neuroConstruct: a tool for modeling networks of neurons in 3D space

Padraig Gleeson, Volker Steuber and R. Angus Silver

Department of Physiology, University College London, Gower Street, London WC1E 6BT, United Kingdom

Running title: Modeling neural networks in 3D

Correspondence address:
Prof. R. A. Silver
Department of Physiology
University College London
Gower Street
London
WC1E 6BT
a.silver@ucl.ac.uk
Tel: +44 (0)207 679 7830
Fax: +44 (0)207 916 8522

Acknowledgements: We thank Robert Cannon, Fred Howell and Arnd Roth for helpful discussions, the NEURON and GENESIS development teams, especially Michael Hines and Dave Beeman, and Sharon Crook for collaboration on the NeuroML specifications. We are grateful to beta testers of the application and David Attwell, Mark Farrant, Clare Howarth, Zoltan Nusser and Arnd Roth for comments on the manuscript. This work was funded by the MRC, Wellcome Trust and the EC (EUSynapse LSHM-CT-2005-019055). PG holds an MRC Special Research Training Fellowship and RAS is in receipt of a Wellcome Trust Senior Research Fellowship in Basic Biomedical Science.

Summary

Conductance-based neuronal network models can help us understand how synaptic and cellular mechanisms underlie brain function. However, these complex models are difficult to develop and are inaccessible to most neuroscientists. Moreover, even the most biologically realistic network models developed to date disregard many 3D anatomical properties of the brain. Here we describe a new software application, *neuroConstruct*, that facilitates the creation, visualization and analysis of networks of multicompartmental neurons in 3D space. A graphical user interface allows models to be generated and modified without programming. Model descriptions within *neuroConstruct* are simulator independent and based on new NeuroML standards, allowing automatic generation of code for NEURON or GENESIS, which carry out the numerical integration. The validity of models built in *neuroConstruct* was tested by reproducing published models and the simulator independence verified by comparing the same model on two simulators. Moreover, we show how more anatomically realistic network models can be created and their properties compared with experimental measurements by extending a published 1D cerebellar granule cell layer model to 3D.

Introduction

The characteristic 3D structures of brain regions like the cerebellum, hippocampus and cortex and the complex connectivity between and within them are thought to play a key role in determining how information is distributed and processed in the brain. Local circuits with fine scale specificity are also thought to be important for signal processing (Yoshimura and Callaway, 2005). For example, feed forward and feedback loops, which are often present within circuits, have distinct roles in processing (Pouille and Scanziani, 2004). Moreover, neuronal classes exhibit unique morphologies and modeling studies have shown that the shape of the dendritic tree affects the electrical behavior (Mainen and Sejnowski, 1996; Schaefer et al., 2003; van Ooyen et al., 2002; Vetter et al., 2001) and that the spatial pattern of synaptic contacts influences how signals are integrated (Destexhe and Pare, 1999; Jarsky et al., 2005; Markram et al., 1997; Mel, 1993; Rall et al., 1967). Neuronal signaling is not only restricted to point-to-point synaptic transmission but is also mediated by diffuse messengers including nitric oxide (Crepel and Jaillard, 1990; Jacoby et al., 2001), cannabinoids (Alger, 2002 Safo et al., 2006; Wilson and Nicoll, 2002) and neurotransmitters (Mitchell and Silver, 2000). The signal processing carried out by an individual neuron is therefore determined by both its morphology and the 3D structure of the network in which it is embedded.

Understanding how complex brain structures and the myriad of underlying mechanisms interact to produce higher level functions will require the help of network models with biologically relistic features. Several models have been developed that use compartmental neurons, Hodgkin-Huxley type membrane conductances and semi-realistic synaptic connectivity. These have been used to explore the potential mechanisms underlying synchronous activity (Davison et al., 2003; Maex and De Schutter, 1998), cortical oscillations (Traub et al., 2005), hippocampal memory (Kunec et al., 2005) and temporal coding (Buonomano, 2000). They have also provided insights into the potential causes of epileptiform activity in dentate gyrus (Santhakumar et al., 2005) and cortex (Bush et al., 1999). However, virtually all such models to date utilize highly simplified synaptic connectivity, featuring abstract neurons connected in either one (Maex and De Schutter, 1998; Santhakumar et al., 2005) or two dimensions (Medina and Mauk, 2000; Schweighofer and Ferriol, 2000).

Development of biologically realistic network models that include explicit 3D information would allow direct comparison of the model structure with detailed anatomical measurements. Indeed, it is unclear whether the complex synaptic connectivity patterns of regions such as cortex could be generated and verified in fewer than 3 dimensions. 3D network models would allow direct comparison of the spatio-temporal properties of simulated neural activity with experimental measurements using multi-electrode recordings (Nicolelis and Ribeiro, 2002) or 2-photon imaging of multi-unit activity in blocks of tissue (Helmchen and Denk, 2002; Ohki et al., 2005; Stosiek et al., 2003). They could also be extended to simulate volume signaling and brain metabolism. While the development of such 3D network models is theoretically possible with current simulators such as NEURON (Hines and Carnevale, 1997) and GENESIS (Bower and Beeman, 1997), and some preliminary attempts have been made (Berends et al., 2004; Howell et al., 2000), considerable technical difficulties remain. These include a requirement for algorithms that can create the highly nonuniform 3D synaptic connectivity observed in biological networks (Song et al., 2005; Sporns and Kotter, 2004; Yoshimura and Callaway, 2005), a method for verifying connectivity and routines for analyzing network behavior. Indeed, the absence of such tools has prevented the development and use of more biologically realistic 3D network models.

The development of a more integrated understanding of brain function will require closer interaction between experimental and theoretical neuroscientists (De Schutter et al., 2005; Destexhe and Marder, 2004; Segev and London, 2000). At present communication between these two groups, and even between individual theoreticians, is hampered by poor accessibility and inoperability of models. Although many single cell and network models are made available on public databases (Hines et al., 2004), their utility as research tools is often restricted to those familiar with the specialist scripting languages, which are simulator specific. For example, a neuronal model written in NEURON script cannot be used as part of a GENESIS simulation, unless it is converted manually, thereby limiting its interchange and reuse. Although Graphical User Interfaces (GUI) have significantly improved the accessibility of single cell modes, simulating network behavior is still rather inaccessible to all but a small group of computational neuroscientists.

We have developed a new software application, *neuroConstruct*, that facilitates the creation, and analysis of networks of multicompartmental neurons in 3D space. Automated cell placement and generation of synaptic connectivity, together with 3D visualization and analysis features, allow the creation and verification of models with greater anatomical accuracy than has been achieved previously with script files. The graphical interface and automated generation of command scripts for NEURON or GENESIS allow models to be built, modified and run without programming, enhancing the accessibility of network models. Model reuse and interchangeability are facilitated through the implementation of a simulator-independent model description based on NeuroML standards (Crook et al., 2007; Goddard et al., 2001). We describe and test the functionality of *neuroConstruct*, by reproducing the results of several published models. Moreover, we extend a 1D model of the granule cell layer of the cerebellar cortex (Maex and De Schutter, 1998) to 3D, showing how models can be verified against known anatomy, and provide an example of behavior, previously observed *in vivo* (Vos et al., 1999), that cannot be captured in the original 1D model.

Results

Outline of Application

neuroConstruct is a desktop application written in the platform-independent Java programming language, that can be used to generate network models of compartmental neurons with many biologically realistic features. These including realistic cell morphologies, voltage- and ligand-gated ion channels, cell densities, 3D synaptic connectivity patterns and gross structure of different brain regions. This is achieved through a ‘point and click’ GUI that is accessible to non-programmers. Cell and network models built in *neuroConstruct* can be automatically simulated on either the NEURON or GENESIS platform. The latest version of the application, including the models described in this paper and a number of tutorials and extra documentation, is freely available for download from *www.neuroConstruct.org*.

The functionality of *neuroConstruct* can be grouped into five main areas (Figure 1A).

1) *Importation and validation of morphologies*. Reconstructed neuronal morphologies, commonly used in conductance based neuronal models, can be imported into *neuroConstruct* in various formats (e.g. Neurolucida) and automatically checked for errors. More abstract morphologies, such as in Figure 1B, can be created manually, when a smaller number of compartments is used to represent the cell.

2) *Creation of simulator independent conductance-based cell models*. Explicit modeling of detailed cellular mechanisms, such as the conductance changes produced by voltage- and ligand-gated ion channels, is essential for reproducing many of the complex behaviors observed in real neurons. Cell mechanisms can be defined in *neuroConstruct* in a simulator independent format and cell models can be created by specifying the complement and density of membrane conductances on the cell structure (Figure 1B).

