

AUTOMATE DEVELOPMENT OF HODGKIN-HUXLEY MODELS

Computer/Software Requirements

A computer with Neuron, Anaconda3 and BMTool installed. (See Appendix B for BMTool installation)

Alternatives:

NeuroLab – Online Computational Neuroscience research training platform NO INSTALLATION ([Coming soon](#))

NeuroVM – Prebuilt computational neuroscience virtual machine (<https://tylerbanks.net/projects/neurovm/>)

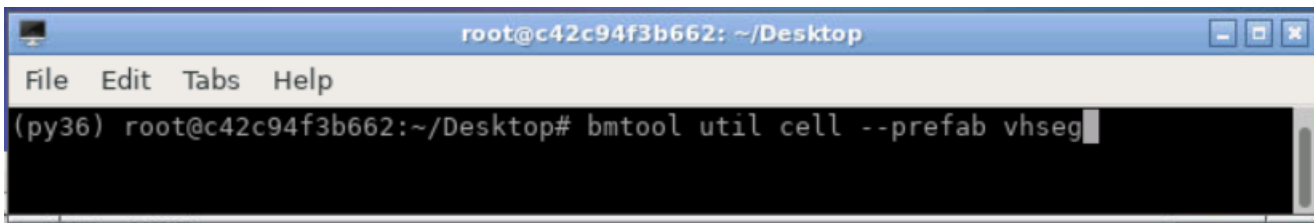
USER INPUT

Provide biological cell properties (passive properties: v_{rest} , R_{in} , τ , and F-I curve in the form of spike rate for different current values) and obtain a Hodgkin-Huxley model in NEURON code delivered directly to your email inbox.

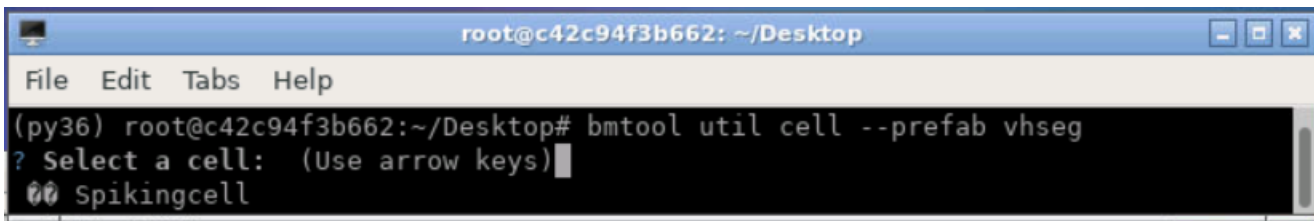
Getting Started

From the command line (terminal) execute

```
> bmtool util cell --prefab vhseg
```

