Examining the Effect of Action Potential Back Propagation on Spike Shape

Janelia Farm Research Campus

Chandan Singh
August 23, 2014

Introduction

Spike sorting is a process used in neuroscience that aims to identify the waveforms of firing neurons in an extracellular voltage recording. When a neuron fires an action potential, it creates a voltage time series with a particular spike shape. When an electrode is placed at a point in extracellular space to measure this voltage, the spike shapes of all the neurons near the electrode determine the voltage it records. Distant neurons appear as small spikes in the recording while nearby neurons appear as large spikes. Spike sorting aims to filter out background noise in the recording and assign each spike to a specific neuron.

Spike sorting algorithms assume that the spike shapes of individual neurons are the same every time they fire. However, it is possible that this isn't the case. Every time a neuron fires, there are large current fluxes into the soma, and these fluxes largely determine the shape of the neuron's spike. These current fluxes in the soma show little variability, but as early as 1996, action potentials were found to back propagate into the dendrites of neurons. The action potentials back propagating into the dendrites of a neuron would also affect its spike shape. Furthermore, it was hypothesized in 1996 that the back propagation of these action potentials was dynamically controlled by inhibitory synapses (Spruston et al). If the back propagation of action potentials were under dynamic control, currents would travel down different branches of the dendrite each time it fired. The spike shapes of individual neurons would change as a result, and this would have an effect on spike sorting algorithms.

This project aimed to quantify how significant an effect the back propagation of action potentials down dendrites has on the spike shape of a neuron. The effect was examined through computer simulations of a neuron using the software packages NEURON and Neurocube. If the effect is large enough, it will have an impact on spike sorting algorithms.

Methods

NEURON

NEURON is a software package that models the intracellular interactions of a neuron. Though NEURON was only used through Neurocube in the final analysis of this project, it is important to understand the details of this model. In addition, further work in this direction will likely utilize NEURON to change specific properties in the model of the neuron or to change parameters such as the time step of the neuronal recording.

NEURON is an implementation of an intracellular compartmental model developed by Hines & Carnevale in 1997. It divides the neuron into small segments, which it then models as cylinders. These cylinders have an axial resistance, a leakage resistance, currents, and capacitances. These electrical properties are used in the compartmental model to properly simulate the currents and voltages generated by the firing of the neuron.

The first step of the project was to generate a model of a neuron whose dendrites could be dynamically turned on or off to stop the back propagation of action potentials. In order to do this, experimentally obtained neuron morphology data was retrieved from Neurocube, and then fed into a NEURON model. This data created a morphologically correct model of a cortical pyramidal cell from cat tissue.

Once the model was created, the model was stimulated at the soma and the resulting action potentials were measured. Once the model was up and running, the back propagation of action potentials down the dendrites was controlled. This was done in two ways: using an inhibitory synapse and blocking ion channels in the dendritic membranes. The more successful method followed Spruston's thesis, which stated that inhibitory synapses could control the back propagation of action potentials down dendrites. When an inhibitory synapse was put on one branch of a dendrite, the action potential was effectively stopped in that branch while it was free to travel down the other.

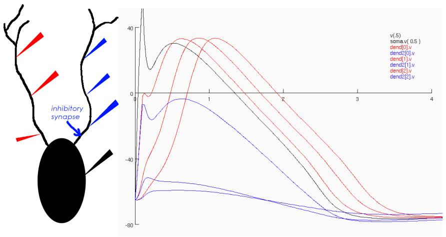

Figure 1. Action potential traveling down two branches in a dendrite: the blue curves represent successive segments of the branch that was affected by an inhibitory synapse.

Although these specific models weren't put into Neurocube, they demonstrate the back propagation of neuron signals down the dendrites and they show the effect an inhibitory synapse can have. Further work can use NEURON to test the results of more detailed dendritic back propagation. The time step for the Neuron output can be decreased (the time step used was about .15 milliseconds), and the method of stopping a back propagating action potential can be controlled.

Neurocube

Neurocube is a software package written in Matlab that models extracellular electrode recordings based on work by Camunas-Mesa & Quiroga. It was used to generate recordings of the voltage generated at different points around the firing neuron. Neurocube first uses NEURON to simulate the intracellular firing of a neuron. Then it uses the output of NEURON to calculate the potential each neural segment creates utilizing mathematical formulas such as Coulomb's law. It then properly sums the potentials generated by each of the neural segments to find the electrical potential at extracellular points in space as a function of time.