3) *Network Generation*. Once models of the various cell types within a brain region have been created in *neuroConstruct*, they can be placed within a region of 3D space at a specified density.

Layered structures, such as the cortex, can be created from stacks of contiguous regions. Once the cells are arranged, synaptic connections can be generated according to specified sets of rules.

4) *Simulation Management*. Network simulations are carried out by automatically generating script files for one of the currently supported simulator packages (NEURON or GENESIS). The simulation can be executed and the results stored in text files.

5) *Network Analysis*. Network simulations can be loaded back into *neuroConstruct,* which provides several functions for network visualization (Figure 1C) and analysis. For more specialized analysis, script files are created that allow simulation data to be imported into commonly used numerical analysis packages (MATLAB/Octave or IGOR Pro/NeuroMatic).

Description of Functionality and Validation of Application

Neuronal Morphology

Neuronal models with complex morphologies have been used to investigate various aspects of synaptic integration and neuronal excitability (De Schutter and Bower, 1994; Destexhe and Pare, 1999; Golding et al., 2005; Hanson et al., 2004; Jarsky et al., 2005; Mainen et al., 1995; Migliore et al., 1995; Poirazi et al., 2003; Rapp et al., 1996; Schaefer et al., 2003; Vetter et al., 2001), and public databases have been produced that contain examples of anatomical reconstructions of stained neurons (Ascoli, 2006; Cannon et al., 1998). However, using such morphology files in compartmental models is complicated by the fact that they are often in different formats, anatomical and electrical compartments are not equivalent and there are subtle differences in how the morphological information is used by different simulators. To overcome these problems *neuroConstruct* can import and visualize morphology files with different formats (Figure 2A), including Neurolucida (*.asc), GENESIS readcell compatible format (*.p), most NEURON/ntscable generated morphology files (*.nrn or *.hoc), and Cvapp (*.swc) format (Cannon et al., 1998). The two main objects used for specifying compartmentalized morphologies in *neuroConstruct* are sections, which are unbranched cables with uniform biophysical properties, and segments, which define the 3D points along sections (Figure 2B; Experimental Procedures). This simulator independent representation of the morphology allows the same model to be mapped onto different simulator structures. *neuroConstruct* also has a recompartmentalization function that can reduce the total number of compartments, while conserving morphological features such as total membrane area and section length (Figure 2B; Experimental Procedures), thereby speeding up simulations (Supplementary Figure 1). Large scale networks of thousands of neurons often use cell models with fewer compartments to minimize the computational overhead (Santhakumar et al., 2005; Traub et al., 2005). These simple morphologies (Figure 1B,C) can also be created manually in *neuroConstruct* and handled in the same way as for more detailed cells.

The simulator-independent morphological representation within *neuroConstruct* is closely related to MorphML, a new standard for describing neuronal morphologies (Crook et al., 2007). MorphML is based on XML (eXtensible Markup Language), and is the core of Level 1 of the NeuroML framework (Goddard et al., 2001) (http://www.neuroml.org/). Key advantages of using XML are the facilitated exchange of information between different applications, the simple validation of morphology files and the ability to include structured metadata describing the contents of the file.

Cell Mechanisms

Neuronal signaling is mediated by a variety of subcellular, membrane and synaptic mechanisms. Models of cellular mechanisms can be simple, such as a synaptic conductance waveform, or more complex like Hodgkin-Huxley formulations of voltage-gated conductances, which depend on both voltage and time, and their conductance density can be nonuniformly distributed over the cell membrane. Such models form a core part of any conductance based neuronal simulation, but their implementation is one of the more complicated aspects of using existing simulation packages such as NEURON and GENESIS. Although the mathematical framework used to describe such mechanisms (e.g. maximum channel conductance, reversal potential, rate equations) is general and familiar to many neuroscientists, it is implemented in the two simulators in different ways, which can act as a barrier to the creation and exchange of models.

Models of cell mechanisms are implemented in *neuroConstruct* using a ChannelML based description, which forms part of Level 2 of the NeuroML framework. Figure 3 shows how a ChannelML file, in this case describing a synaptic conductance mechanism, can be used. It consists of an XML file containing the physiological parameters in a structured format that can be validated against a specification, reducing the probability of errors. Information in XML files can easily be transformed into other formats with an XSL (eXtensible Stylesheet Language) mapping file (Figure 3). We have created XSL files which map ChannelML descriptions of cell mechanisms onto NMODL (Hines and Carnevale, 2000) format for NEURON and onto the appropriate object in a GENESIS script file. The simulator-independent XML format promotes compatibility with future simulators: for each newly supported simulator, a single XSL file needs to be created which maps the files onto its specialized format. The nature of XML also allows translation of the file into HTML, allowing the cell mechanism to be presented in an easy to read format with metadata including units, references and an explanation of the model formalism. Moreover, this facilitates online archiving and plots showing the properties of the cell mechanism can be generated, further increasing the transparency of the data in the file.

A number of ChannelML templates including Hodgkin-Huxley type models of ion channels and synaptic mechanisms are included with *neuroConstruct*. The values of the parameters in these models can be easily modified through the GUI to match the channel kinetics in a particular cell type, either from a published model or directly from experimental measurements of these parameters (e.g m_∞, τ_m, h_∞ and τ_h for a Hodgkin-Huxley type model of a Na^+ channel). However, ChannelML specifications are still in development and do not cover all possible channel descriptions that have been used in published models. To allow for unsupported channel models and to provide greater backwards compatibility with older published models, channel mechanism files in NMODL (*.mod files) or GENESIS script (e.g. tabchannel based) can be incorporated into cell models created with *neuroConstruct*, but simulations can then only be run on the platform for which the mechanism was written. For example, synaptic plasticity mechanisms and Markov models, which are not presently covered by the current ChannelML specifications, can still be incorporated into a model built with the current version of *neuroConstruct* by inserting an appropriate NMODL file.

The use of different systems of units can lead to errors in translation between the different simulators. GENESIS relies on physical quantities being in a consistent set of either SI units or physiological units (cm, ms, mV etc.). NEURON has its own system of units based on physiological units. Conversion between these systems of units is handled automatically by *neuroConstruct*.

Creation of Cell Models

Once a cellular morphology has been created in *neuroConstruct*, either through importation of a reconstructed morphology, or the creation of a simplified abstract morphology, groups of sections can be defined to distinguish axons, somata and dendrites. Subgroups of sections such as proximal, oblique and apical dendrites can also be defined. Once the cellular mechanisms have been added to a project, their distribution can be specified for each of these cell regions. For example, a non-uniform channel density can be implemented by varying the conductance density in each group (Figure 1B). Ion concentration mechanisms (e.g. activity-dependent intracellular Ca^{2+} concentrations) can also be added in this way, as can passive electrical properties (specific membrane capacitance and specific axial resistance), allowing spine densities to be simulated without additional compartments. If there is no adequate model for a particular cell type available, a new cell model can be created from experimental data using *neuroConstruct* by modifying the

ChannelML templates or adding native code. However, developing cell models generally requires the adjustment of model parameters to reproduce cell behavior. Such model optimization can be currently carried out with *neuroConstruct* using a manual parameter search, but if many parameters need to be changed, automated optimization (e.g. using an evolutionary algorithm) may be necessary. At present this requires writing specialist NEURON or GENESIS code or controlling the model optimization with another program (e.g. C++ or MATLAB; see Discussion).

It is possible to explicitly simulate action potential propagation (APP) along an axon with Na^+ and K^+ conductances in *neuroConstruct*, but it is often convenient to simply calculate the time the action potential takes to get from the soma to the synaptic terminal, since this reduces computational overhead. Moreover, the speed of APP is more likely to be experimentally measured than the properties, distribution and densities of ion channels along the axon. APP Speed can be specified for cells in *neuroConstruct*, and the synaptic delay arising from APP is calculated from this and the axonal morphology.

Comparison of a Cell Model on Two Independently Developed Simulators

To test *neuroConstruct*'s simulator-independent representation of channel mechanisms, we have recreated a published model of a cerebellar granule cell (GrC), originally written in GENESIS (Maex and De Schutter, 1998), and compared its properties on the NEURON and GENESIS simulators. The model, which was developed from earlier work in the turtle (Gabbiani et al., 1994), contains multiple ion conductances, including a fast inactivating Na^+ conductance, three K^+ conductances (delayed rectifier, A type and Ca^{2+} dependent), a high voltage activated Ca^{2+} channel and a hyperpolarization activated H conductance. The model also has a passive leak conductance and an exponentially decaying pool of calcium. In the *neuroConstruct* version all channels are specified in ChannelML. Figure 4A compares membrane potential during a depolarizing current step for simulations run on GENESIS and NEURON. Even at the end of the action potential train there is only a small difference between the two simulators. The timing of the final action potential depended strongly on the integration time step and the two simulators converged to values separated by less than 1 ms after a 500 ms simulation run (Figure 4B). The root mean square difference between the voltage traces decreased steadily over the range of commonly used simulation time steps (Figure 4C), indicating the traces as a whole, not just the final spike, converged. The internal state variables were also similar on each of the simulators under these conditions (Figure 4D). To investigate how significant the difference between the simulators was, we compared the RMS error to that produced by a simulation where the Na^+ channel conductance had been altered by only 1% (Figure 4C, dashed line). The error due to the difference in simulator choice is much smaller than this error, and thus insignificant when taking into account biological variability and the likely uncertainty in a measured conductance density.