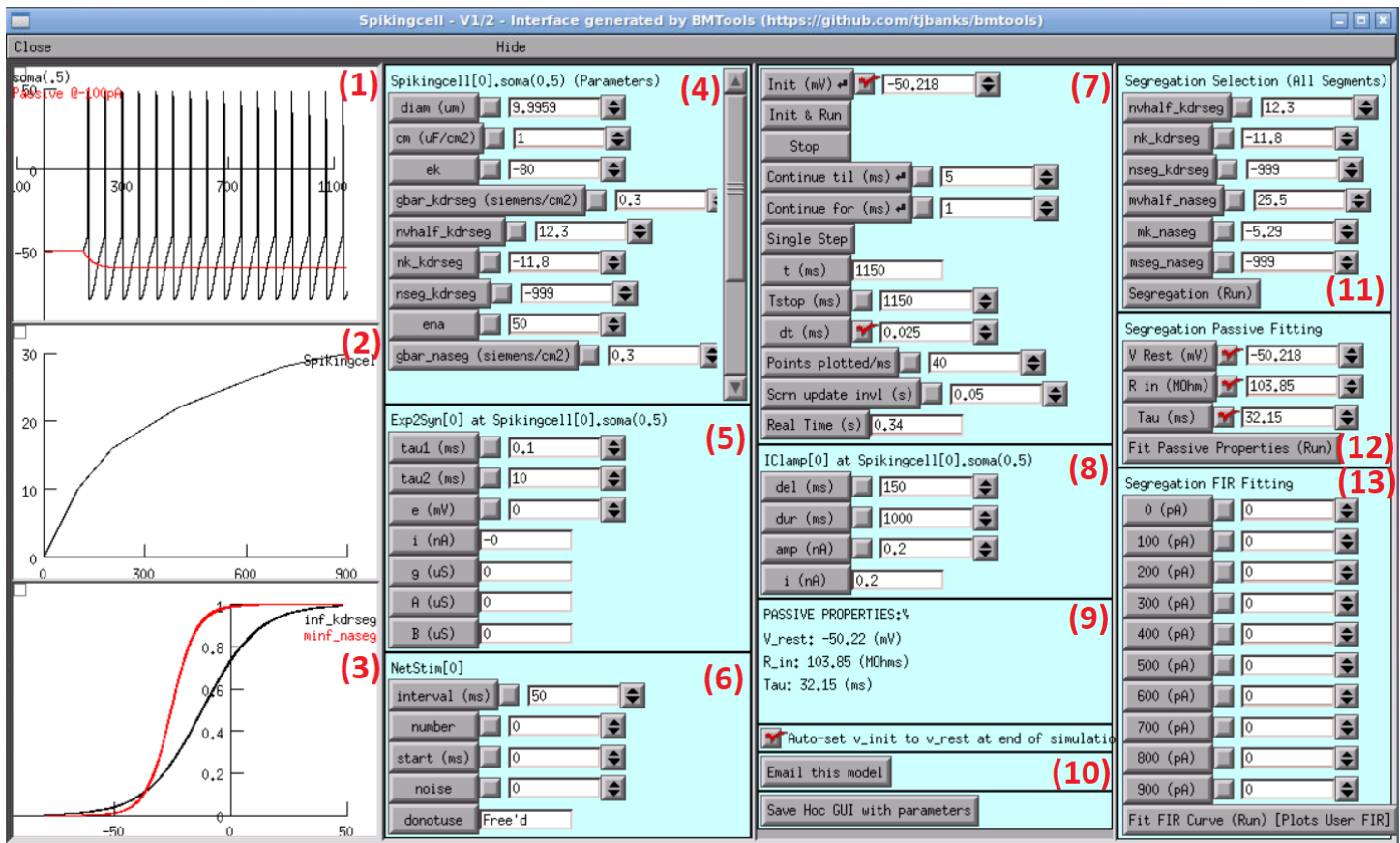


A collection of pre-made cells will be downloaded, compiled and presented for your selection. Select “Spikingcell” and press enter to get started with a simple sodium/potassium spiking cell.



You will then be presented with an interface to tune your new cell. The panels in the figure from top left to bottom right are as follows:

(1) Soma voltage trace, **(2)** Cell F-I curve, **(3)** Channel activation inf curves, **(4)** Soma parameter tuning – change channel density and reversal potentials here **(5)** Synaptic input channel – change reversal and time constants here, **(6)** Synapse control – edit the number and frequency of synaptic inputs, **(7)** Run control, **(8)** Current clamp input, **(9)** Calculated cell passive properties, **(10)** Additional functions – email yourself a copy of your tuned model as it’s displayed, **(11)** Channel $V_{1/2}$ properties – separate channel activation to simplify tuning, changes all section channels **(12)** Automatically adjust cell properties to a biological cell you may have observed, **(13)** Plot an F-I curve to target while you adjust your cell properties

