

Recent Developments in NEURON

Michael L. Hines^{a*} and Nicholas T. Carnevale^b

^a Department of Computer Science, Yale University,
PO-Box 208285, New Haven, CT, USA 06520-8285

^b Department of Psychology, Yale University,
PO-208205, New Haven, CT, USA 06520-8205

*Corresponding Author: Tel: +203-737-4232, email: michael.hines@yale.edu

This is a revised draft of: Hines ML and Carnevale NT (2005). Recent Developments in NEURON. Brains, Minds and Media, Vol.1, bmm221 (urn:nbn:de:0009-3-2210).

Abstract

We describe four recent additions to NEURON's suite of graphical tools that make it easier for users to create and manage models: an enhancement to the Channel Builder that facilitates the specification and efficient simulation of stochastic channel models; an enhancement to the Cell Builder that enables the convenient specification of spatially nonuniform properties in anatomically complex cells; the Model Viewer, which presents a browsable and quickly understood summary of the properties of models of individual cells and networks; and the Import3D tool, which simplifies conversion of detailed morphometric data into computational models of neurons.

Keywords

NEURON simulator, ModelDB, stochastic channels, channel distributions, model discovery, morphometric data

1 What's new with NEURON?

NEURON [Hines 2005] has undergone many revisions and enhancements since its inception. Most of these involve internal details that, while beneficial for accuracy or performance in one way or other, are not always apparent to users. However, over the past decade a particular need has emerged to make the task of specifying and managing computational models more convenient. This need is a consequence of the accelerated adoption of computational modeling by experimentalists, which, combined with advances in quantitative characterization of the properties of neurons and neural circuits, has stimulated the development of models that are growing rapidly in number and complexity.

To meet this challenge, we have added many new classes and procedures to NEURON that increase the power and flexibility of this simulation environment, and created graphical interfaces for these new features so that users can easily apply them. Examples of these include

Model specification tools

- *Point Process Group Manager* for dealing with point sources of current, such as current and voltage clamps
- *Kinetic Scheme Builder* for specifying reaction schemes
- *Channel Builder* for specifying the properties of voltage- and ligand-gated ion channels
- *Cell Builder* for managing anatomical and biophysical properties of model cells
- *Network Builder* for prototyping small networks of biophysical and artificial neuron models as an initial step to developing large-scale models
- *Linear Circuit Builder* for creating models that involve gap junctions, ephaptic interactions, dual-electrode voltage clamps and dynamic clamps, and other arbitrary combinations of model neurons and electrical circuit elements

Simulation control tools

- *Variable Step Control* for automatic adjustment of the state variable error tolerances that govern adaptive integration
- *Multiple Run Fitter* for optimization of function and model parameters

Analysis tools

- *Impedance Tools* for electrotonic analysis, including computation of input and transfer impedance, voltage transfer ratios, and the electrotonic transformation
- *Model View* for automatic discovery and systematic display of model structure and parameters
- *Import3D* for importing Neurolucida and other morphometric data formats to NEURON

Documentation and tutorials for many of these tools are available at NEURON's WWW site <http://www.neuron.yale.edu/>. In this paper we present four of the most recent enhancements to NEURON: stochastic channels in the Channel Builder, parameter inhomogeneity over subsets in the Cell Builder, the Model Viewer, and the Import3D tool. Further details about NEURON are provided in *The NEURON Book* [Carnevale *in press*].

2 The Channel Builder

The Channel Builder is a graphical interface for creating voltage- and ligand-gated channels whose state transitions are described by kinetic schemes and/or HH-style differential equations (including the Borg-Graham equations [Borg-Graham 1991]). The look (figure 1) and feel of this native hoc implementation are very similar to the Java-based Channel Builder from Robert Cannon's Catacomb [Cannon 2005]. A configured Channel Builder can be saved to a file that contains a plain text specification of the mechanism which is human-readable. Channels constructed with this tool execute slightly faster than equivalent mechanisms created with NMODL, the programming language used to add new mechanisms to NEURON.