We also compared simulations on NEURON and GENESIS of a morphologically complex neuron (Mainen et al., 1995), (Figure 2A), to test the simulator-independent representation of both the channel mechanisms and the morphology. Simulations on both simulators closely reproduced the results of the original model (Supplementary Figure 1). These results confirm that model descriptions in *neuroConstruct* produce similar results on the NEURON and GENESIS simulators.

Construction of Networks of Neurons in 3D Space

Several models of brain regions have been built, but most have circumvented the difficulties of reproducing biologically accurate connectivity by using a simplified network structure with logical connections, or by reducing the number of dimensions. Construction of 3D network models of different brain regions with more anatomically realistic synaptic connectivity is a core feature of *neuroConstruct*.

Cell Placement in 3D

Cells in *neuroConstruct* are arranged in Cell Groups, which are created by specifying the cell type, the 3D region in which the cells are found and the packing pattern used to fill the space. Regions can currently be boxes, spheres, cylinders or cones and multiple regions can be used to create composite structures such as the layers found in the cerebellum (Figure 1C) and cortex. Cells can be packed in these regions in a number of different ways. For brain areas with high cell densities possible patterns include: cubic close packing for maximum density in 3D space; evenly spaced packing in 3D with cell body centers aligned; hexagonal planar patterns; single cells placed at a specified location; cells placed in a one dimensional line. However, for many brain regions the random cell placement option is more realistic (Figure 5A). The number of cells in a specified region can be set and whether cells should avoid the space occupied by existing cell bodies or can overlap can be specified. This allows cell densities to be matched to experimentally measured values for a particular brain region.

Generation of Synaptic Connectivity Patterns

Once all cell groups are positioned in 3D, synaptic connections can be made. The set of rules specifying synaptic connections between cell groups, and the associated type of synaptic mechanism, is termed a Network Connection. There are two different ways in which Network Connections can be generated in *neuroConstruct*. The first, which we term a Morphology Based Connection, is designed for networks of cells whose axons have invariant morphology, those that have been measured explicitly, or where the axon terminal is large and contacted by dendrites from many cells. This scheme works by creating connections whenever the distance between axonal and dendritic segments of the pre and postsynaptic cells fall within specified bounds. Several other parameters can also be set including the number of synapses per cell, regions on the pre- and postsynaptic cell where synapses are permitted and the maximum and minimum connection lengths. Figure 5A shows how the Morphology Based Connection algorithm can be used to generate connections between simplified models of GrCs and Purkinje cells (PCs) in the cerebellar cortex.

GrC axons consist of an ascending axon and a T-shape bifurcation giving rise to a so called parallel fiber (PF), which passes through the planar dendritic arbor of PCs (Figure 5Ai-ii). The PF sections were specified as potential locations of presynaptic connections and a subset of the PC dendritic sections (Figure 5Ai-ii, red) were specified as possible postsynaptic connection locations. In this case the number of connections between pre and postsynaptic neurons was constrained to a maximum of one (Figure 5Aiii).

The second algorithm, which we term a Volume Based Connection, is designed for cases where the axon is a dense, highly arborized structure, distributed over a specific region of space, as commonly found in the cortex. Figure 5Bi shows a simplified model of a cortical interneuron and a cylindrical volume that defines the bounds of its axonal arborization. The diagram of a simplified pyramidal cell in Figure 5Bii shows the subset of its dendritic tree where connections of that type are permitted. When the cells are placed in 3D, regions of the dendritic trees of a number of pyramidal cells which fall within the axonal arbor of the interneurons are potential candidates for connections. These are made randomly based on the user defined connectivity conditions, which include the number of connections per source cell and the maximum number of connections on each target cell. Other shapes including cones and spheres can be used to define the 3D bounds of axonal arborizations. The probability of making a synaptic connection within this volume can be uniform or be a generic function of the x, y, z coordinates relative to the soma, allowing a preference for local connections.

Finding a suitable route for an axon through a complex 3D space is potentially difficult and computationally demanding. We have circumvented this by providing an option within *neuroConstruct* to allow neuronal processes to take up zero volume thereby reducing the packing problem to cell somata. Local circuits with spatially correlated synaptic connectivity, such as feed-forward inhibition (Yoshimura and Callaway, 2005), can be implemented probabilistically by ensuring axonal and dendritic overlap of the excitatory and inhibitory cells in a particular region. Alternatively, synaptic inputs from a particular input cell or cell group can be targeted to excitatory and inhibitory cell groups located in the same region. A local feed forward inhibitory loop can then be generated by making short range inhibitory synaptic connections onto excitatory cells. This flexibility in the generation of circuits should allow a wide range of spatially nonuniform local circuits to be generated in *neuroConstruct*.

There is also provision to introduce randomness into the amplitude and delay of the synaptic conductance (this can be fixed, or random with a uniform or Gaussian distribution) for both of the connectivity algorithms. Moreover, the spatial location of the network elements can be used to simulate interesting 3D phenomena. We have used the ability to insert native code in *neuroConstruct* and the 3D spatial information contained within models to develop a very simple model of extracellular diffusion. Supplementary Figure 2 shows a 3D network model with a diffusible signaling molecule that transiently inhibits the AMPA synaptic conductances. Changes in the synaptic weight can be visualized directly by replaying the simulation in *neuroConstruct* or exported and plotted as a function of distance from the source. Although preliminary, this simulation illustrates the potential for creating more realistic models of volume based signaling involving NO, canabinoids and neurotranmitters together with the supply and removal of metabolic factors.

Network visualization and input properties

A wide range of network properties can be visualized through the GUI. These range from the overall gross structure of the network down to the location and properties of individual synaptic connections. As network models can vary widely in sizes, there are a number of functions in *neuroConstruct* that allow large networks and cells with complex morphologies to be visualized. These include showing the dendrite and axon as lines, or just showing ball-shaped somas, rather than the full 3D structure of each cell. Figures 5Biii and 8A show these options for simple cells, but they also apply for highly detailed morphologies (e.g. Figure 2A). An adjustable transparency mode is available for visualizing cells deep within networks. This allows an individual cell, defined groups of cells or cells within a defined region to be highlighted (Figures 1C and 7B) and there is also an option to inspect their associated synapses. These functions allow cells to be viewed in large networks of thousands of cell on most standard desktop computers. A number of functions for analyzing the anatomical properties are also available within *neuroConstruct* (see Extension of the 1D cerebellar GrC Layer model to 3D). The cell placement and network connectivity can be imported and exported in NetworkML format (Level 3 of the NeuroML framework). This allows saving of generated networks and also enables networks created with other applications (e.g. custom MATLAB scripts) to be loaded into *neuroConstruct* for visualization and use in simulations.

External activation of a network with defined patterns of stimuli can be achieved in several ways. Any of the generated cell groups can receive two main types of inputs: current steps of specific duration, delay and amplitude, or random trains of synaptic inputs, with a defined input frequency or a range of frequencies. Both of these types of input can be applied to all cells in a group, to a fixed percentage of cells, or to cells inside or outside a defined 3D region. The last option can be used to apply spatially organized input patterns to networks, for example in order to investigate transformation of spatial activity patters between layers.

Verification of Network Models implemented in *neuroConstruct*

We verified the ability of *neuroConstruct* to generate accurate network models by implementing two published network models and comparing the behavior of the *neuroConstruct* version to the original model. The first model tested was a conductance-based model of the cerebellar GrC layer (Maex and De Schutter, 1998). GENESIS simulation scripts for this model were obtained from http://www.tnb.ua.ac.be/models/network.shtml and the network specified by these scripts, consisting of 12 mossy fiber (MF) inputs, 75 GrCs and 4 Golgi cells (GoCs), was recreated in *neuroConstruct* (visualized in Figure 6A). The GrC model outlined previously (Figure 4), and a GoC model with similar ChannelML based channel mechanisms were used in the network (Experimental Procedures). A key conclusion from the original study was that GrC firing becomes entrained by GoC feedback inhibition even though excitation is via a random MF input. Figure 6B shows population spike time histograms of the original model (left) and the model generated in *neuroConstruct*. Although there are minor differences in the exact spike times due to subtle differences in the connectivity, the behavior of the two models is similar: both exhibit a strong tendency for the GoCs to spike in synchrony after around 100 ms, while the GrCs fire in a small time window before these spikes.