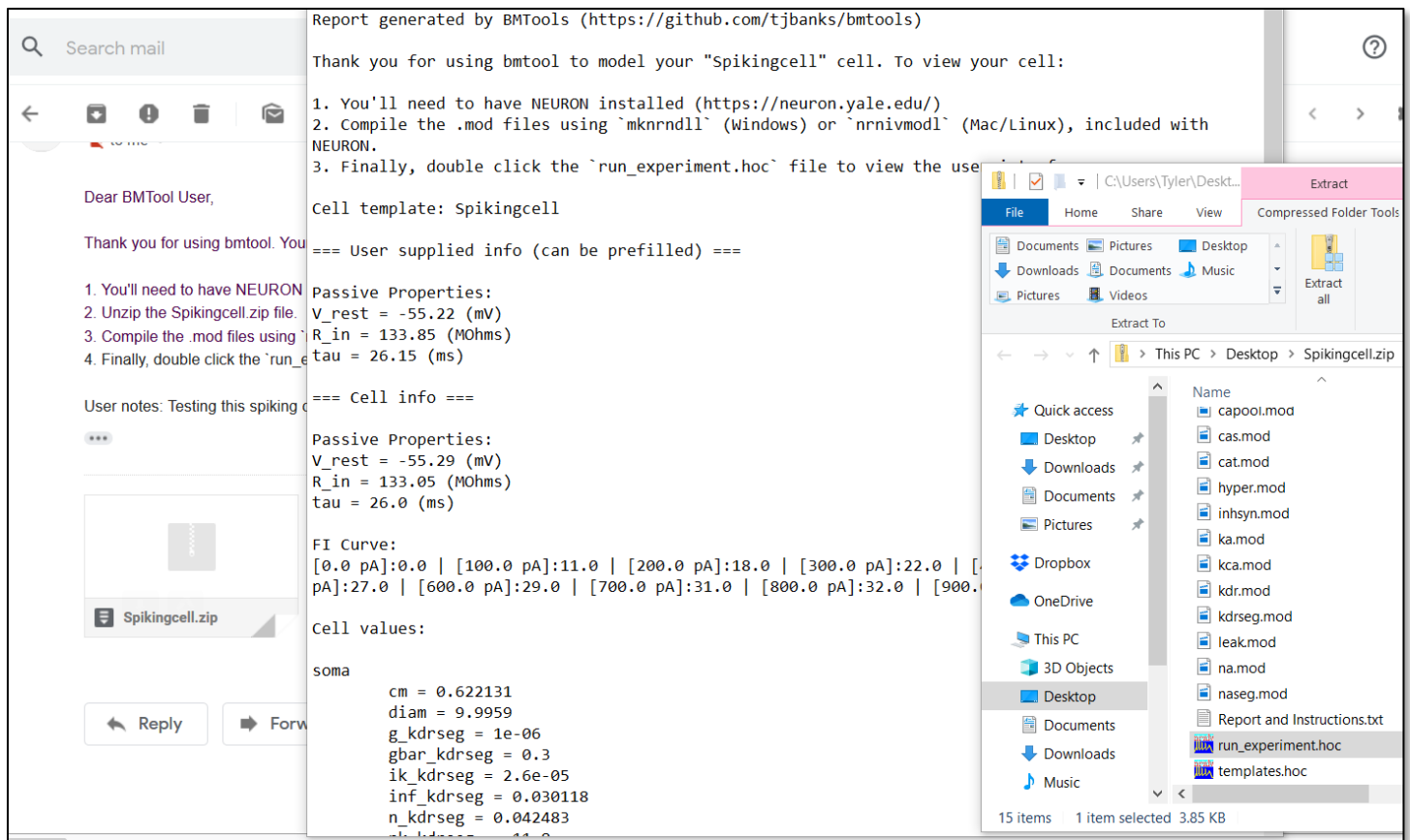


User input:

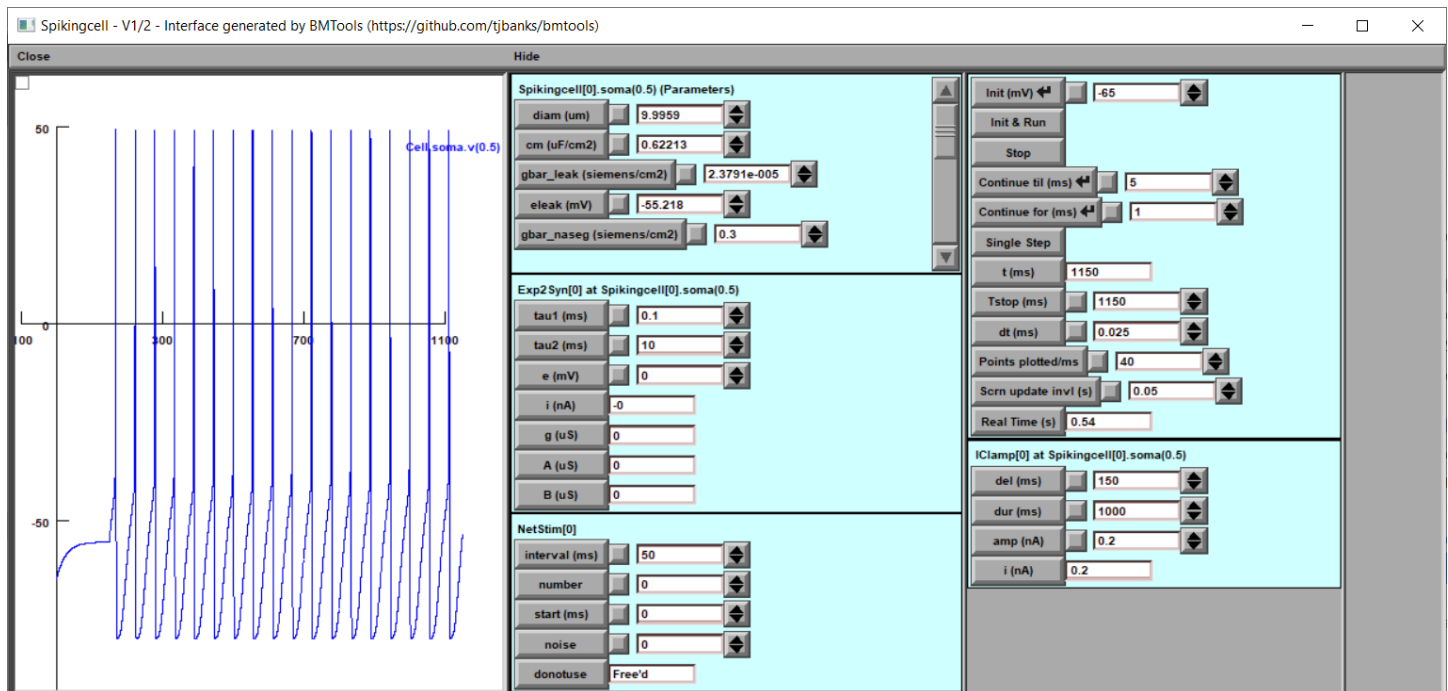
1. Enter your desired passive properties into box 12 (Segregation Passive Fitting)
2. Eg: V Rest: -55, R in: 50, Tau 60
3. Then click "Fit Passive Properties (Run)"
4. Cell parameters in box 4 (Soma parameter tuning) will change and the cell will re-plot. Note the passive properties change to best match your input in box 9.
5. Input your desired cell F-I curve in box 13, this will plot the curve in box 2 in blue. You can then use this as a reference while tuning your cell's properties.
6. Adjust you cell parameters from box 4, use the scroll bar for additional properties. Change the potassium reversal potential (ek), potassium conductance (gbar_kdrseg), sodium reversal potential (ena), sodium conductance (gbar_naseg) and others to achieve the desired properties for your cell.
7. Once you are satisfied with your cell click "Email this model" in box 10
8. Provide your email address and click ok, providing any notes in the popup box that follows



9. Check your email for your new model with a generated report, simplified tuning interface, and instructions on how to run the cell on your computer.