When it was first added to NEURON, the Channel Builder dealt only with models in which the gating states are continuous functions of time, i.e. a continuous system approximation to a large population of channels with discrete states (figure 2 left). The latest version of this tool includes many improvements, the most notable of which may be the efficient simulation of stochastic single channel activity. In this mode, the gating states and simulated conductance make abrupt transitions between discrete levels (figure 2 right, figure 3), as would be produced by the opening and closing of individual channels in a population of countably many channels.

3 The Cell Builder

It has long been known that biophysical parameters such as channel densities are not constant throughout a cell but instead vary with location, and models that incorporate such inhomogeneities have been appearing with increasing frequency in recent years. To support the construction of models with inhomogeneous parameters, a new feature has been added to the Cell Builder: the ability to specify that parameters can vary with location as functions of an independent variable. This graphically supports the idiom

```
forsec subset for (x, 0) { rangevar_suffix(x) = f(p(x)) }
```

where `rangevar_suffix(x)` is the parameter of interest, $p(x)$ is a domain function over the subset and f is any expression. Built-in domain functions are arc (path) distance from the soma, radial distance from a point, and distance along an axis in the xy plane.

Figure 4 illustrates one of the steps involved in specifying an inhomogeneous distribution of sodium conductance over the apical dendritic field of a hippocampal CA1 pyramidal neuron. The goal is for conductance density to decrease linearly with distance along the long axis of the cell (i.e. approximately perpendicular to the pyramidal cell layer). This figure shows the Subsets page of the Cell Builder, which is where one sets up the domains over which parameters will vary, and specifies the domain functions that will be applied. The expression f that governs how a parameter varies with $p(x)$ is specified on the Biophysics page of the Cell Builder (figure 5).

4 The Model Viewer

As computational models grow increasingly complex, there has been a corresponding increase of the difficulty of finding out exactly what properties have been embodied in a

model. This is hard enough when dealing with one's own models, but it can be almost excruciating if the model was developed by someone else.

To alleviate this problem, NEURON now has a tool called the Model Viewer which offers a concise yet complete textual and graphical summary of model properties. The summary is initially presented as a very brief, top-level outline with nodes that can be selectively expanded by a few mouse clicks to reveal as much or as little detail as one desires. The Model Viewer (figure 6) provides a means for quickly examining the anatomical and biophysical properties that are present, and their distribution in space. The Model Viewer works equally well with models of individual cells and networks of cells, in any combination of biophysical and/or artificial spiking neurons. It is a very convenient tool for ensuring that one's own models have been properly configured, and also for discovering the properties of models obtained from ModelDB or other sources.

5 The Import3D tool

Creating models based on detailed morphometric data is a recurring problem in computational neuroscience. The difficulty of this task is compounded by the plethora of file formats that have been used, and the fact that such data often contain errors (e.g. orphan branches or trees) that may seriously damage the translation. Over the years, several different standalone programs have been developed to translate morphometric data into model specification code, the most recent of which is cvapp [Cannon 2002]. However, the utility of these programs has been limited by factors such as cross-platform incompatibilities, requirements that users compile source code or install and configure third-party software, and lack of support that renders them obsolete in the face of data format and operating system changes.

To help modelers work with detailed morphometric data, NEURON now has a GUI tool for that converts morphometric data files into models. The Import3D tool (figure 7) can read Eutectic, SWC, and NeuroLucida classic and Version 3 files, and can export the data directly into the CellBuilder (figure 8) or generate a "top level" instance of the model. Its controls and graphical interface facilitate quick identification of orphan trees and other errors. The Import3D tool automatically identifies and repairs many common problems, and helps users identify other errors that require the exercise of judgment and manual editing of a copy of the original morphometric data.

6 Summary

The four enhancements to NEURON described in this paper offer significant benefits to users who work with models that are anatomically and/or biophysically complex. In the near future we expect to add another to NEURON that allows GUI specification of second messenger pathways, pumps, and simple forms of ionic accumulation.