Since the neurons in the GrC layer model had only a single compartment and the implementation of the network did not test the section specificity of the connectivity algorithms in *neuroConstruct* (Figure 5), we also recreated a 527 cell model of the dentate gyrus (Santhakumar et al., 2005), which had 4 types of multicompartmental neurons, with 11 channel mechanisms and section specific connectivity, albeit in 1D (Experimental Procedures; visualized in Fig. 6C). A key result in this study was to show that increased MF connectivity (sprouting) could generate epileptiform network activity. Figure 6D shows the raster plots of spike times for the original model and *neuroConstruct* versions of the model. Focal stimulation at 5 ms caused the central 100 granule cells to fire synchronously producing the initial line in the plot. The extra connectivity in the network that mimicked MF sprouting caused the activity to propagate to the other cells in the network over approximately 200 ms in both the original and *neuroConstruct* version. These results show that *neuroConstruct* can faithfully reproduce two of the most advanced published models, demonstrating its ability to recreate models of different brain regions and the validity of the internal implementation of cell mechanisms and neuronal connectivity.

Extension of the 1D cerebellar Granule Cell Layer model to 3D

While recreating existing 1D models is important for validation of the application, such models do not illustrate the ability of *neuroConstruct* to generate network models in 3D space. We have therefore used *neuroConstruct* to extend the 1D network model of the cerebellar GrC layer to 3D. For comparison, we have used the same parameters for the cell and synaptic mechanisms as in the original model (Maex and De Schutter, 1998). The model consists of the GrC layer containing the MF terminals and the GrC and GoC cell bodies and the molecular layer with connections between the PFs and GoC dendrites (Figure 7A). 32 GoC bodies were packed randomly in the GrC layer, and 96 MFs, represented by a single segment for each glomerulus, were placed around these. Finally, 600 GrCs were packed randomly, avoiding the space taken up by the existing cell bodies. The GrC axons projected to the molecular layer region and bifurcated to produce PFs. Synaptic connections were generated with the Morphology Based Connection algorithm (Experimental Procedures). We used the anatomical analysis functions in *neuroConstruct* to check that the network was connected in line with both input parameters and measured anatomical properties. These include the number of connections made by the pre and postsynaptic cells (Figure 7C,D) and the distances between the GrC somata and the MF terminals, which correspond to dendritic length (Figure 7E). In this case dendrites were substantially longer than for real GrCs (Eccles et al., 1967; Ito, 1984) because GrC density in this model was reduced to well below the biologically realistic value. Figure 7B illustrates one of the visualization features of *neuroConstruct*, by highlighting the connections of a single GoC. This example demonstrates that network models can be created in 3D and that their anatomical properties can be quantitatively analyzed in a way that can be directly compared to measured anatomical parameters.

Simulation Management

In addition to specifying the cell structures, cell mechanisms, network connections, and network stimulation protocols, extra information on simulation duration, time step and numerical integration method to be used during simulations is specified through the application interface. Moreover, a number of different parameters can be specified for saving and/or plotting during a simulation (e.g. membrane potential, ion channel conductance/current/state variables, calcium concentration, spike times, etc.). It is also possible to specify ranges of parameters (e.g. stimulation amplitude and duration) over which to run multiple simulations, allowing basic parameter space searches. Before the simulation is run a number of validity checks can be carried out (morphology compliance, appropriate simulation timestep, temperature during simulation, etc.), to catch many of the common sources of error in neuronal simulations, which can increase in number when model elements from multiple sources are used. Simulation scripts are automatically generated by mapping the internal representation of the model into the native format of each of the supported simulators. The simulations are initiated through the *neuroConstruct* interface and run on standard versions of NEURON or GENESIS. There is no interaction between the simulator and *neuroConstruct* during the numerical integration.

At the time of generating the script in the native language of the simulator, *neuroConstruct* also stores a number of important simulation parameters (cell populations, numerical integration methods, random seed used to generate network, etc.) in a structured format. The *neuroConstruct* simulation browser interface facilitates the management of recorded simulations, which is particularly important when multiple simulations are run in a batch job. A *neuroConstruct* project can contain a number of Simulation Configurations to illustrate different aspects of a model. One Simulation Configuration can be used for each key finding (e.g. reproducing each figure in the accompanying paper) or alternatively, network models of different sizes can be included to analyze network scalability. With further development of a network model (e.g. addition of new neuron classes), new Simulation Configurations can be introduced.

Analysis of Network Activity

Network activity can be visualized and the simulation replayed at various levels of detail by reloading the simulation results produced by NEURON or GENESIS back into *neuroConstruct.* Several features for analyzing network activity are illustrated in Figure 8, which shows the 3D GrC layer model with cell processes and connections removed for clarity. At the most basic level of analysis, the voltage of specific cells can be plotted (Figure 8B) or a raster plot of spike times generated (e.g. Figure 6D). A histogram of inter-spike intervals shows that GoCs (Figure 8C, red) and GrCs (Figure 8C, black) in the 3D network model fire with similar intervals as for the 1D model (Figure 6C), with the multiple peaks in the GrC histogram reflecting the fact that GrCs do not fire on every GoC cycle. This similar behavior reflects the similar mean number of inputs per cell type and synaptic properties in the two models. Interestingly, our simple 3D GrC layer model also exhibited spatio-temporal properties that are not present in the 1D model. To quantify these properties we defined two analysis regions, with cells which shared different PF inputs (Figure 8A); GoCs in these regions (beam A and beam B) connect with largely non overlapping sets of GrCs (Figure 7B). As Figure 8B shows, action potentials from cells in the same beam were more closely aligned at the end of the simulation than in different beams (black and red traces, blue and green traces). We investigated this further by comparing the correlation in spike times over the whole 4s simulation run between a GoC in beam A (cell 31, Figure 8A, red) with other GoCs in the same beam and GoCs in beam B. A much higher correlation was found between this cell and the 4 other GoC cells in beam A (Figure 8D) than with the 6 GoCs in beam B (Figure 8E). This behavior is consistent with experimental results comparing simultaneous recordings from GoCs along and across PF tracts (Vos et al., 1999). This simple model demonstrates that network models can be generated and analyzed in *neuroConstruct* with more realistic anatomical properties and behaviors than have been achieved previously.

Several other standard functions for network analysis are built into *neuroConstruct*. However, since some questions may require more specialized analysis tools, extra files are automatically created for easy loading of the data into MATLAB and IGOR Pro, two popular numerical analysis packages. A number of script files are included for quick analysis (e.g. for generating rasterplots, spike histograms, etc.) in MATLAB or GNU Octave (*www.octave.org*), an open source application compatible with MATLAB script files. Files are also generated for importing simulation data into IGOR Pro. These can be analyzed with NeuroMatic (*http://www.neuromatic.thinkrandom.com*), an open source set of functions for IGOR Pro, specifically for analysis of electrophysiological data. Experimental data traces can also be imported into *neuroConstruct* for direct comparison with simulation data.

Discussion

neuroConstruct is a new platform independent software tool for constructing, visualizing and analyzing conductance-based neural network models with properties that closely match the 3D neuronal morphology and connectivity of different brain regions. A user friendly GUI allows models to be built, modified and run without the need for specialist programming knowledge, providing accessibility to both experimentalists and theoreticians studying network function. Models built with *neuroConstruct* are simulator independent and can be automatically mapped onto the NEURON or GENESIS simulation environments for numerical integration. Model components are stored in a simulator-independent XML format, allowing interchange and reuse of model components across simulators. We have demonstrated the functionality of *neuroConstruct* by creating and analyzing a simple 3D network model of the cerebellar GrC layer.

Construction of 3D neural network models

Quantitative measurements of network properties including cell densities, numbers of synaptic connections between cell groups and dimensions of axonal and dendritic fields are available for several brain regions including cortex (Douglas and Martin, 2004; Somogyi et al., 1998; Thomson et al., 2002) and cerebellum (Harvey and Napper, 1991; Sultan and Bower, 1998). However, generating biologically realistic 3D neuronal network models from such data has proved difficult using the direct scripting approach. This is because, unlike many random artificial networks, networks of neurons in the brain exhibit inhomogeneous connectivity probabilities (Lubke et al., 2003), spatial clustering and an enhanced probability of certain multi-cell motifs (Song et al., 2005; Sporns and Kotter, 2004). These are due to a prevalence of local connections and the presence of local circuits (Yoshimura and Callaway, 2005), which are thought to be essential for local computations (Pouille and Scanziani, 2004).