Neurocube does not actually run Neuron to simulate the spiking of a single neuron, but rather it accepts the output of NEURON. Neurocube ships with it the output of NEURON for 4 preset Neuron morphologies and five sets of parameters for each of those four morphologies. It gives these as the currents through each segment of the neuron over a period of time. In order to simulate turning off a dendrite, the current in the sections corresponding to those dendrites must simply be set to 0. However, this method is rather blunt, as inhibited action potentials are still likely to produce some small currents.

Neurocube was able to recapitulate Spruston's data. As in the NEURON simulations, the neuron was stimulated at the soma and its effects were measured. When the voltage was taken at the soma and at a point along one of the dendrites, it was shown that the voltage peaked at the dendrite after the soma, thus validating that the action potential was propagating from the soma back down the dendrites (Figure 2).

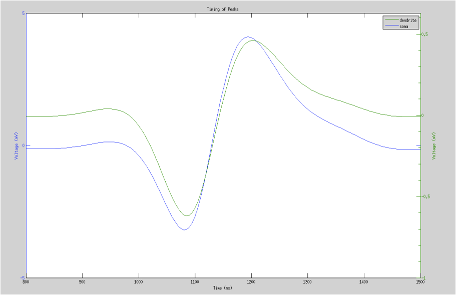

Figure 2. Voltage waveform at the soma (in blue), and at a point along the dendrite (in green). As expected, the dendrite peak is after the soma peak and has a smaller amplitude.

Simulation

Neurocube was used to run a simulation of a neuron firing normally, and of a neuron firing with the currents going through one dendrite completely removed. The dendrite had several branches and was reasonably large, making up about 10.3% of the neuron's volume. Its center of mass was rather far from the dendrite (about 232.2 microns.) The simulation was run with a neuron in the center of an 800^3 micron^3 volume with 101^3 electrodes evenly spaced around it. The neuron was a cortical pyramidal cell from a cat's somatosensory cortex, and it was a standard data set that was shipped with Neurocube. After the simulation was run, the raw data consisted of the voltage at each of the electrodes over 4 milliseconds. There were two sets of data: the original data, which showed the voltages of the original neuron, and the altered data, which showed the voltages of the neuron with one of its dendrites removed.

Data Analysis

Raw Data

The first step to analyzing the data was visualizing the raw data. To do this a two-dimensional cross section of data was taken through the center of the neuron, and the RMS of the curve recorded by each electrode was displayed as the color of a square centered at the position of the electrode (Figure 3).

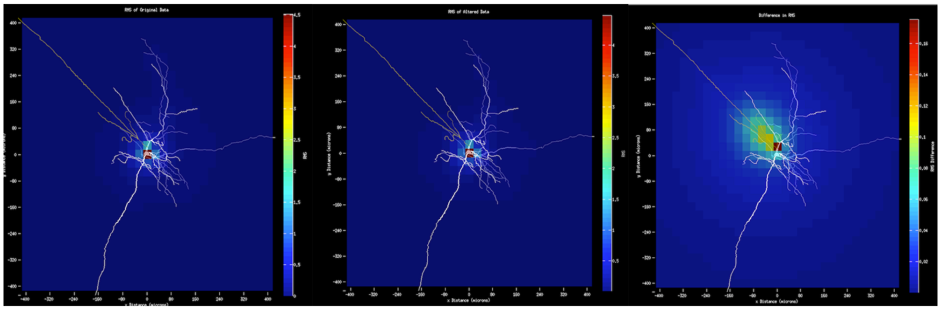

Figure 3. RMS of the original data, the altered data, and the subtracted data respectively. As expected, the original and altered data both look very similar because there is a much larger current in the soma then in the removed dendrite. However, if the curves recorded by the original data are subtracted by the curves recorded by the altered data and then the RMS is taken, there is a substantial difference in the RMS at points near the dendrite.

This RMS of the difference of the curves was examined in three dimensions, and was found to be highest at the center of mass of the dendrite. However, the RMS does not decrease steadily as the electrode is moved farther away from the center of mass of the dendrite. One reason for this is that even in the removed dendrite, there is a higher current density near the soma. The currents going through the farthest branches of the dendrite are considerably smaller than the currents going through the dendrite near the soma.