These enhancements also exemplify a recent trend in model specification: the replacement of procedural descriptions with form-based descriptions. In addition to promoting the use of GUI tools for filling out the forms, the form-based approach offers a much greater possibility of sharing model descriptions among disparate simulator programs through a common exchange format such as NeuroML [Goddard 2001].

References

- [Borg-Graham 1991] L. Borg-Graham. Modelling the non-linear conductances of excitable membranes. In J. Chad and H. Wheal, editors, *Cellular Neurobiology: A Practical Approach*, chapter 13, pages 247-275. IRL/Oxford University Press, 1991.
- [Cannon 1998] R.C. Cannon, D.A. Turner, G.K. Pyapali, and H.V. Wheal, 1998. An on-line archive of reconstructed hippocampal neurons. *Journal of Neuroscience Methods*, 84:49-54.
- [Cannon 2002] R. Cannon. cvapp, 2002. <http://www.compneuro.org/>.
- [Cannon 2005] R. Cannon. Catacomb, 2004. <http://www.enorg.org/>.
- [Carnevale 2005] N.T. Carnevale and M.L. Hines, 2005. *The NEURON Book*, Cambridge University Press, in press.
- [Goddard 2001] N. Goddard, M. Hucka, F. Howell, H. Cornelis, K. Skankar, D. Beeman, 2001. Towards NeuroML: model description methods for collaborative modeling in neuroscience. *Phil. Trans. Royal Society, series B*, 356-1412:1209-1228.
- [Hines 2005] M.L. Hines. NEURON, 2005. <http://www.neuron.yale.edu/>.

Requirements

NEURON and its documentation are available at no charge from <http://www.neuron.yale.edu/> for MS Windows, OS X, and UNIX/Linux.

Figure 1: The open state is called m_{3h1} in this kinetic scheme formulation of the Hodgkin-Huxley sodium conductance model, which was implemented with NEURON's Channel Builder.