Email received, "Report and Instructions.txt" and zip folder file listing.



Simplified user interface (run_experiment.hoc)

APPENDIX A - THEORY

An automation tool used to highlight findings of the paper “Distinct current modules shape cellular dynamics in model neurons. Neuroscience 334:309-331” paper by Alturki A, Feng F, Nair A, Guntu V, Nair SS (2016) [\[link\]](#)

Preliminary Information

" ... We hypothesized that currents are grouped into distinct modules that shape specific neuronal characteristics or signatures, such as resting potential, sub-threshold oscillations, and spiking waveforms, for several classes of neurons. For such a grouping to occur, the currents within one module should have minimal functional interference with currents belonging to other modules. This condition is satisfied if the gating functions of currents in the same module are grouped together on the voltage axis; in contrast, such functions are segregated along the voltage axis for currents belonging to different modules. We tested this hypothesis using four published example case models and found it to be valid for these classes of neurons. ..."

SEGREGATION APPROACH: 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. [\(Alturki et al. 2016\)](#)

The approach is 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.

User Inputs

Passive	R_{in} , t , V_{rest}
Current Injection properties	F-I curve, Bursting, LTO & HTO
Synaptic Properties	Latency, Weight
Morphology	Can be downloaded from Allen Institute
Type of Channels & Properties of Channels	t_m, t_n, t_h , m_∞ , n_∞ , h_∞ (If available)
Any additional feature user may provide	

Segregation Procedure

STEP 1. Obtain the required neurobiological information from the user/neuroscientist (see table on previous page). This includes neuronal signatures to be matched and known current channels and their properties. If some of the channels and properties are unknown, we need to “fill them in” from information in the biological literature (see Appendix for a compilation of such information).

STEP 2. Make a plot of the various m , h and τ curves of the channels. Divide the plot into distinct regions/zones as defined in Alturki et. al (2016) and plot the curves to enable visual checking (Fig. 1).

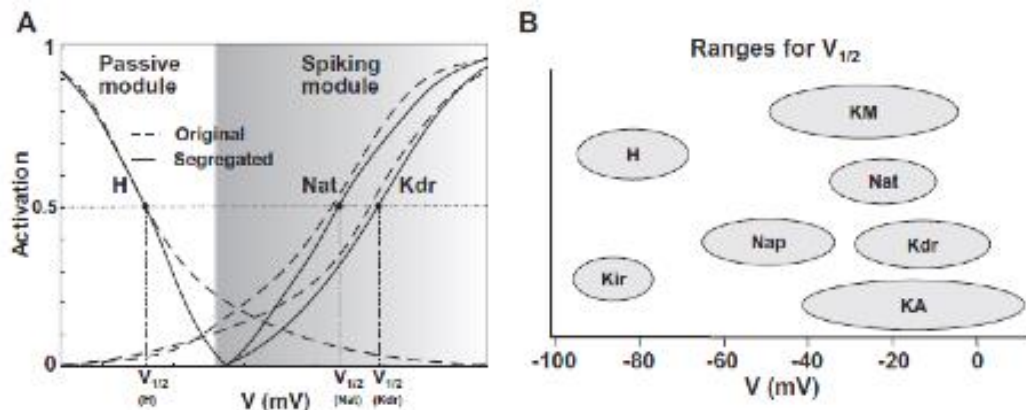


Fig. 2. Illustration of the segregation idea. (A) Implementing the segregation on a cartoon model that has two current modules: passive and spiking, and includes three ionic currents: H, Nat, and Kdr. The corresponding activation functions (dotted lines) are reshaped using the segregation approach (solid lines) to eliminate overlap between modules. (B) Ranges of half-activation voltages for several ionic currents. (adapted from Fig. 2.21 of Izhikevich, 2007).

This will enable adjustment of parameters such as conductance for channels that affect one module without affecting properties of the other module(s), considerably simplifying the tuning process compared to existing automated tuning methods. See table in appendix for range of parameters.

STEP 3. Design each of the modules sequentially, starting with the passive module. The passive module typically contains leak and possibly H channels (see plot generated in step 1 for segregation zones). 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.

Step 3-1. Passive Module

Module Inputs: Area, Input resistance $[R_{in}]$, $[\tau]$, $[V_{rest}]$.