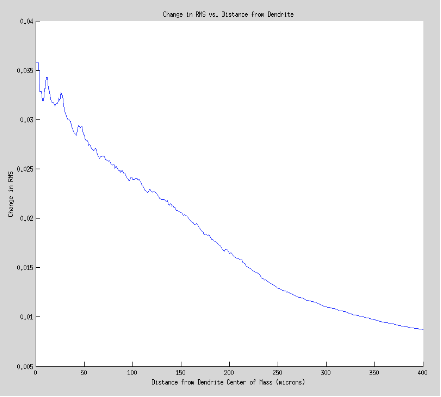

Figure 4. The RMS difference as a function of the distance from the center of mass of the dendrite; the RMS is highest at the center of the mass and decreases as the electrode is moved farther away.

Knowing that the greatest RMS changes are along the dendrite and at its center of mass, samples can be taken at regular points along the dendrite in a line through its center of mass. If these samples are taken, the original and altered curves seem to differ more father from the soma (Figure 5). However, because the signal decreases far away from the soma, the absolute difference in the curves becomes very small because the amplitude of the curves is so small.

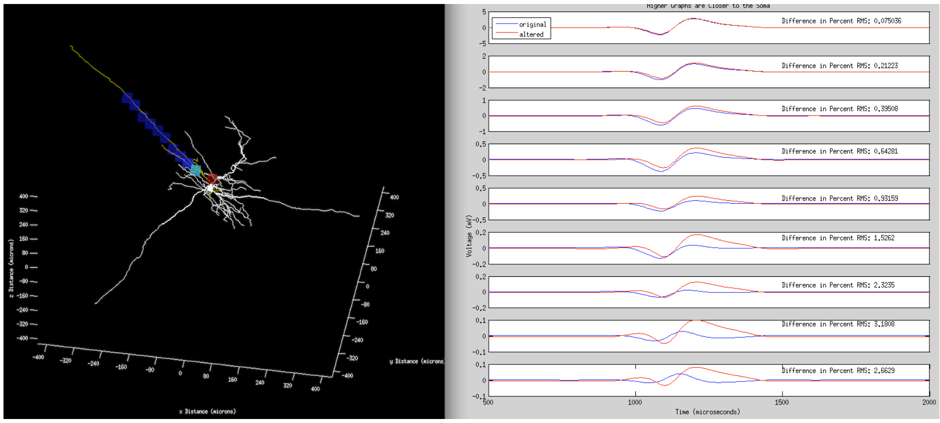

Figure 5. The graphs on the right show the original waveform (in blue), and the altered waveform (in red) along the points shown on the neuron. The graphs at the top are closer to the soma. Note that the waveform amplitudes decrease considerably as the electrode is moved farther from the soma.

Normalized Data

After looking at these differences, it was important to note not just the absolute difference in RMS between the original and altered curves, but the differences as a percentage of the original RMS. The data at each electrode was normalized by dividing each RMS difference by the RMS of the original waveform at that electrode. Points close to the soma showed a very small percentage change in their RMS (Figure 6). This is reasonable because the large currents in the soma mask the effect of the dendrite.

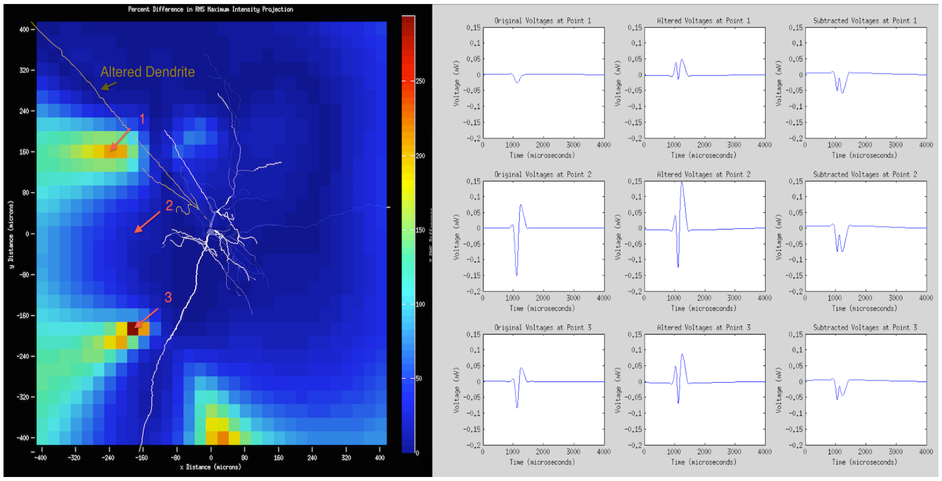

Figure 6. This figure shows a cross section of the data with the percentage change in RMS shown by the colors. On the right are the original, altered, and subtracted curves respectively for three sampled points. Even though point two had a large absolute change in RMS, its percentage change was small because its original waveform had a large RMS.