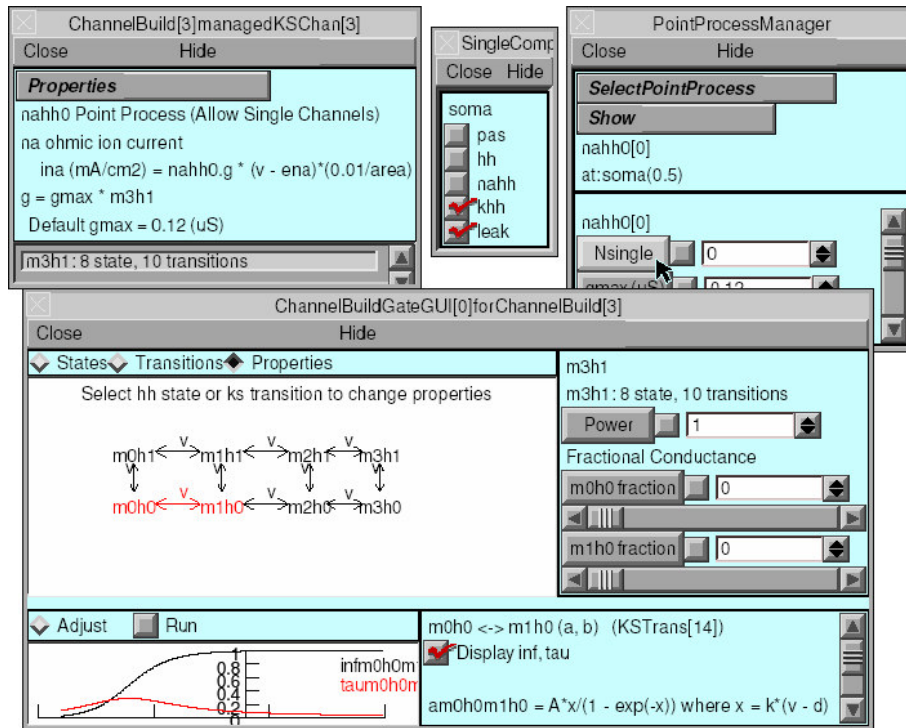


Figure 2: The Channel Builder's N_{single} parameter determines whether the channel model is simulated as a continuous or discrete system. Left: With N_{single} equal to 0, this sodium conductance model acts as a continuous system, i.e. g_{Na} varies continuously with time, and m_{3h1} is the "open fraction" of maximum g_{Na} . Right: When N_{single} is nonzero, m_{3h1} is the number of open channels. Here N_{single} is 1000, so the model acts like 1000 independent, voltage-gated channels, producing stochastic fluctuations in g_{Na} and i_{Na} and causing "channel noise" to appear in the membrane potential.

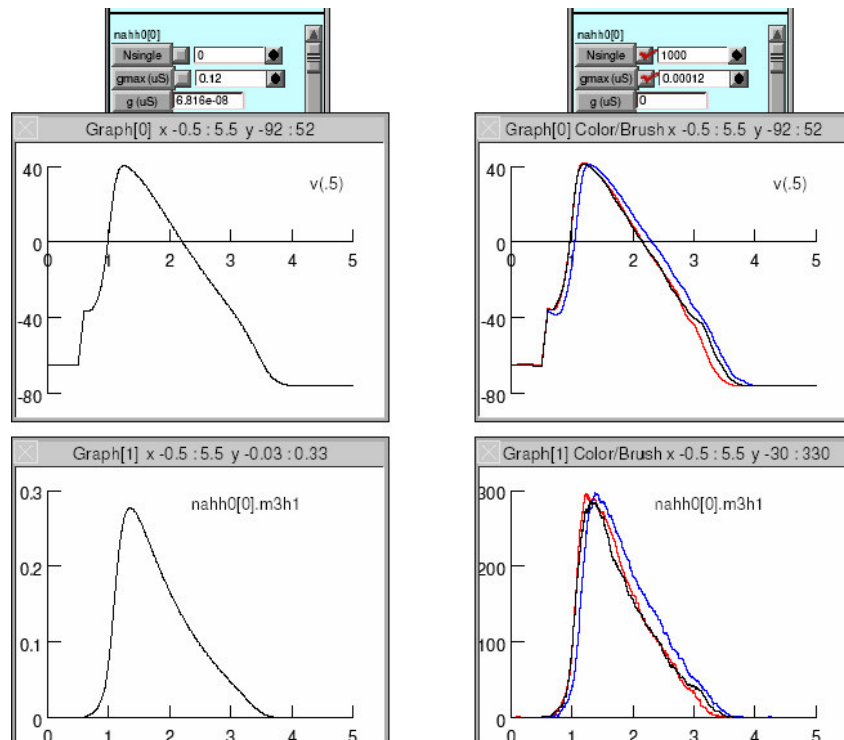


Figure 3: Close examination of the number of open channels reveals how fastidiously NEURON computes state transitions when adaptive integration is used.

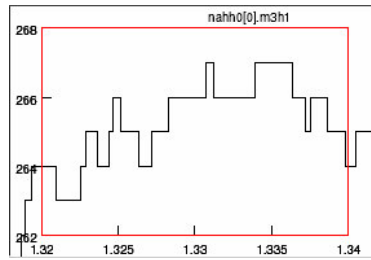


Figure 4: Variable parameter domains are specified on the Subsets page of the Cell Builder. Here the domain function $p(x)$ is the linear distance along the axis indicated by the blue line, and the origin for distance measurements is at the center of the blue square. By using the horizontal slider (in the right panel of the Cell Builder), one can preview the correspondence between p and the architecture of the cell. The red branches (sections) indicate that portion of the apical dendritic tree that is closer to the origin than the value of p shown on the graph (0.701413 in this figure, where 0 is at the origin and 1 is the most remote point in the tree); the remainder of the apical tree is shown in black.