Module Outputs: Ranges for cm , g_{leak} , E_{leak} from biology (Izhikevich, 2007)

Cell property calculations:

- Assuming V_{rest} is within the range for E_{Leak} , select $E_{Leak} = V_{rest}$

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 (since there is no hyper-polarization activated current such as H) Meaning
- $R_{in} = 1/(Area * g_{bar leak})$ will hold for spiking cells, therefore
- **$g_{bar leak} = 1/(R_{in} * Area)$**

Cell capacitance [C]:

- Tau is calculated by the resistance times capacitance ($R * C = R * Area * cm$)
- In the cell it will be Capacitance/Leak Conductance ($tau = cm * Area / g_{bar leak}$)
- **$cm = tau * g_{bar leak} * Area$. Make sure that cm is within the biological range**

We can use the equations above to calculate the output values

Step 3-2. Spiking Module (Work In Progress)

Cells assumes the form $C = g_{bar leak} * (V_m - E_{Leak}) + g_{bar Na} * m^3 h (V_m - E_{Na}) + g_{bar K} * n^4 (V_m - E_K)$

Module Inputs – Either 2D data of (I,F) or spline coefficients, $[R_{in}]$, $[tau]$, $[V_{rest}]$

Module Outputs – Ex: $gbar_Na$, $gbar_K$

Step 3-3. Bursting Module

For bursting part, we use the method from [\(Traub et al. 2003\)](#).

Module Inputs: Number of Spike per burst, Frequency of bursting

Module Outputs- Ex: $gbar_Nap$, $gbar_Cat$

An example of these modules can be found on [github](#).

Step 3-4. LTO Module

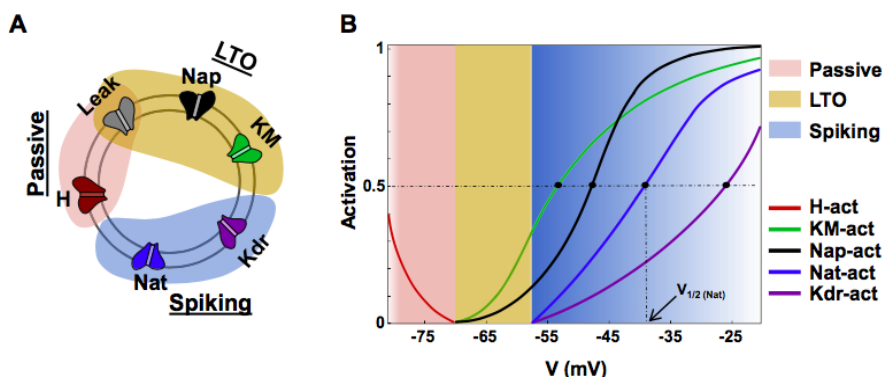


Fig. 1. Current segregation hypothesis. (A) cartoon layout showing current sets involved in the three distinct modules that implement the neuronal characteristics for an illustrative neuron with the following modules: passive properties, low-threshold oscillations, and spiking. (B) Activation curves for gating functions of the currents belonging to the various modules.

