# Segregation of activation channels and their associated conductance levels

## Background

By separating the voltage activation of channels into distinct regions as defined in Alturki et. al, we can adjust cell parameters like conductance for channels that affect spiking without affecting other properties like resting membrane potential. This results in a more easily understood and tunable cell.

## Objective

Given passive properties [**Rin, tau, Vrest**] and a frequency vs current plot (FIR curve) of a biological neuron, can we generate a set of channel conductance [**g\_bar**] levels for a spiking neuron [**Leak, Na, K**]? A more realistic model neuron with low frequency oscillations?

The ultimate goal is to allow a user to input these parameters and we generate the associated model cell.

The project will be divided into 4 phases, producing 3 separate modules to be verified independently. The passive module consists of algebraic calculations to build the passive cell. The spiking module will add Sodium and Potassium channels to the cell and use a machine learning algorithm to determine optimal conductance to fit the provided FIR curve best. The low threshold oscillation (LTO) module will add additional channels to introduce additional neuron dynamics. Finally, we build an automated, easy to use GUI that builds the model cell for the user.

## Passive module

This module can be built entirely algebraically.

Cells assumes the form **C= g\_bar leak \* (ELeak – Vm)**

Variables defined by us: Cell length [**L**] (Cylinder length and diameter)

Variables defined by the user: Input resistance [**Rin**], [**tau**], [**Vrest**].

Cell property calculations:

Leak channel reversal [**ELeak**]:

* Value will be the supplied resting membrane potential
* **ELeak = Vrest**

Leak channel conductance [**g\_bar leak**]:

* Based on the segregation hypothesis presented in [Alturki et. Al], and assumption the activations are segregated here, the resting membrane potential will be entirely dictated by the leak reversal. Meaning
* Rin = 1/g\_bar leak will hold for spiking cells, therefor
* **g\_bar leak = 1/Rin**

Cell capacitance [**C**]:

* Tau is calculated by the resistance times capacitance (R\*C)
* In the cell it will be Capacitance/Leak Conductance (tau = C/g\_bar leak)
* **C = tau\*g\_bar leak**

**Spiking module**

Cells assumes the form **C= g\_bar leak \* (Vm - ELeak) + g\_bar Na\*m3h(Vm - ENa) + g\_bar K \*n4(Vm - EK)**

Extended m/n equation here with half-activation detailed.

This module will extend the cell beyond the passive module by inserting ion channels associate with spiking (Na/K). We then determine the Sodium and Potassium channel conductance using machine learning to fit an FIR curve supplied by the user.

Module Inputs – FIR curve, [e\_leak, g\_bar leak, capacitance for a more robust network capable of handling multiple cell types] **THINK MORE ON THIS**

Module Outputs – Na, K Channel conductance, v-halves, activation slope??, channel-reversal **THINK MORE ON THIS**

Data Generation:

A training dataset will be generated by numerous Neuron runs with randomized parameters within a parameter space consistent with biology. This will ensure that the machine learning algorithm does not output values that may be unrealistic.

Python based Neuron code should be used for good programming hygiene.

Data Generation Process:

1. Create a cell with parameters randomly generated within pre-defined space

Randomized parameters include:

* 1. Na, K channel conductance
  2. Na channel m gate half activation
  3. K channel n gate half activation
  4. [e\_leak, g\_bar leak, capacitance for a more robust network]

1. Run the cells while subjecting them to a set of current injections to generate the FIR curve
2. Cell’s parameters and FIR curve will then be saved as a single line in a dataset file.
3. Re-adjust the cell’s parameters randomly and automatically starting with step 1. Stop when the desired number of cells has been generated

**THINK MORE ON THIS**

The more cells we generate the better. 10,000+

Machine learning process:

1. Columns will be chosen to be input/output for the algorithm to be trained.
2. Several machine learning algorithms/configurations should be assessed for accuracy.

**THINK MORE ON THIS**

Algorithms to consider:

* Neural Networks
  + ANN - Simple 3 layer network
  + DNN - Deep 4+ layer network

## LTO (Low threshold oscillation) module

To be determined. The process should be similar just with added parameters.

## GUI

Thoughts:

Jupyter notebook with sliders and input fields can be used to generate plots and download links easily.

Hosting would be easiest using Binder.