**Semi-automated Electrophysiology Analysis: Instructions**

**06/13/18: Beta version ready.**

By Hanbing Song, advised by Christina Weaver

Code included in this bundle:

Test\_ana: Main entry code.

Test\_RN: Analysis for RN only.

Also local procedures:

AP\_ana.m: Analyze AP measurements.

createFit.m: Tau fitting

fix\_ephys\_jumps\_simple.m: Fix RN offsets

get\_dVdt.m: take derivatives of Vm.

Hilo\_ana: AP analysis for High/LowRN analysis

Output.txt: Main output. For each time running the analysis code, new results will be outputted appending at the end of this file.

Output\_RN.txt: Output for RN only.

[Note 061318: The RN procedure has been separated for faster usage. Other features can be separated as needed. ]

**Instructions:**

1. Set the ‘path’ variable in MATLAB where executable scripts are found: This should only need to be done once.

You need to tell MATLAB where to look for these analysis scripts. With MATLAB open, click on “Set Path” (top right of screen). The Set Path menu will pop up. Click on “Add Folder…”. Select the folder containing all these electrophysiology scripts. Then click “Save”, and then “Close”.

1. For each neuron that you will analyze, create a new folder with the same name as the neuron itself. Then make sure that you have gone into FitMaster and exported the appropriate protocols into separate MATLAB files, named as follows:

* Mar10IR1c-RN.mat: The file for the RN protocol (which will also be used for tau fitting)
* Mar10IR1c-HighRn.mat: HighRN protocol and sag
* Mar10IR1c-LowRn.mat: LowRN protocol (These High & Low files will also be used for AHP analyses and spike counting / F-I curve creation)
* Mar10IR1c-ramp.mat: Rheobase protocol
* Mar10IR1c-AP.mat: AP analysis (200ms protocol).

**Important:** the name of the folder and the basename of all of these files should match – for example, ‘Mar10IR1c’ in both places – not “Mar10IR1c” in one place and “MAR10ir1C” in another. For the instructions below, we will call this “NEURONNAME”.

**Note:** If only the RN protocol will be run, only the “\*-RN.mat” file is needed.

1. Create a new folder where all of the analysis results will be written. This can be the same one where the MATLAB .m files are, but does not have to be. (We will call this “NeuronEphysResults” in this Instructions document.) Move all the neuron data subfolders here.
2. Edit the MATLAB scripts to specify which files should be analyzed. The script that will run all the neuron analyses is called test\_ana.m; to run only the RN analyses, it is test\_RN.m . Go into this file, and edit the cell array variable named ‘Neuronlist’. Then save the file. This variable be changed to include all the neurons that should be analyzed, e.g.

Neuronlist={'Mar10IR1c','Mar10IR1d','Apr3IR2e'};

will analyze Mar10IR1c, then Mar10IR1d, and finally Apr3IR2e.

Note: A future version of the code will allow the user to choose the neuron folders from a GUI. This manual editing is a temporary step.

1. Run the analysis:

Change into the “NeuronEphysResults” directory (you can use ‘cd’ at the command line, or the “Current Directory” window). If you have the script editor window open, you can just click “Run” (green triangle). Or you can type the script name at the MATLAB command line (either test\_ana or test\_RN).

The following user interaction is required: For each neuron, select which traces of RN that should be fixed (click to choose one and command+click to include others. If fitting all, just pick select all). A figure will be displayed to help you.

The following output is generated:

Output.txt: a tab-delimited .txt file containing the results of the analysis. (If only the RN protocol is run, this file is called output\_RN.txt.) Results from each subsequent neuron analyzed will be appended as a separate line to the end of this file, beginning with the date the neuron was analyzed. This file can be imported easily into Excel.

Then for each neuron analyzed, the following output is generated:

NEURONNAME\_APoutput.txt: Shape analyses from each AP data from the AP trace we picked for analysis. The last line is the stats of such measurements (average, STD and such).

NEURONNAME\_High.txt: For each trace that generates APs in the \*HighRN.mat protocol, this file reports AP\_num, delay, mean\_ISI, std\_ISI and CV\_ISI (CV equals the standard deviation divided by the mean, and is a measure of regularity), AHP, followed by the spiking time of each AP.

NEURONNAME\_Low.txt: Analogous to the HighRN output above, but for the \*LowRN.mat protocol.

A new folder NEURONNAME \_figs, containing all the figures accompanying all the measurements. High\_RN and Low\_RN are the raw traces, ISI, F-I curves from HighRN and LowRN protocol. High/LowRN\_\*\*pA figures are the traces marked with fAHP and sAHP. \*.fig files (larger) are opened in matlab for zoomin/out. .png files are for preview. [In the future, if more debugging/protocol changes are required, the .fig file (native MATLAB figure file) can be used to review the data analyzed .]

1. **Check your work! Open up the figures generated for all of the neurons you analyzed, and verify that the measurements seem to be correct**. If anything looks wrong, try the analysis again, and/or contact Hanbing and Christina.

**Analyses Performed**

Here are some details about the analyses performed.

1. **RN fitting and Vr estimation**

First the data for the RN analysis and tau fitting are corrected by removing the ‘jump’ occurring at the start of the current step. This is done by deleting the first few time points right after the step onset. Then all data are realigned to the same averaged value of the membrane potential before the pre-pulse.

Method for RN fitting: Averages the membrane potential Based on the original 9 traces, After correcting the initial offset and re-aligning the baseline, the average membrane potential is computed within the 85%-95% of the 200-ms time window for all current steps. Linear regression to compute RN (slope of I-V curve) and Vr (corresponding V under 0 pA injection) is done first by fitting to all data. Then the user chooses which traces to use, and the regression is repeated.

1. **Tau fitting**

Search for the -10 pA step (usually the fourth trace) after fixing/alignment and use exponential function for fitting. This part might need more testing (getting the right fitting region).

1. **HighRN protocol**

For all the current steps, if no AP is generated, then plot everything altogether.

If APs are generated, plot the raw traces, identify the time of each AP (INCLUDING doublets!!) to count the peak time and ISI. Store individual AP time in separate files.

Mark the fAHP point as the first local minimum of Vm after peak with 20ms (needs testing). Mark the sAHP point as the last minimum Vm at the end of the current step.

Compute sAHP (sAHP point minus Vm during the pre-pulse) and mAHP (minimum between two AP peaks – Threshold before the former AP).

The F-I curves are computed in two ways:

1. Compute frequency as 1000/ISI, then take the average.
2. Count the number of spikes, then divide by 2 sec. (This is the current Luebke lab approach).

Also from HighRN – computing the Sag: Use the first trace from the HighRN protocol. Find the global minimum Vm point and the average of 85%-95% of current step Vm response as the stable Vm. The difference is the sag.

1. **Same as HighRN analyses, but using the LowRN protocol data.**

Note: Sometimes the baseline current is not 0, so +130pA would result in the +120pA in the current trace.

Solution: Approximate the current to the nearest standard current (+120 -> +130pA).

For 3 and 4, if No AP generated, put 0 as AP count to keep output format consistent.

1. **AP analysis:**

Use the first trace generating > 3 Aps in the 200ms protocol. Isolate each AP from the threshold point excluding doublets. Measure AP Threshold, Peak, amplitude, rise, decay and half duration. Write data from the second AP to the output file, but store all the AP data in the \*\_APoutput.txt files. Example AP from the figure shows what we measure (second AP).

1. **Rheobase**

By default, use the first trace of ramp protocol. Compute the time of the first AP and the slope of the ramp current. Use the first AP time and the to compute the associated current as baseline+time\*slope. Slope is also written to output file.