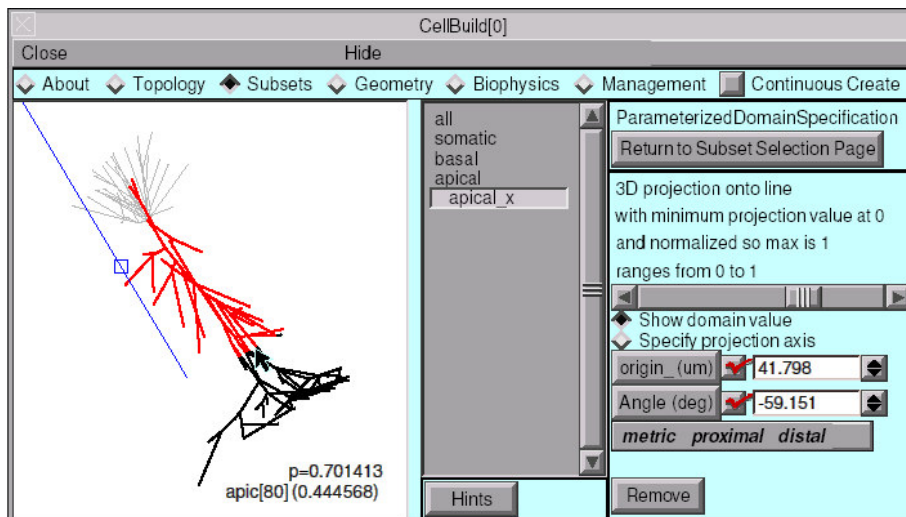


Figure 5: The relationship between an inhomogeneous parameter and its domain function is specified on the Biophysics page of the Cell Builder. In this example, sodium conductance density $g_{\text{nabar_hh}}$ decreases linearly with distance along the axis indicated by the blue line (note equation in the right lower panel of the Cell Builder, just below the button labeled **$f(p)$ show**).

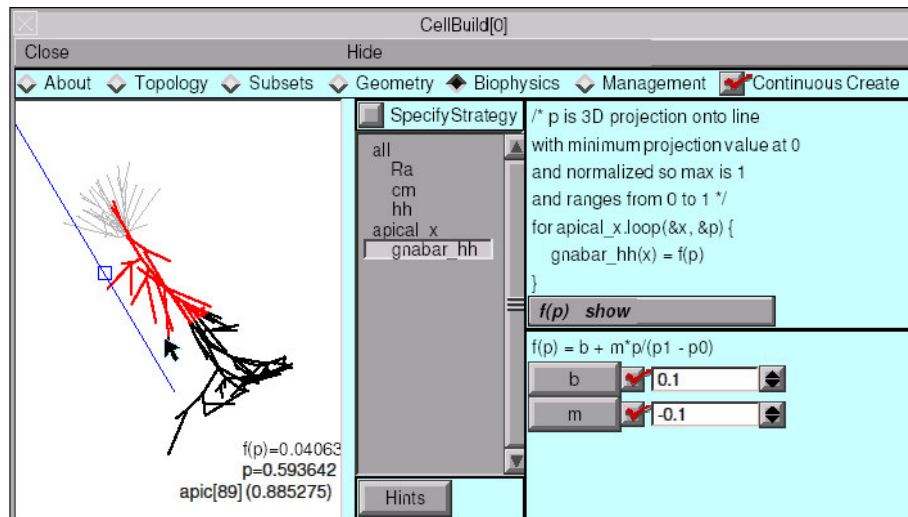


Figure 7: An Import3D tool after it has read one of the files in the Duke/Southampton Archive of Neuronal Morphology [Cannon 1998]. This tool automatically repairs many simple errors, and clicking on the Neurolucida filter facts button brings up a panel that describes the kinds of corrections that it applies. It also facilitates detection and localization of other errors that require user intervention (see figure 8).