Several core functions within *neuroConstruct* facilitate the generation of 3D network models with increased biologically realism. These include the ability to import neuronal reconstructions in multiple file formats and the automated placement of cells in defined 3D patterns. Two algorithms enable synaptic connectivity to be generated in 3D space with subcellular specificity. The first was designed for cell models with fully reconstructed axons, axons that are rather invariant (e.g. PF-PC and Schaffer collateral-CA1 synaptic connections (Shepherd, 1998)) and large terminals that innervate many postsynaptic cells (e.g. cerebellar MF). The second is designed for cells with dense axonal arborizations that project over a particular region of 3D space (e.g. spiny stellate cells in cortex (Lubke et al., 2003) and various interneurons in cortex, hippocampus and cerebellum (Shepherd, 1998)). While some spatial clustering in connectivity will be generated by these connection algorithms, a stronger weighting of local connections can be achieved by applying a connection probability that decays with distance. Local subcircuits with a common input (e.g. feed forward inhibition) can be generated by combining specific excitatory inputs onto local subgroups of excitatory and inhibitory cells and a local inhibitory connection. 'Small world' properties (Watts and Strogatz, 1998) could also potentially be generated by creating two types of network connection, one with highly interconnected short-range connections and the other with sparser long-range connections. Splitting synaptic connections into different subgroups could be used to generate highly skewed lognormal distribution of synaptic weights (Song et al., 2005). The biological realism of networks generated in *neuroConstruct* with these approaches can be examined with the advanced visualization and anatomical analysis functions, which allow direct quantitative comparison of the model structure with anatomy (Figure 7).

Model accessibility and reuse

The accessibility of large scale neural network models is currently limited by the fact that they are large specialized programs, often written in different languages, which run on different simulators (Maex and De Schutter, 1998; Santhakumar et al., 2005; Traub et al., 2005). Modifying and rerunning such programs can be difficult and requires specialist programming knowledge. While recent efforts have been made to improve accessibility with the development of GUI interfaces in NEURON and GENESIS, network models are usually written and run using script files. We have addressed this issue by developing a GUI for *neuroConstruct* that allows networks to be built, modified, visualized and analyzed. Moreover, *neuroConstruct* automatically writes the simulation code and runs it on the chosen simulator (NEURON or GENESIS). No programming knowledge is therefore needed to create, modify, run and analyze a large network simulation using *neuroConstruct*. These features of *neuroConstruct* make neural network simulations more accessible to non programmers, thereby providing a new tool for both research and teaching.

The latest NeuroML specifications (Crook et al., 2007; Goddard et al., 2001) form the core of our simulator-independent model descriptions. Increased adoption of these standards, which are also used in the latest version of NEURON and which will form the basis of model descriptions in GENESIS 3/MOOSE (currently under development at http://sourceforge.net/projects/moose-g3), will promote greater model reuse and collaboration between research groups on cellular and network models.

Current limitations and future developments

The present version of *neuroConstruct* generates models that can be run on single processor machines. The scale of simulations that can be run and visualized is therefore limited by the processor and video memory, respectively. We have run simulations of up to 5,000 multicompartmental neurons (50,000 simulated compartments) on a single processor with 2 GB of memory. This simulation could be visualized with a 128MB video card. For larger simulations the processor and video memories would have to be scaled up accordingly (we have visualized 50,000 multicompartmental neurons with a 256MB video card and 8GB of RAM). Simulations such as that illustrated in Figure 8 (728 compartments) can take 1-2 hours for a 4-sec simulation on a single processor, so if larger scale simulations or extensive parameter searching is required parallelization may be necessary. We plan to include features in the next major release of *neuroConstruct* for distributing multiple individual simulations using CONDOR (http://www.cs.wisc.edu/condor) and parallelization of the simulations using the recently developed parallel version of NEURON (Migliore et al., 2006), which is being used for simulations in the Blue Brain Project (Markram, 2006). Parallel simulations will also be a key feature in GENESIS 3/MOOSE and this platform will be supported in future versions of *neuroConstruct*.

Presently, functions beyond the scope of the *neuroConstruct* GUI interface can be added with appropriately inserted native NEURON or GENESIS code. We intend to improve this flexibility by including a Python-based (http://www.python.org) scripting interface to *neuroConstruct*. This will allow greater access to the internal variables of a model allowing easier parameter searching and model optimization. *neuroConstruct* is closely linked to the NeuroML initiative, and future extensions to ChannelML will provide a wider scope for implementing new channel types and plasticity mechanisms and will include interaction with Systems Biology standards (Finkelstein et al., 2004; Kitano, 2002) such as with SBML (Hucka et al., 2003) and CellML (Lloyd et al., 2004). This opens the possibility of interaction with 3D diffusion-reaction packages like MCell (http://www.mcell.cnl.salk.edu) and VCell (http://www.nrcam.uchc.edu), although the difference between the morphological representations in these and compartmental neuronal simulators could be restrictive. On the network connectivity side, work is ongoing in the NeuroML project with the developers of Topographica (Bednar et al., 2004), NEST (Diesmann and Gewaltig, 2002) and Neurospaces (Cornelis and De Schutter, 2003), to gain a broad consensus on descriptions of network connectivity.

Towards more realistic models of brain function

By including 3D spatial information, models built in *neuroConstruct* can be used to investigate the role of diffusion in brain function. We have used this information to implement a simple model of a diffusible substance that transiently depresses synaptic conductances (Supplementary Figure 2). Although oversimplified, this proof of concept simulation illustrates how *neuroConstruct* could be used to examine volume signaling mechanisms such as NO and the relationship between metabolism and neuronal activity, which underlies functional imaging (Attwell and Iadecola, 2002).

The modular structure of *neuroConstruct* allows new features to be included as they are developed, ensuring compatibility with future advances in our understanding of brain mechanisms. Extensions currently envisioned include automated generation of heterogeneous cell morphologies, using approaches similar to L-Neuron (Ascoli et al., 2001) or NeuGen (Eberhard et al., 2006). Moreover, detailed reconstruction of large blocks of tissue using serial block face scanning EM could provide more accurate information about the 3D circuit topology and local spatial arrangements of synapses. Indeed, it even opens the possibility of including ultrastructure at 30 nm resolution (Briggman and Denk, 2006), which would allow more detailed diffusion models. If such EM data stacks were converted into compartmentalized anatomical objects and stored in MorphML format, they could then be directly imported into *neuroConstruct* and used to build network models. This combination of technologies would open the possibility of 3D network modeling with unprecedented levels of biologically realism.

Conclusion

By providing a tool for building, modifying, visualizing and analyzing network models in 3D space using a user friendly GUI, without the need for programming, *neuroConstruct* increases the accessibility of modeling brain function. The new functionality and accessibility provided by *neuroConstruct* makes it suitable for both experimentalists and theoreticians. It can also be used for teaching network function in health and disease. The 3D models generated will allow simulations of increased biological realism, enabling more direct comparisons with results from new experimental methods for measuring neural activity in 3D at high spatial and temporal resolution.

Experimental Procedures

Simulator-independent morphological information

NEURON and GENESIS deal with complex neuronal morphologies differently. In NEURON unbranched neurites are specified by a sequence of 3D points and diameters, outlining their shape (termed sections). Membrane surface area is computed from these values, as is the axial resistance for a given length. Sections can be subdivided into evenly spaced segments (by specifying the variable *nseg*) for numerical integration, which is carried out at their centre points. In contrast, GENESIS uses a single unit, termed a compartment, as the building block for both neuronal morphology and numerical integration. *neuroConstruct* uses a simulator-independent representation of neuronal morphology, which allows translation between each of these formats. A section is defined in *neuroConstruct* as an unbranched part of a dendrite or axon with uniform biophysical characteristics along its length. Each section contains a number of segments, each of which is specified by a 3D end point and diameter. *neuroConstruct* sections map directly onto NEURON sections, and the 3D points along their lengths are given by *neuroConstruct* segment endpoints. However, the number of anatomical segments in *neuroConstruct* is not the same as *nseg* in NEURON. Instead, the nseg value is a property of the *neuroConstruct* section. For GENESIS, each segment is mapped to a single compartment. There is a one to one mapping between *neuroConstruct* and MorphML format (Crook et al., 2007); segments are mapped to segment elements and sections are mapped to cable elements.

neuroConstruct can import morphologies in NEURON and GENESIS format and from neuronal reconstruction tools such as Neurolucida, which handles branching structures in a similar way to NEURON (Crook et al., 2007). Automatic checks in *neuroConstruct* signal potential problems with morphologies: reconstructions can contain dendritic segments of zero diameter or zero length and dendritic subtrees can be detached from the main cell. Manual editing of the imported morphologies is possible and *neuroConstruct* can also recompartmentalize neurons, allowing morphologies originally in Neurolucida or NEURON format (e.g. 4000-5000 segments) to be mapped onto GENESIS models with a reduced number of compartments (~500-1000; Supplementary Figure 1). The overall structure of the cell is preserved, and each section (with ~5 to 10 segments) is mapped onto two single segment cylindrical sections (Figure 2B) corresponding to GENESIS compartments. The radii of the cylinders are chosen to preserve cell membrane surface area, total length and axial resistance along sections.