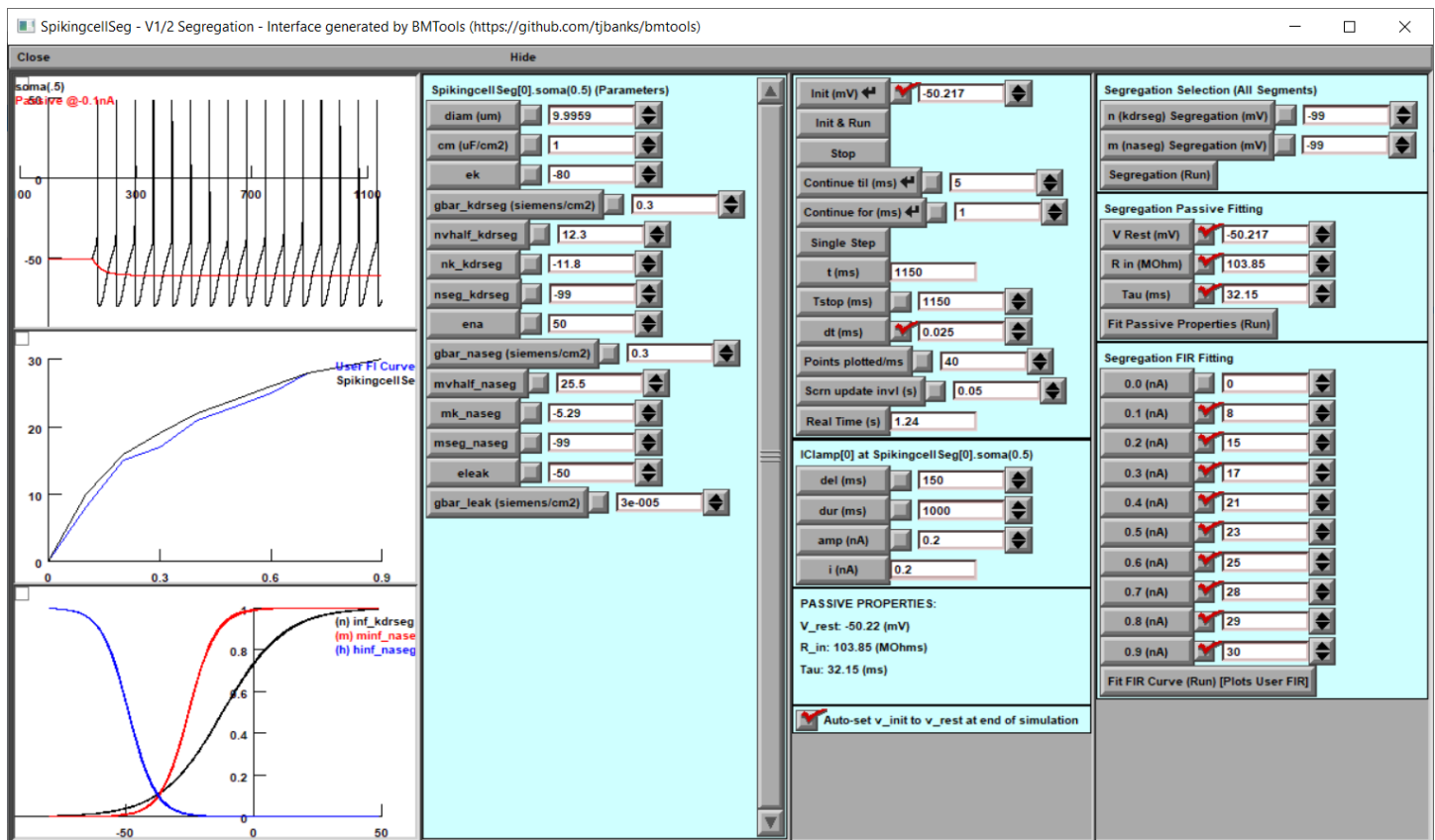
The biologically range for LTO is 0.5–9 Hz and 2–6 mV ([Pape et al. 1998](#)). We need to activate Nap and KM currents immediately above rest and segregated the H current to be active only below 3mV higher than Vrest. The half activation voltage ($V_{1/2}$) and slope factor (k) for Nap and KM need to be adjusted to ensure that the LTOs occurred around 59mV. We then iteratively tuned the maximal conductance for KM so that it dominated leak. Upon membrane depolarization (via current injection) to 59 mV.

Step 3-5. HTO Module

Both Nap and Nat were blocked to abolish LTOs and spiking. Maximal conductance values of KM and Kdr were tuned such that leak was not dominant, else it precluded oscillations. A strong KM current was found to prevent membrane depolarization to HTO levels when sodium channels are blocked. Also, since KM affects LTOs and Ca and Kdr affect spiking, tuning this module involved iteration with properties of LTO and spiking/ adaptation modules since the HTO module did not have unique currents of its own.

APPENDIX B – IMPLEMENTATION | BASE MODEL DEVELOPMENT

BMTool VHSeg Usage



Motivation

The segregation technique requires an intimate knowledge of the workings of NEURON mod files and cell tuning. While it is nearly impossible to provide an automated solution to segregation entirely, it is relatively easy to facilitate the tuning of a cell utilizing the segregation technique. Given a simulated cell with appropriately defined variables, BMTool can provide a window into the properties of that cell and help facilitate tuning toward the desired biological properties.

Obtaining the Tool

From your **Anaconda Prompt** (or preferred python console) run the following command:

```
pip install bmtool
```

Obtaining Later Releases

From your **Anaconda Prompt** (or preferred python console) run the following command:

```
pip install --upgrade bmtool
```

Example

1. Segregate Modules

- If you have been provided pre-segregated mod files you may be able to skip this step.
- The user will need to make the desired activation variables' "inf" value available as well as provide a variable to gate (eg: mseg below) the "inf" value.
- Example case – Fast spike generating Na channel (Not all mod files are the same, this may not work for all cases)

na.mod

```
1 : Fast spike generating Na current channel
2
3 NEURON {
4     SUFFIX na
5     USEION na READ ena WRITE ina
6     RANGE gbar, g
7     RANGE minf, hinf, mtau, htau
8     RANGE ina, mseg
9 }
10
11 UNITS {
12     (mA) = (milliamp)
13     (mV) = (millivolt)
14 }
15
16 PARAMETER {
17     gbar (siemens/cm2)
18     mseg = 999
19 }
20
21 ASSIGNED {
22     v (mV)
23     ena (mV)
24     ina (mA/cm2)
25     minf
26     hinf
27     mtau (ms)
28     htau (ms)
29     g (siemens/cm2)
30 }
31
32 STATE {
33     m h
34 }
35
36 BREAKPOINT {
37     SOLVE states METHOD cnexp
38     g = gbar*m*m*m*h
39     ina = g*(v-ena)
40 }
41
42 INITIAL {
43     rate(v)
44     m = minf
45     h = hinf
46 }
47
```

```

48 DERIVATIVE states {
49     rate(v)
50     msegment(v)
51     m' = (minf-m)/mtau
52     h' = (hinf-h)/htau
53 }
54
55 PROCEDURE rate(v (mV)) {
56     UNITSOFF
57     minf = 1.0/(1.0+ (exp ((v+25.5)/(-5.29))))
58     mtau = (2.64 - 2.52/(1.0+ (exp ((v+120)/(-25)))))
59     hinf = 1.0/(1.0+ (exp ((v+48.9)/(5.18))))
60     htau = (1.34/(1+ (exp ((v+62.9)/(-10))))) * (1.5 + 1.0/(1.0 + (exp
61 ((v+34.9)/(3.6)))))
62     UNITSON
63 }
64
65 : Segmentation functions
66
67 PROCEDURE msegment(v){
68     if (v > mseg){
69         minf = 0
70     }
71 }

