Explanation of the Matlab functions in the *stocHHastic* package

The attached Matlab code implements the stochastic Hodgkin-Huxley model with ion-channel gating modeled as Markov chains. We provide both the full Markov chain model as well as its stochastic-shielding approximation (folder **HH**). In addition, we provide the code comparing the theoretical covariance matrices for the transitions and occupation numbers in the full and simplified models with the covariance matrices obtained from the simulations (folder **QandC**).

1. Folder HH

Programs with "full_mark" in the title utilize a full markov process for ion channel stochasticity; "simp_mark" in the title have the shielding approximation.

Files with "_Vclamp" at the end are entirely in voltage clamp and must have the voltage given to the program, while files with the "_Iclamp" suffix are in current clamp. Please see the comments in the files for details of parameter values.

The eight common inputs parameters to both current clamp and voltage clamp functions are:

dt: timestep in ms,

T: total time interval (duration of simulated signal) in ms,

Nanoise and **Knoise**: binary variables representing whether or not sodium and potassium, respectively, are stochastic or deterministic,

Nanum and **Knum**: channel densities for sodium and potassium per μm^2,

trans: duration of the transient (initial deterministic period that IS still included in the output),

area: membrane patch size in μm^2.

Scripts with "_Vclamp" at the end also have a ninth input,

v: the voltage at which they are clamped in mV.

The current clamp scripts return a value for the membrane potential that the voltage clamp scripts do not, but after that both output the same nine values; total current, sodium current, potassium current, a timetrack vector that is in seconds, a sodium matrix showing the number of channels in each Markov state, a potassium matrix showing the number of channels in each Markov state, the total number of sodium channels, the total number of potassium channels, and the time it took the simulation to run.

Figure 2 is in voltage clamp, and was created with the respective files with the suffix "_Vclamp" using the parameters given in the figure caption. Figure 3 was created with the Matlab files with suffix "_Iclamp" using parameters in the figure caption.

2. Expanded Model

These files are very similar in notation and style to the HH models, with “full_mark” and “simp_mark” representing programs without and with the shielding approximation, respectively.

All of the input and output parameters described for the HH package are the same, however there are additional inputs

Pnum and **SKnum**: channel densities for P/Q channels and SK channels, respectively

Idc: Input current (should be a vector of equal length as there are timesteps)

g_input: Input conductance (with reversal potential of 0mV)

Buffering_constant: The buffering constant represents a constant that the amount of calcium entering the cell is divided by, to account for the endogenous buffering of the cell. Since the calcium concentration will only be used to gate the SK channels, this is a valid approximation.

Catau: The time constant at which calcium is removed from the cell, modeled as an exponential decay.

And additional outputs

I_pca and **I_sk**: P/Q calcium and SK currents

Caconc: the concentration of calcium in the cell

PCa and **SK**: the number of channels in each Markov State throughout the trajectory

g_current: the current due to the g_input conductance

Spike-triggered averages are computed as the normalized cross-correlogram of the input current with the spike events (represented as delta functions).

To recreate responses to current or conductance input, simply run the files “OUnoise” for Ornstein-Uhlenbeck noise or “poisson_input” for Poisson conductance events (note that OUnoise needs to be called as a function but poisson_input is simply run as a terminal script), and use the outputs as the Idc or g_input input for the model script. The other variable should be simply set to a vector of zeros.

Note that the Hodgkin-Huxley Files in the other package do not have an input current value, so to recreate those results, simply set the values for Pnum and SKnum to 0 in the expanded model

3. Folder QandC

The m-file names are self-explanatory. Files with prefix “simulations” compute the covariance matrices Q and C from the simulated data of stochastic ion channel gating in voltage clamp, whereas files with prefix “theoretical” compute those matrices from the theoretical expressions shown in the manuscript. File names containing “full_MC” use the full Markov chain model and files names containing “simp_MC” use the stochastic-shielding approximation. Although three of the four m-files are defined as functions, they do not take input parameters, as they are defined in the within the file. These files where defined as functions so that they can call functions defined within the files.

The program “theory_vs_sims_CovN_MC.m” compares the results from theory and simulations using the output files (mat-files) of the other three m-files. The mat-files used for the manuscript are provided as an example, along with .eps figures showing the results.

By **Nicolaus T. Schmandt** (n.schmandt@gmail.com) & **Roberto Fernández Galán** (rfgalan@case.edu),

Case Western Reserve University, July 2012.