Implementation of cell mechanisms

Models of cell mechanism can be specified in one of two ways; either using a simulator independent ChannelML description of the mechanism or by providing a file in the native script language of the simulation platform (NMODL or GENESIS script) which creates and sets the parameters for the object enabling the mechanism. As mentioned in the results, the ChannelML representation is mapped onto the simulator specific description using an XSL file (Figure 3). For NEURON the NMODL files are compiled automatically before running the simulation. Cell mechanism script files will only be created if they are present on cells in the simulation. Numerous examples of channel and synaptic mechanisms in ChannelML can be obtained at http://www.morphml.org:8080/NeuroMLValidator and these can be converted online to NMODL and GENESIS formats to show how the elements in a ChannelML file are mapped onto simulator specific objects.

Connectivity algorithms

Once cells are arranged in 3D, connections can be created between two cell groups, or within a single cell group. Connections can be defined relative to the pre or postsynaptic cell. Each source cell is assigned a number of connections, which can be fixed or variable within set bounds. One or more synaptic mechanisms are associated with the connection and these can have variable or fixed weights and internal delays. With the Morphology Based Connection the target cell can be chosen at random, can be the closest available cell or the closest cell from a random selection of n locations. Maximum and minimum connection lengths can also be set. For the Volume Based Connection an axonal arborization volume is defined and any appropriate target segment falling within this region is a candidate for a connection. Non uniform connectivity is generated by allocating putative connection locations with a connection probability that depends on radial distance or x, y, z coordinated relative to the source soma. The spatial dependence function can be defined by the user. A connection is made if a random number (0 -1) is ≤ the connection probability, otherwise another random location is picked until all connections are made.

Network model details

In the 1D GrC model, MFs were explicitly modeled as single compartments, firing Poisson trains of random spikes. 12 MFs were placed along a 900 μm line and the 75 GrCs and 4 GoCs were placed above these, displaced vertically to facilitate visualization of the connections. Synaptic connections were made between the MFs and GrCs according to the following conditions: both AMPA and NMDA receptor synaptic mechanisms are present at each connection, with random weights (multiplicative factors of a physiological synaptic conductance) in the ranges 5.1 to 6.9 and 3.4 to 4.6, respectively; connections were made from each GrC to four random, unique MFs within a horizontal distance of 400 μm, resulting in a radius of influence of 5 times MF separation. This differed from the combinatorial expansion approach in the original paper, but resulted in equivalent number of inputs onto each GrC, a similar radius of influence for each MF and more anatomical realism in the stochasticity of the connection. The connections between the GrCs and GoCs were created using an AMPA receptor synaptic mechanism specific for this connection with a random weight in the range 0.51 to 0.69, and with each GrC connecting to all four GoCs. The inhibitory connections between the GoCs and GrCs have the following properties: a $GABA_A$ receptor synaptic connection with a random weight of 38.25 to 51.75; each GrC was connected to closest GoC.

In our 3D GrC layer model (Figure 7), GrCs consisted of a soma with the same properties as used in (Maex and De Schutter, 1998), and had a bifurcated axon which formed the parallel fibers (PFs). The segments for these axons were not explicitly simulated; an Action Potential Propagation Speed was used to provide the extra synaptic delay. GoCs also consisted of a soma taken from the previous model, but have a single dendrite of similar length to the ascending segment of the GrC. The extent of GrC layer simulated was 500 µm along the PF axis, 1 mm wide and 50 µm deep. The number of cells reflects a scaling up of 8 times of the 1D network model. There was a difference in the distribution but not the modal number of MF connections to each GrC in the 3D network model (max 7, min 3, mean 4), reflecting experimentally measured numbers (Eccles et al., 1967).

The network model of the dentate gyrus (Figure 6 C) created in *neuroConstruct* was based on that generated by the NEURON script published online to accompany the paper (Santhakumar et al., 2005) (http://senselab.med.yale.edu/senselab/modeldb/ShowModel.asp?model=51781). Since only 5 of the 11 channel mechanisms used in the model were covered by the current ChannelML specifications, implementation of this model in *neuroConstruct* reuses existing NMODL files for the remaining channel mechanisms. The *neuroConstruct* generated network reproduces the more anatomically realistic topographic strip rather than the topographic ring (Figure 3 of (Santhakumar et al., 2005)) used to eliminate edge effects. The downloaded simulation scripts were altered to position the perforant path at GrCs 200-299, and these were also the cells stimulated in our network model. Selection of postsynaptic target cells was determined by setting maximum and minimum distances for synaptic connection lengths along the line of cell bodies using the Morphology Based Connection Algorithm. This method of network generation, together with the different network topology, resulted in similar means but some differences in the standard deviations of synaptic convergence (Table 3 of (Santhakumar et al., 2005)). The small divergence of behavior when the wave of activation reached the end of the strip (Figure 6D) is also due to the change from a ring structure to a linear network topology.

Figure 1: Overview of neuroConstruct

(A) Schematic view of the main functionality of *neuroConstruct*.

(B) Main *neuroConstruct* GUI showing a single abstract cell with a Na^+ channel conductance density that varies on different parts of the cell membrane.

(C) Main interface to *neuroConstruct* showing the visualization of a simple network using transparency feature to highlight a single cell and its connections.

Figure 2: Detailed Cell Morphologies in *neuroConstruct*

(A) A detailed reconstruction of a neocortical pyramidal cell (Mainen et al., 1995) imported into *neuroConstruct* from a NEURON morphology file.

(B) Detail of a small part of a dendritic tree. In the upper view, all 3D detail present in the original morphology file is included. Sections (between the blue spheres) contain a number of 3D points with associated diameter, each of which is the endpoint of a segment (small grey conical frusta). NEURON uses this information to compute membrane area and axial resistance, but only performs numerical integration at a limited number of points along the section (red spheres). As GENESIS compartments have no internal structure, a representation of the dendritic tree in a reduced number of sections must be used (lower view; Experimental Procedures).

Figure 3: Use of ChannelML for Specifying Cellular Mechanisms.

A ChannelML file (the code fragment shows the parameters needed to specify a double exponential time course synapse) can be converted into script files in the native language of various neuronal simulators (currently NEURON and GENESIS), using an XSL (eXtensible Stylesheet Language) file for each mapping. HTML representations of the XML file provide a more readable view of the mechanism and associated metadata, and plots can be generated to view its properties.

Figure 4: Comparison of Granule Cell Model Implemented in *neuroConstruct* and Run in

NEURON and GENESIS

(A) Membrane potential response of a granule cell model (Maex and De Schutter, 1998) to a 500 ms current pulse of 10 pA at a simulation time step of 0.01 ms.

(B) Dependence of timing of last action potential on integration time step. The final peak of the traces in A are initially out of phase, but as the numerical integration time steps become shorter the peak times converge.

(C) Dependence of the root mean square of the difference between traces on integration time step. The minimum at 0.001 ms is due to the final peaks overlapping before converging at slightly different times in each simulator. The dotted line shows the RMS error between the original cell on GENESIS and one with a 1% difference in Na^+ conductance density.

(D) Values of some of the internal state variables as a function of time (ms) displayed as screenshots from NEURON (left) and GENESIS (right).

Figure 5: Connectivity Schemes Used to Generate Network Connections between Cell Groups in *neuroConstruct*

(A) Simplified morphology of a cerebellar granule cell (GrC) (i) including soma and axon. Parallel fiber sections, highlighted in red, indicate presynaptic sections where synapses are permitted. Simplified morphology of a Purkinje cell (PC) (ii) with red dendritic sections showing postsynaptic sections where synapses are permitted. Connections between multiple GrCs and a PC made using the Morphology Based Connection algorithm (iii). Green and red spheres show the sites of pre and post synaptic connection, respectively.

(B) Simplified morphology of a cortical interneuron (i) including soma, dendrites and a cylindrical volume (grey shading) defining boundaries of the axonal arbor. Simplified morphology of a cortical pyramidal cell (ii) with red dendritic sections showing postsynaptic sections where synapses are permitted. 3D Connections between multiple interneurons and pyramidal cells made using the Volume Based Connection algorithm (iii). Sites of pre and post synaptic connections are linked by lines changing from green to red.

Figure 6: Implementation of Existing Network Models in *neuroConstruct*.