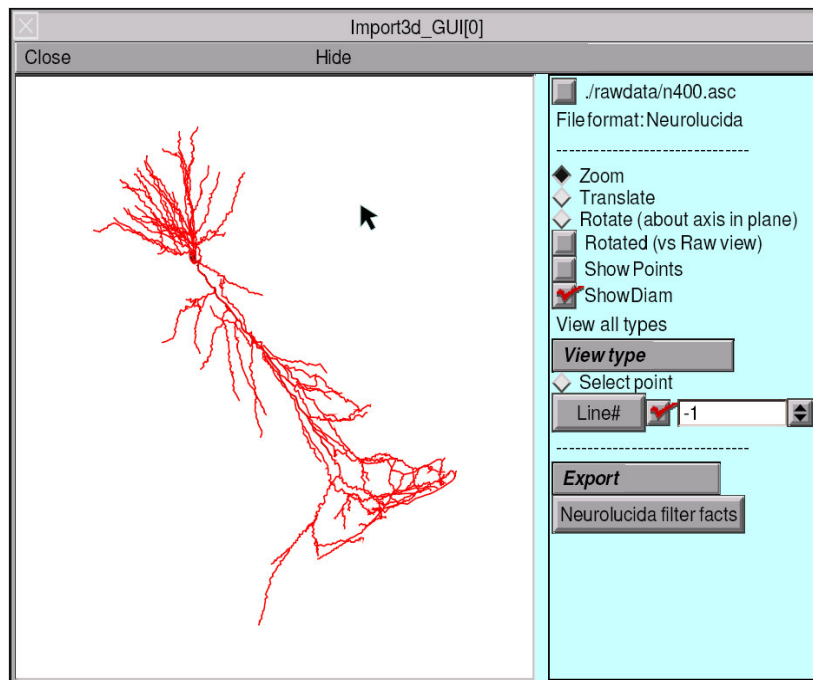
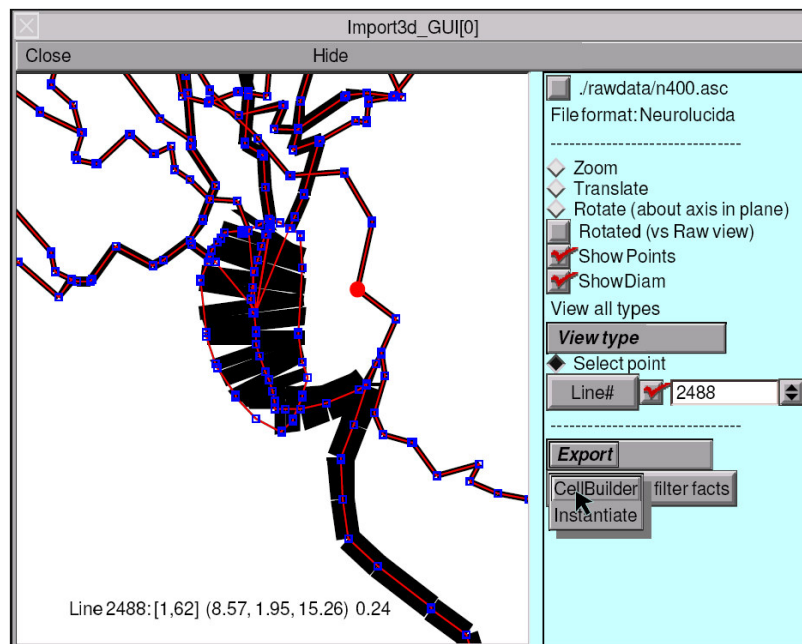


Figure 8: A high magnification view centered on the soma of the cell shown in figure 7, with Show Points active so that the location of each (x,y,z,diam) measurement is marked by a blue square. Clicking on one of these points places a large red dot over it, and the number of the corresponding line in the data file appears in the numeric field that belongs to the Line# button. This numeric field, and the spinner widget to the right of it, can also be used to navigate through the data file. Also notice that a copy of the line itself appears at the bottom of the graph's canvas. This helps users locate errors in a morphometric data file that require manual intervention. For example, the presence of both a soma outline (thin red trace around periphery of the soma) and a user-defined centroid with diameters duplicates the same information; one of them must be removed by editing the file before the data can be employed in a computational model.



Acknowledgements: The development of NEURON is supported by NINDS NS11613.