The normalized results show some large changes at points far away from the neuron; points 400 microns away from the soma show changes as large as 300 percent. However, this is because the original RMS of curves that are far way is very small, and thus even small changes from the nearby dendrite can have drastic effects on the normalized RMS change. However, at distances far away from the dendrite, these large percentage changes might not be significant because the signals are so small anyway. They would likely be indistinguishable from noise by an electrode.

The normalized waveforms were analyzed within 200 microns of the soma, because electrodes should be within this distance in order to properly pick up a neuron's signal. The following graph shows the percentage change in the RMS of the waveforms at different distances from the soma. There are large effects at far distances from the soma. Though large differences are expected near the dendrite, these same effects are seen all around the soma; the data was measured as the average on the surface of concentric spheres starting at the soma. One reason for this is that a large majority of the current going through the removed dendrite was near the soma of the neuron. Thus, it is not surprising that there are effects radiating in all directions out from the soma as opposed to only following the path of the dendrite.

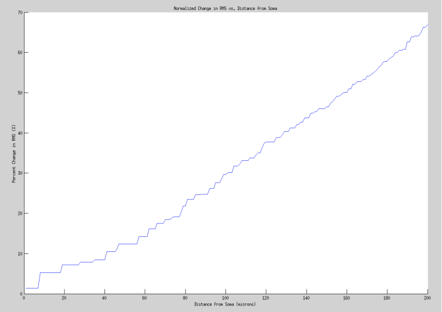

Figure 7. The percentage change in RMS as a function of the distance from the soma. There is approximately a 30% change in the RMS at 100 microns, and a 68% change at 200 microns.

Comparison to Noise

Noise was measured and analyzed in the voltage recordings to see its effect on spike shape. This project considered two main sources of noise present in neuronal voltage recordings: one source of noise was the neuronal noise generated by the firing of nearby neurons, and the other source was the shot noise generated by the random motion of electrons. Both of these noise sources were compared to the noise generated by dendritic back propagation. Typically, a neuron's spike is significantly larger than the noise picked up in an extracellular voltage recording, but the effects of noise can be important to spike-sorting algorithms.

To see the effect of noise on neurons, the voltage at the soma of a single neuron was measured. Neurons were placed all around this neuron with a density of 300,000 neurons / mm^3. Then, the neurons within a certain distance from the center neuron's soma were removed, and their effects were measured. As expected, the neuron's spike was more easily discernible when the other neurons were farther away from it.

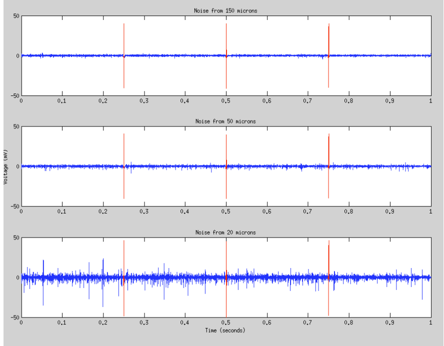

Figure 8. Voltage recordings at the soma of a neuron with both neuronal and shot noise. The graphs show the noise generated by randomly distributed neurons 150, 50, and 20 microns from the soma of the center neuron. The red areas show the center neuron firing.

The disturbance was then broken down into its shot noise and neuronal noise components. It was found that the shot noise component was so small that it was negligible. The voltage RMS of the shot noise was calculated using the formula $(4kTR)^{1/2}$ $B^{1/2}$. Even with a resistance of 10 mega-ohms and a bandwidth of 10 kilohertz, this RMS was several orders of magnitude below the voltage RMS produced by the neuronal noise.

The next step was to examine the neuronal noise. Randomly dispersed neurons were generated, and then neurons within a certain distance of the center neuron were cleared away. The RMS of each of the noise curves was taken and plotted. As expected, the RMS decreases as the distance to the noise-generating neurons is increased. The RMS value of the neuronal noise and the RMS of the dendrite voltage recording are both about .38 microvolts when the noise-generating neurons are roughly eighty microns away. Thus, the dendrite being on or off has the same effect on the RMS of a recording as neurons firing eighty microns from the soma of a dendrite.