(A) Visualization of a 1D Granule cell (GrC) layer network model from (Maex and De Schutter, 1998). Mossy Fibers (MFs, bottom) are connected via excitatory synapses to GrCs (middle), which were in turn connected to Golgi cells (GoCs, top). The GrCs receive inhibitory connections from GoCs.

(B) Spike time histograms (bin size 1 ms) as produced by the script files released with the original publication (left) and for the *neuroConstruct* model of the network (right). Spikes for the GrCs are in black and the GoCs are in red.

(C) Replication of a network model of the dentate gyrus (Santhakumar et al., 2005). The model consists of (from the top down) 500 GrCs with two dendritic branches, 6 basket cells, 15 mossy cells and 6 hilar cells. The 10,000+ synaptic connections have been removed for clarity. The network receives a brief perforant path focal stimulation, mainly on the central 100 GrCs. Cell coloring reflects network activity 110 ms after stimulation.

(D) Raster plots of dentate gyrus GrC activity in the original published model and in the *neuroConstruct* implementation of the network.

Figure 7: Extension of a 1D Model of Granule Cell Layer to 3D

(A) Visualization of a 3D granule cell (GrC) model based on a published 1D model (Maex and De Schutter, 1998). Mossy fiber (MF) terminals (blue), GrC somas (orange) and Golgi cell (GoC) somata (green) are packed in a 3D region (500μm in PF direction, 1mm parasagittally, 50μm in thickness) representing a section of the GrC layer of the cerebellar cortex. The ascending segment and parallel fibers of the GrCs extend into the molecular layer, as do the single dendrites of the GoCs.

(B) A single GoC and associated network connections highlighted using the the transparency option in *neuroConstruct*. The range of connection lengths is larger than the experimentally reported values for the GoC dendritic tree (~200 μm, (Dieudonne, 1998)) due to the reduced cell density.

(C) Histogram showing the distribution of numbers of synaptic connections received by GrCs. A truncated Gaussian distribution of numbers of connections (mean 4, min 3, max 7) was made to different MFs, the closest terminals to the GrC somas being chosen. Axis variables shown at bottom of window in C-E.

(D) Histogram showing the distribution of numbers of synaptic connections made by the 96 MFs in the network.

(E) Distribution of distances between connected MF terminals and GrC somas, corresponding to dendritic length.

Figure 8: Network Analysis of a 3D Granule Cell Layer Model

(A) View of 3D granule cell (GrC) layer model showing only the cell bodies. Two regions are identified, beam A and beam B, which experience different parallel fiber (PF) input.

(B) Voltage traces of four Golgi cells (GoCs) at the end of a 4 sec simulation run, with network connectivity as outlined previously and 50 Hz Poisson input to the mossy fibers (MFs). Black trace is from cell 31, red trace is from cell 5, both of which are found in beam A. Cells 13 (blue) and 15 (green) are in beam B. Axis variables shown at bottom of window in B-E.

(C) Interspike interval histograms of the GrCs (black) and GoCs (red). The peak at approximately 40 ms reflects the observed average firing rate of the GoCs of 23.8 Hz, the single peak resulted from regular GoCs spiking. The GrCs have a lower average firing rate and do not fire on every GoC cycle, hence the multiple peaks in the histogram

(D) Crosscorrelation between cell 31 and the other 4 GoCs in beam A, each color graph representing a different cell. The y axis represents the probability of finding a spike in the other cell in a time window of 1 ms with the specified offset.

(E) Crosscorrelation between cell 31 and the 6 GoCs in beam B, with identical axes to D.

References

Alger, B. E. (2002). Retrograde signaling in the regulation of synaptic transmission: focus on endocannabinoids. Prog Neurobiol *68*, 247-286.

Ascoli, G. A. (2006). Mobilizing the base of neuroscience data: the case of neuronal morphologies. Nat Rev Neurosci *7*, 318-324.

Ascoli, G. A., Krichmar, J. L., Nasuto, S. J., and Senft, S. L. (2001). Generation, description and storage of dendritic morphology data. Philos Trans R Soc Lond B Biol Sci *356*, 1131-1145.

Attwell, D., and Iadecola, C. (2002). The neural basis of functional brain imaging signals. Trends Neurosci *25*, 621-625.

Bednar, J. A., Choe, Y., De Paula, J., Miikkulainen, R., Provost, J., and Tversky, T. (2004). Modeling cortical maps with Topographica. Neurocomputing *58-60*, 1129-1135.

Berends, M., Maex, R., and De Schutter, E. (2004). A detailed three-dimensional model of the cerebellar granule cell layer. Neurocomputing *58-60*, 587-592.

Bower, J. M., and Beeman, D. (1997). The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System, Springer, New York).

Briggman, K. L., and Denk, W. (2006). Towards neural circuit reconstruction with volume electron microscopy techniques. Curr Opin Neurobiol *16*, 562-570.

Buonomano, D. V. (2000). Decoding temporal information: A model based on short-term synaptic plasticity. J Neurosci *20*, 1129-1141.

Bush, P. C., Prince, D. A., and Miller, K. D. (1999). Increased pyramidal excitability and NMDA conductance can explain posttraumatic epileptogenesis without disinhibition: a model. J Neurophysiol *82*, 1748-1758.

Cannon, R. C., Turner, D. A., Pyapali, G. K., and Wheal, H. V. (1998). An on-line archive of reconstructed hippocampal neurons. J Neurosci Methods *84*, 49-54.

Cornelis, H., and De Schutter, E. (2003). NeuroSpaces: separating modeling and simulation. Neurocomputing *52-4*, 227-231.

Crepel, F., and Jaillard, D. (1990). Protein kinases, nitric oxide and long-term depression of synapses in the cerebellum. Neuroreport *1*, 133-136.

Crook, S., Gleeson, P., Howell, F., Svitak, J., and Silver, R. A. (2007). MorphML: Level 1 of the NeuroML Standards for Neuronal Morphology Data and Model Specification. Neuroinformatics *(in press)*.

Davison, A. P., Feng, J., and Brown, D. (2003). Dendrodendritic inhibition and simulated odor responses in a detailed olfactory bulb network model. J Neurophysiol *90*, 1921-1935.

De Schutter, E., and Bower, J. M. (1994). An active membrane model of the cerebellar Purkinje cell. I. Simulation of current clamps in slice. J Neurophysiol *71*, 375-400.

De Schutter, E., Ekeberg, O., Kotaleski, J. H., Achard, P., and Lansner, A. (2005). Biophysically detailed modelling of microcircuits and beyond. Trends Neurosci *28*, 562-569.

Destexhe, A., and Marder, E. (2004). Plasticity in single neuron and circuit computations. Nature *431*, 789-795.

Destexhe, A., and Pare, D. (1999). Impact of network activity on the integrative properties of neocortical pyramidal neurons in vivo. J Neurophysiol *81*, 1531-1547.

Diesmann, M., and Gewaltig, M.-O. (2002). NEST: An Environment for Neural Systems Simulations, Vol Forschung und wisschenschaftliches Rechnen, Beitrage zum Heinz-Billing-Preis 2001, Gottingen: Ges. fur Wiss. Datenverarbeitung).

Dieudonne, S. (1998). Submillisecond kinetics and low efficacy of parallel fibre-Golgi cell synaptic currents in the rat cerebellum. J Physiol *510 (Pt 3)*, 845-866.

Douglas, R. J., and Martin, K. A. (2004). Neuronal circuits of the neocortex. Annu Rev Neurosci *27*, 419-451.

Eberhard, J. P., Wanner, A., and Wittum, G. (2006). NeuGen: A tool for the generation of realistic morphology of cortical neurons and neural networks in 3D. Neurocomputing *70*, 327-342.

Eccles, J. C., Ito, M., and Szentagothai, J. (1967). The cerebellum as a computational machine (Berlin, Heidelberg, New York, Springer-Verlag).

Finkelstein, A., Hetherington, J., Li, L. Z., Margoninski, O., Saffrey, P., Seymour, R., and Warner, A. (2004). Computational challenges of systems biology. Computer *37*, 26-33.

Gabbiani, F., Midtgaard, J., and Knopfel, T. (1994). Synaptic integration in a model of cerebellar granule cells. J Neurophysiol *72*, 999-1009.

Goddard, N. H., Hucka, M., Howell, F., Cornelis, H., Shankar, K., and Beeman, D. (2001). Towards NeuroML: model description methods for collaborative modelling in neuroscience. Philos Trans R Soc Lond B Biol Sci *356*, 1209-1228.

Golding, N. L., Mickus, T. J., Katz, Y., Kath, W. L., and Spruston, N. (2005). Factors mediating powerful voltage attenuation along CA1 pyramidal neuron dendrites. J Physiol *568*, 69-82.