```

d. Mod file modification procedure:

- i. Review at the state variables in the STATE block (line 33) – m and h are likely activation or inactivation variables
- ii. In the DERIVATIVE block – verify the inf values (lines 51 and 52) in the standard ($x' = (x_{inf} - x)/\tau$) equation are variables (to be plotted) (NEURON block (line 7))
- iii. In the location where the “inf” variables are set – take note of the “k” value or slope. A negative value indicates this is likely an activation variable, while a positive value indicates an inactivation variable
- iv. You may elect to create additional vhalf and k variables by creating a separate parameter (not shown) – for minf vhalf is 25.5 and k is -5.29
- v. A simple gating procedure at the end of the file (lines 67-71) will ensure that values less than the desired segregation point will result in minf = 0. Call this procedure before providing the derivative update (line 50)

e. Compile your mod file (nrnivmodl or mknrndll)

f. From your Anaconda prompt run the bmtool wizard

```
bmtool util cell vhseg
```

g. If you have more than one hoc file in your directory – specify your cell template file manually:

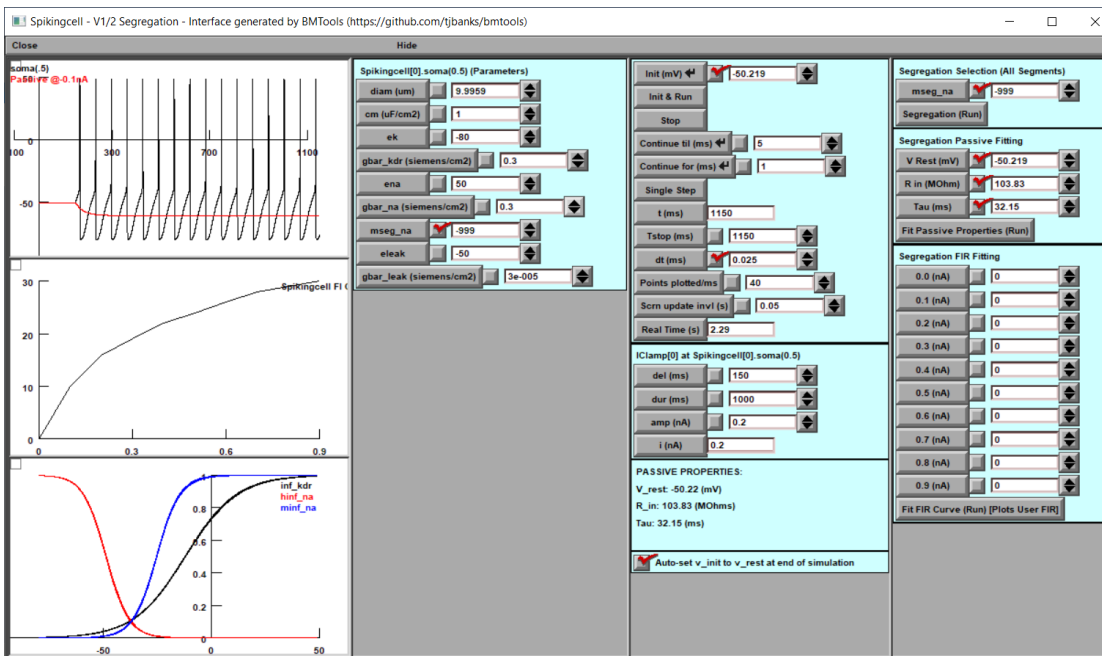
```
bmtool util cell --hoc cell_template.hoc vhseg
```

h. Select the appropriate variables by following the wizard’s prompts

```
(py36dev1) C:\Users\Tyler\Desktop\git_stage\two-cell-hco>bmtool util cell --hoc cell_template.hoc vhseg
? Select a cell: (Use arrow keys)
  Adaptationcell
  Burstingcell
  HCOcell
» Spikingcell
  Templatecell
```

```
? Select inf variables to plot (space bar to select): (Use arrow keys to move, <space> to select, <a> to toggle, <i> to invert)
  • inf_kdr
  • hinf_na
» • minf_na
  o g_kdr
  o gbar_kdr
  o ik_kdr
  o n_kdr
  o tau_kdr
  o g_na
```

```
? Select segregation variables [OR VARIABLES YOU WANT TO CHANGE ON ALL SEGMENTS at the same time] (space bar to select)
: (Use arrow keys to move, <space> to select, <a> to toggle, <i> to invert)
» • mseg_na
  o g_kdr
  o gbar_kdr
  o ik_kdr
```



