ModelDB README

L2/3 Pyramidal Neuron: f–I Curves and K^+/Ca^{2+} Channel Kinetics

Dr. Suraj Honnuraiah
Harvard Medical School
suraj_honnuraiah@hms.harvard.edu

1. Overview

This ModelDB entry contains all HOC, MOD, and morphology files required to reproduce:

- Firing-rate (f–I) curves of a biophysically detailed Layer 2/3 pyramidal neuron.
- Potassium-channel voltage-clamp kinetics (A-type, D-type, M-type, delayed-rectifier, etc.).
- Subthreshold current–voltage (I–V) characterizations used in the Supplementary Figures.

All simulations are implemented in **NEURON 8.x** using the HOC interpreter and compiled MOD files.

2. Directory Structure

```
/ (model root)

 Firing_rate_L2_3_ZM_Optimal_CFA.hoc     <-- Main full-cell f-I simulation
 j8.hoc                                   <-- L2/3 pyramidal morphology

 *.mod                                    <-- Ionic mechanisms
    kv.mod, km.mod, kd.mod, kaprox.mod,
    kca.mod, ca.mod, CaT.mod,
    cad.mod, cad2.mod, SlowCa.mod,
    na.mod, child.mod, childa.mod

 K3St.hoc, KD.hoc, KM.hoc, KA.hoc,
    Kaprox.hoc, Kadist.hoc, Ksoma.hoc       <-- Single-compartment K+ clamp scripts

 Subthreshold_IV_Plots.hoc                <-- Subthreshold I-V analysis
 Display.ses                              <-- Optional NEURON GUI session
```

3. Requirements

- NEURON 8.x (or later) with support for `nrnivmodl` / `mknrndll`.
- All HOC, MOD, and morphology files placed in the same working directory.
- Ability to run the platform-specific NEURON `special` executable.

4. Compilation Instructions

Before running any simulations, compile *all* mechanism (`.mod`) files.

Linux / macOS

From the model root directory, run:

```
nrnivmodl
```

Windows / macOS (GUI)
Drag the folder containing the `.mod` files onto the NEURON application **mknrndll**, or invoke `mknrndll` manually. Compilation will create a directory such as:

```
./x86_64/special
```

All simulations should be started using this compiled `special` binary.

5. Running the Main f–I Simulation

To reproduce the full firing-rate curves, run:

```
./x86_64/special Firing_rate_L2_3_ZM_Optimal_CFA.hoc
```

This script:

- Loads the L2/3 morphology file `j8.hoc`.
- Inserts all Na^{+}, K^{+}, Ca^{2+}, Ca-dependent, and passive mechanisms.
- Applies somatic current injections over a range of amplitudes (default: 0 to 1.75 nA).
- Sweeps selected K^{+} conductances (default: M-type, gKm) in an *outer* loop.
- Uses APCount to record spikes for each (current, conductance) condition.
- Produces one text file per conductance level with:

```
<current (nA)>   <spike_count>
```

These output files together reconstruct the f–I curves in the Supplementary Figures.

6. Adjusting Current Steps and Ion-Channel Sweeps

The main HOC file, `Firing_rate_L2_3_ZM_Optimal_CFA.hoc`, contains two key loops:

Inner Loop: Current Injection Sweep
Controls the range and resolution of somatic current steps used to build the f–I curve.

- You may change the minimum and maximum injected current.
- You may adjust the step size (e.g., from 0.05 nA to 0.02 or 0.1 nA) to increase or decrease sampling density.

Outer Loop: Ion-Channel Conductance Sweep
Controls which ion-channel conductance is systematically varied across simulations.

- By default, the somatic M-type K^{+} conductance (gKm) is swept.
- Users can instead sweep:
  - Delayed-rectifier K^{+} (Kv),
  - D-type K^{+} (KD),
  - A-type K^{+} variants (KA, Kaprox, Kadist),
  - Ca^{2+}-activated K^{+} (KCa),
  - or any other conductance defined in the MOD files.
- To do so, replace the conductance variable in the outer loop with the desired channel variable.

This design enables systematic exploration of how different ion channels shape the f–I relationship.

7. Potassium-Channel Voltage-Clamp Simulations

Single-compartment voltage-clamp simulations are provided via:

- `K3St.hoc`
- `KD.hoc`
- `KM.hoc`
- `KA.hoc`
- `Kaprox.hoc`
- `Kadist.hoc`
- `Ksoma.hoc`

Each script:

- Creates a 20 μm $\times$ 20 μm isopotential soma.
- Inserts exactly one K^+ mechanism for isolated kinetic analysis.
- Applies a standardized three-phase SEClamp protocol (baseline $\rightarrow$ command voltage $\rightarrow$ baseline).
- Runs at $\sim$35°C with $dt \leq 0.025$ ms (values set internally).
- Records membrane voltage, K^+ conductance, and gating variables.

These traces yield the activation curves and time courses used in the Supplementary K^+ channel figures.

8. Subthreshold I–V Analysis

`Subthreshold_IV_Plots.hoc` performs subthreshold current–voltage characterizations of the same L2/3 model:

- Computes passive and active I–V curves.
- Can be used to extract input resistance and sag (if Ih is present).

Run with:

```
./x86_64/special Subthreshold_IV_Plots.hoc
```

9. Notes and Troubleshooting

- All `.mod` files must be compiled together; missing mechanisms will cause NEURON errors.
- Temperatures, time steps, and simulation durations are set inside individual HOC scripts.
- The model is deterministic (no stochastic gating), so repeated runs with the same parameters yield identical results.
- If errors occur, check:
  - that you are using the compiled `special` executable,
  - that all mechanisms listed in this README are present and compiled,
  - and that `j8.hoc` resides in the working directory.

10. Citation

If you use this model, please cite:

Honnuraiah et al. (2025). *Non-reciprocal callosal projections and input gradients underlie interhemispheric communication in binocular visual cortex*. Cell Reports.