The effects are a small percentage of the original RMS at the soma, but farther away, this percent change becomes much larger. Figure 7 shows how the effects of the dendrite back propagation get larger as the electrode is out from the soma. Similarly, the effect of neuronal noise increases.

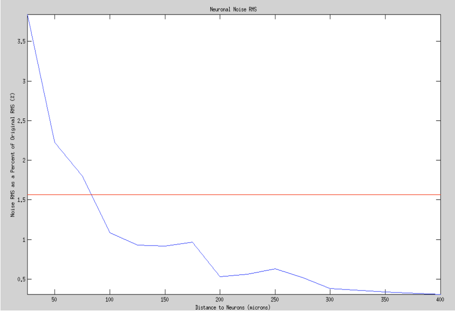

Figure 9. RMS values from neuronal noise at different distances. The values are displayed as a percent of the RMS of the original firing neuron at the soma. The red line shows the RMS of the dendrite of the center neuron firing by itself, without the rest of the neuron.

Though the RMS values show that the dendrite's effect is the same as noise-generating neurons 80 microns away, the noticed effect is smaller. This is because the RMS of the neuronal noise includes times when there are no neurons firing, or there are positive and negative voltages from hyperpolarized and depolarized neurons canceling each other out. The RMS of the firing dendrite, on the other hand, contains only nonzero voltages from when the dendrite is firing. The amplitude of the waveform produced by the firing dendrite is similar to the maximum amplitudes produced by noise-generating neurons 300 microns away.

Conclusion

In the final analysis, the simulation provides a rough guideline to the percentage change in RMS that can be expected at a certain distance from the soma due to dendritic back propagation (Figure 7). In addition, it shows how the effects of dendrite back propagation compare to noise in biological recordings (Figure 9). However, there are multiple factors to consider when weighing the significance of these percentage changes, and how they apply to biological situations.

First, the method of removing the dendrite in the Neurocube simulation was very blunt. Simply removing all the currents in a dendrite when the soma fires an action potential is an extreme case that is unlikely in a real scenario. In a biological situation it is likely that some currents flow through the inhibited dendrite, at least passively. A NEURON simulation could be run to see the effects of allowing some, but not all current to flow through the dendrite; this would essentially be partially removing the dendrite. However, the data presented here represents an upper bound for what could happen in a biological situation. It is likely that the currents through a dendrite will be only partially removed, but the data shown here represents the worst possible case that the currents through the dendrite were entirely removed.

Another factor to consider is the specific neuron used in the simulation. The relative mass and current density in different dendrites and neurons will yield different changes in the RMS. The dendrite removed in this simulation was reasonable large, making up about 10% of the neuron's volume, and its center of mass was rather far from the dendrite (about 232 microns.) When these parameters change, the percentage change in RMS will also.

In order to shift from a computer model to a biological situation, the constraints on the model must be considered. The properties of each segment of a neuron are only as accurate as the biological data the Neurocube creators used. In order to simulate the effects of different properties in different segments of the neuron, new NEURON simulations must be run and fed to Neurocube. This simulation only used the output of NEURON with the default parameters that the creators of Neurocube gave it. In addition, the NEURON simulation output its currents with a time step of about .15 milliseconds, which could be decreased to improve the simulation's accuracy.

It is important to make this data useful for spike sorting. Only a substantial change in spike shape will make a difference in spike sorting algorithms, as they contain a large amount of noise from distant neurons. It is important also not only to consider the average percent RMS changes shown in Figure 8 but also to look at some of the drastic changes that occur at individual points. Some of the points shown in Figure 6 show that there are specific locations where drastic RMS percentage changes occur. If an electrode recording is taken from these spots, there will be a larger impact on spike sorting.

Finally, noise must be considered in any spike-sorting algorithm. The project found that shot noise was negligible when compared to noise generated by nearby neurons. It further found that roughly eighty microns was the distance at which dendritic action potentials had the same effect as neuronal noise. The distance between neurons differs based on what biological sample is used, and thus this project's findings must be adapted. With very densely packed neurons, it is likely that dendrite back propagation might not need to be accounted for at all. However, if there is space between neurons, dendritic back propagation might be very important to spike sorting. All types of noise present in biological recordings should be further analyzed in order to fully test the effects of dendritic back propagation on spike sorting.