i. The wizard can also be skipped entirely and run in a single command:

```
bmtool util cell --hoc cell_template.hoc --template Spikingcell vhseg -
infvars inf_kdr,minf_na,hinf_na -segvars mseg
```

j. A more advanced example case where additional sections can be plotted in additional windows (--othersec dend[0],dend[1]) and the leak conductance and reversal potential variables are specified manually (--gleak gl_ichan2CA3 --e leak el_ichan2CA3)

```
bmtool util cell --template CA3PyramidalCell vhseg --othersec
dend[0],dend[1] --infvars inf_im --segvars gbar_im --gleak gl_ichan2CA3 -
-e leak el_ichan2CA3
```

2. Utilizing the Window functions
 - a. Explanation of window widgets:

Window 1

Column 1

Widget 1 - Soma Voltage (Or user selected section)

Widget 2 - FIR plot (n additional cells will be created with the same parameters and run at the same time to generate curve realtime)

Widget 3 - Plot activation inf variables vs voltage to show segregation

- a. Plot vertical lines where segregation occurs for passive, hto, lto, etc.

Column 2

Widget 1 - A custom built nrnsecmenu that controls all cells' parameters simultaneously (Main and FIR generating cells) [will be made general enough to allow for multiple sections on the same cell]

Column 3

Widget 1 - Control menu (Init & Run etc.)

Widget 2 - Current clamp for main cell

Widget 3 - Passive properties calculated

Column 4

Widget 1 - Custom functions for segregation

1. Module segregation (passive, hto, lto)
 - a. Input field for voltage segregation line
 - b. A variable specified (segvars) will change all segments values (useful for unified reversal too)
2. Fit passive properties
 - a. Input fields (V_{rest} , R_{in} , τ)
 - b. Submit Button -> calculates and sets appropriate leak channel parameters, re-runs simulation
3. Fit FIR Curve (Future work)
 - a. Input field (current1:spikes1,current2:spikes2...)
 - b. Button "Run fitting technique"

To be determined. Likely options:

 - Genetic algorithm is run, cycles through simulations, displaying best fitting cell so far
 - c. Stop fitting algorithm

3. Runtime

- a. The window will run twice to obtain the most accurate V_{Rest} , R_{in} and τ values, adjusting v_{init} automatically

- b. Adjust the segregation variables in the top right corner
- c. Specify the desired Passive properties beneath
 - i. Click Fit Passive Properties
 - ii. The appropriate calculations take place and the cell is updated and ran
 - iii. Passive properties are re-calculated and displayed in the bottom of the 3rd column
 - iv. They will be closest when segregation is successful
- d. Enter your desired FI Curve parameters and re-run the simulation
 - i. Currently, the user specified curve will simply be plotted to provide a reference when tuning conductance variable manually. There may come a time when genetic algorithms will assist in tuning the cell and the FI curve specified will serve as input.

Bmtool VHSeg Help

```
> bmtools util cell vhseg --help
```

```
Usage: bmtool util cell vhseg [OPTIONS]
```

Alturki et al. (2016) V1/2 Automated Segregation Interface, simplify tuning by separating channel activation

Options:

```
--title TEXT          Print all debug statements
--tstop INTEGER       Starting FIR Curve amps (default: 0)
--debug              Ending FIR Curve amps (default: 1000)
--fminpa INTEGER      Increment the FIR Curve amps by supplied pA (default:
--fmaxpa INTEGER      100)
--fincrement INTEGER  Specify the inf variables to plot, skips the wizard.
--infvars TEXT        (Comma separated, eg: inf_mech,minf_mech2,ninf_mech2)
--segvars TEXT        Specify the segregation variables to globally set,
--segvars TEXT        skips the wizard. (Comma separated, eg:
--segvars TEXT        mseg_mech,nseg_mech2)
--eleak TEXT          Specify the eleak var manually
--gleak TEXT          Specify the gleak var manually
--othersec TEXT       Specify other sections that a window should be
--othersec TEXT       generated for (Comma separated, eg: dend[0],dend[1])
--clampsec TEXT       Specify sections that a current clamp should be
--clampsec TEXT       attached. Root section will always have a clamp.
--clampsec TEXT       (Comma separated, eg: dend[0],dend[1])
--synsec TEXT         Specify sections that a synapse should be attached.
--synsec TEXT         Exp2Syn default, unless --syntype specified. (Comma
--synsec TEXT         separated, eg: dend[0],dend[1])
--syntype TEXT        Specify the synapse mechanism that will be attached to
--syntype TEXT        the cell (Single type)
--synloc TEXT         Specify the synapse location (Default: 0.5)
--help               Show this message and exit.
```