifiability and speed. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1896):2161–2179, 2009.

[8] Meron Gurkiewicz and Alon Korngreen. A numerical approach to ion channel modelling using whole-cell voltage-clamp recordings and a genetic algorithm. *PLoS computational biology*, 3(8):e169, 2007.

[9] Bertil Hille. *Ionic channels of excitable membranes*. Sinauer associates Sunderland, MA, 2001.

[10] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

[11] Stephen W Jones. Are rate constants constant? *The Journal of physiology*, 571(3):502–502, 2006.

[12] James P Keener and James Sneyd. *Mathematical physiology*. Springer, 2009.

[13] Akinori Kuruma and H Criss Hartzell. Bimodal control of a Ca2+-activated Cl- channel by different Ca2+ signals. *The Journal of general physiology*, 115(1):59–80, 2000.

[14] David G Levitt. Modeling of ion channels. *The Journal of general physiology*, 113(6):789–794, 1999.

[15] Larry S. Liebovitch, Daniela Scheurle, Marian Rusek, and Michal Zochowski. Fractal methods to analyze ion channel kinetics. *Methods*, 24(4):359 – 375, 2001.

[16] LS Liebovitch. Testing fractal and markov models of ion channel kinetics. *Biophysical journal*, 55(2):373, 1989.

[17] Steven B Lowen, Larry S Liebovitch, and John A White. Fractal ion-channel behavior generates fractal firing patterns in neuronal models. *Physical Review E*, 59(5):5970, 1999.

[18] OB McManus, DS Weiss, CE Spivak, AL Blatz, and KL Magleby. Fractal models are inadequate for the kinetics of four different ion channels. *Biophysical journal*, 54(5):859–870, 1988.

[19] Synaptosoft. Channellab. `http://www.synaptosoft.com/Channelab/`, Accessed May 9th, 2013.

[20] Corey Thibeault, Michael Wiwchar, Marissa Huebner, Ramon J. Ayon, Iain A. Greenwood, and Normand Leblanc. Allosteric and state-dependent interactions can explain the paradoxical block and stimulation of native and TMEM16a-induced calcium-activated chloride currents by anthracene-9-carboxylic acid. In *2011 Experimental Biology meeting, Washington D.C.*, April 2011.

[21] M Uebachs, C Schaub, E Perez-Reyes, and H Beck. T-type ca2+ channels encode prior neuronal activity as modulated recovery rates. *The journal of physiology*, 571(3):519–536, 2006.