Technical Details

Specifying Run Parameters

Many changes were made in order to run a simulation with multiple electrodes, and in order to specify the number, locations, and types of neurons in the simulation. These changes were made in the Matlab files neurocube.m and run_simulation_par.m, and are documented in the code.

Neurocube.m is the file to run to start the simulation. It sets up the parameters for the simulation and creates a small GUI to start the simulation. All the parameters are located at the top of the file. The GUI was not used, so once the file is run, the simulation will start by itself. Once the simulation is run, in order to save the data the user must click the 'Save Simulation' button.

Run_simulation_par.m is in charge of doing the actual calculations. It calls methods to read in the output of the NEURON simulations. It also has some parameters that can be changed at the top of the file to specify which models to use.

Data format

The data was saved in text files in the data and plot-making files are located in the mat/DATA folder. The raw data is stored in text files with the naming convention: 'outer_bound_of_cube'+'by'+'num_electrodes_per_dimension'+'type_of_data.txt'. The types of data are original, altered, and zeros (which contains data when the simulation was run with only the dendrite turned on.) If the file name ends in 'RMS', it contains only the RMS of the waveform at each electrode (and is much faster to read.) In order to save a simulation, the user should save click Save Simulation after the simulation has run, and then specify a filename in convert.m to make the data into a text file. Each text file contains a row for each electrode. Each row contains 800 comma-separated numbers each of which is the voltage at a time step for the electrode. Unless the simulation is changed, the voltages are given in millivolts and the time step is 5 microseconds. The top row in the data file corresponds to the first electrode in the array Neurons.Electrodes (located in the neurocube.m file). Keep track of the order of the electrodes, as this is the only way to determine what position in space the data corresponds to.

Visualizing the Data in Matlab

The data was visualized in Matlab. The scripts used to do this visualization are located in the mat/DATA folder. Code to render a maximum intensity projection is in makeVoxelPlots3d.m and code to make any of the 2D plots in this paper is in makeSinglePlots.m.

Changing Neurons in Neurocube

Neurocube reads in the output of the NEURON simulation from the output folder. This folder contains 20 folders: one for each of the 5 geometries and 4 sets of parameters that Neurocube comes packaged with. Within each of these folders are several files. One will be the neuron_name_geom.dat. This file contains information on the morphology of the neuron. Then there are several files that are called neuron_name_t00.004.dat and where the number after the t changes up to about 73. Each of these files contains the output of NEURON at a given time step. The output is given as the current flowing into our out of each segment of each neurite in the neuron at the given time step. The line numbers correspond to the line numbers in the geom.dat file, so you can tell which current goes with which segment. In order to effectively remove a dendrite without rerunning a NEURON simulation, all of the currents in the segments of that dendrite must be set to 0. This can be effectively be done by finding out which lines in the files contain that dendrite (by looking at the geom.dat file), and then using a simple script to set these lines in the file to 0 (a list of the times steps is available in neuron_name_times.dat.) An example python script for turning off a dendrite in the 51-2a_0001 neuron: is located in mat/DATA/nrn/zeros.py

Running Simulations with Noise

The code for running Neurocube simulations with noise is located in the folder 'Neurocube with Noise'. To use the default thermal noise that the Neurocube writers programmed in, set Par_sim.use_noise to true (this variable is used in run_simulation.m). To set the radius of neurons to delete, set

Par_sim.dsingle. To run the simulation, run neurocube.m, then click Manual and click ‘Update Cube’. To put a neuron in the center, at the electrode, click one of the checkboxes and set Normalized distance to 0. The firing rate is set programmatically at the bottom of neurocube.m. Code for graphing is contained in graphs.m and getCoordinates.m.

References

Camuñas-Mesa, L. A., & Quiroga, R. Q. (2013). A detailed and fast model of extracellular recordings. *Neural computation*, *25*(5), 1191-1212.

Hines, M. L., & Carnevale, N. T. (1997). The NEURON simulation environment. *Neural computation*, *9*(6), 1179-1209.

Spruston, N., Schiller, Y., Stuart, G., & Sakmann, B. (1995). Activity-dependent action potential invasion and calcium influx into hippocampal CA1 dendrites. *Science*, *268*(5208), 297-300.