Hanson, J. E., Smith, Y., and Jaeger, D. (2004). Sodium channels and dendritic spike initiation at excitatory synapses in globus pallidus neurons. J Neurosci *24*, 329-340.

Harvey, R. J., and Napper, R. M. (1991). Quantitative studies on the mammalian cerebellum. Prog Neurobiol *36*, 437-463.

Helmchen, F., and Denk, W. (2002). New developments in multiphoton microscopy. Curr Opin Neurobiol *12*, 593-601.

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

Hines, M. L., and Carnevale, N. T. (2000). Expanding NEURON's repertoire of mechanisms with NMODL. Neural Comput *12*, 995-1007.

Hines, M. L., Morse, T., Migliore, M., Carnevale, N. T., and Shepherd, G. M. (2004). ModelDB: A Database to Support Computational Neuroscience. J Comput Neurosci *17*, 7-11.

Howell, F. W., Dyhrfjeld-Johnsen, J., Maex, R., Goddard, N., and De Schutter, E. (2000). A large Scale model of the cerebellar cortex using PGENESIS. Neurocomputing *32-33*, 1041-1046.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., *et al.* (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics *19*, 524-531.

Ito, M. (1984). The cerebellum and neural control (New York, Raven Press).

Jacoby, S., Sims, R. E., and Hartell, N. A. (2001). Nitric oxide is required for the induction and heterosynaptic spread of long-term potentiation in rat cerebellar slices. J Physiol *535*, 825-839.

Jarsky, T., Roxin, A., Kath, W. L., and Spruston, N. (2005). Conditional dendritic spike propagation following distal synaptic activation of hippocampal CA1 pyramidal neurons. Nat Neurosci *8*, 1667-1676.

Kitano, H. (2002). Systems biology: a brief overview. Science *295*, 1662-1664.

Kunec, S., Hasselmo, M. E., and Kopell, N. (2005). Encoding and retrieval in the CA3 region of the hippocampus: a model of theta-phase separation. J Neurophysiol *94*, 70-82.

Lloyd, C. M., Halstead, M. D., and Nielsen, P. F. (2004). CellML: its future, present and past. Prog Biophys Mol Biol *85*, 433-450.

Lubke, J., Roth, A., Feldmeyer, D., and Sakmann, B. (2003). Morphometric analysis of the columnar innervation domain of neurons connecting layer 4 and layer 2/3 of juvenile rat barrel cortex. Cereb Cortex *13*, 1051-1063.

Maex, R., and De Schutter, E. D. (1998). Synchronization of golgi and granule cell firing in a detailed network model of the cerebellar granule cell layer. J Neurophysiol *80*, 2521-2537.

Mainen, Z. F., Joerges, J., Huguenard, J. R., and Sejnowski, T. J. (1995). A model of spike initiation in neocortical pyramidal neurons. Neuron *15*, 1427-1439.

Mainen, Z. F., and Sejnowski, T. J. (1996). Influence of dendritic structure on firing pattern in model neocortical neurons. Nature *382*, 363-366.

Markram, H. (2006). The Blue Brain Project. Nat Rev Neurosci *7*, 153-160.

Markram, H., Lubke, J., Frotscher, M., Roth, A., and Sakmann, B. (1997). Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex. J Physiol *500 (Pt 2)*, 409-440.

Medina, J. F., and Mauk, M. D. (2000). Computer simulation of cerebellar information processing. Nat Neurosci *3 Suppl*, 1205-1211.

Mel, B. W. (1993). Synaptic integration in an excitable dendritic tree. J Neurophysiol *70*, 1086-1101.

Migliore, M., Cannia, C., Lytton, W., Markram, H., and Hines, M. (2006). Parallel network simulations with NEURON. Journal of Computational Neuroscience.

Migliore, M., Cook, E. P., Jaffe, D. B., Turner, D. A., and Johnston, D. (1995). Computer simulations of morphologically reconstructed CA3 hippocampal neurons. J Neurophysiol *73*, 1157-1168.

Mitchell, S. J., and Silver, R. A. (2000). Glutamate spillover suppresses inhibition by activating presynaptic mGluRs. Nature *404*, 498-502.

Nicolelis, M. A., and Ribeiro, S. (2002). Multielectrode recordings: the next steps. Curr Opin Neurobiol *12*, 602-606.

Ohki, K., Chung, S., Ch'ng, Y. H., Kara, P., and Reid, R. C. (2005). Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. Nature *433*, 597-603.

Poirazi, P., Brannon, T., and Mel, B. W. (2003). Pyramidal neuron as two-layer neural network. Neuron *37*, 989-999.

Pouille, F., and Scanziani, M. (2004). Routing of spike series by dynamic circuits in the hippocampus. Nature *429*, 717-723.

Rall, W., Burke, R. E., Smith, T. G., Nelson, P. G., and Frank, K. (1967). Dendritic location of synapses and possible mechanisms for the monosynaptic EPSP in motoneurons. J Neurophysiol *30*, 1169-1193.

Rapp, M., Yarom, Y., and Segev, I. (1996). Modeling back propagating action potential in weakly excitable dendrites of neocortical pyramidal cells. Proc Natl Acad Sci U S A *93*, 11985-11990.

Safo, P. K., Cravatt, B. F., and Regehr, W. G. (2006). Retrograde endocannabinoid signaling in the cerebellar cortex. Cerebellum *5*, 134-145.

Santhakumar, V., Aradi, I., and Soltesz, I. (2005). Role of mossy fiber sprouting and mossy cell loss in hyperexcitability: a network model of the dentate gyrus incorporating cell types and axonal topography. J Neurophysiol *93*, 437-453.

Schaefer, A. T., Larkum, M. E., Sakmann, B., and Roth, A. (2003). Coincidence detection in pyramidal neurons is tuned by their dendritic branching pattern. J Neurophysiol *89*, 3143-3154.

Schweighofer, N., and Ferriol, G. (2000). Diffusion of nitric oxide can facilitate cerebellar learning: A simulation study. Proc Natl Acad Sci U S A *97*, 10661-10665.

Segev, I., and London, M. (2000). Untangling dendrites with quantitative models. Science *290*, 744-750.

Shepherd, G. M. (1998). The synaptic organization of the brain, Fourth edn (New York, Oxford University Press).

Somogyi, P., Tamas, G., Lujan, R., and Buhl, E. H. (1998). Salient features of synaptic organisation in the cerebral cortex. Brain Res Brain Res Rev *26*, 113-135.

Song, S., Sjostrom, P. J., Reigl, M., Nelson, S., and Chklovskii, D. B. (2005). Highly nonrandom features of synaptic connectivity in local cortical circuits. PLoS Biol *3*, e68.

Sporns, O., and Kotter, R. (2004). Motifs in brain networks. PLoS Biol *2*, e369.

Stosiek, C., Garaschuk, O., Holthoff, K., and Konnerth, A. (2003). In vivo two-photon calcium imaging of neuronal networks. Proc Natl Acad Sci U S A *100*, 7319-7324.

Sultan, F., and Bower, J. M. (1998). Quantitative Golgi study of the rat cerebellar molecular layer interneurons using principal component analysis. Journal of Comparative Neurology *393*, 353-373.

Thomson, A. M., West, D. C., Wang, Y., and Bannister, A. P. (2002). Synaptic connections and small circuits involving excitatory and inhibitory neurons in layers 2-5 of adult rat and cat neocortex: triple intracellular recordings and biocytin labelling in vitro. Cereb Cortex *12*, 936-953.

Traub, R. D., Contreras, D., Cunningham, M. O., Murray, H., LeBeau, F. E., Roopun, A., Bibbig, A., Wilent, W. B., Higley, M. J., and Whittington, M. A. (2005). Single-column thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts. J Neurophysiol *93*, 2194-2232.

van Ooyen, A., Duijnhouwer, J., Remme, M. W., and van Pelt, J. (2002). The effect of dendritic topology on firing patterns in model neurons. Network *13*, 311-325.

Vetter, P., Roth, A., and Hausser, M. (2001). Propagation of action potentials in dendrites depends on dendritic morphology. J Neurophysiol *85*, 926-937.

Vos, B. P., Maex, R., Volny-Luraghi, A., and De Schutter, E. (1999). Parallel fibers synchronize spontaneous activity in cerebellar Golgi cells. J Neurosci *19*, RC6.

Watts, D. J., and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. Nature *393*, 440-442.

Wilson, R. I., and Nicoll, R. A. (2002). Endocannabinoid signaling in the brain. Science *296*, 678-682.

Yoshimura, Y., and Callaway, E. M. (2005). Fine-scale specificity of cortical networks depends on inhibitory cell type and connectivity. Nat Neurosci *8*, 1552-1559.

Figure 1

Gleeson et al. Figure 1

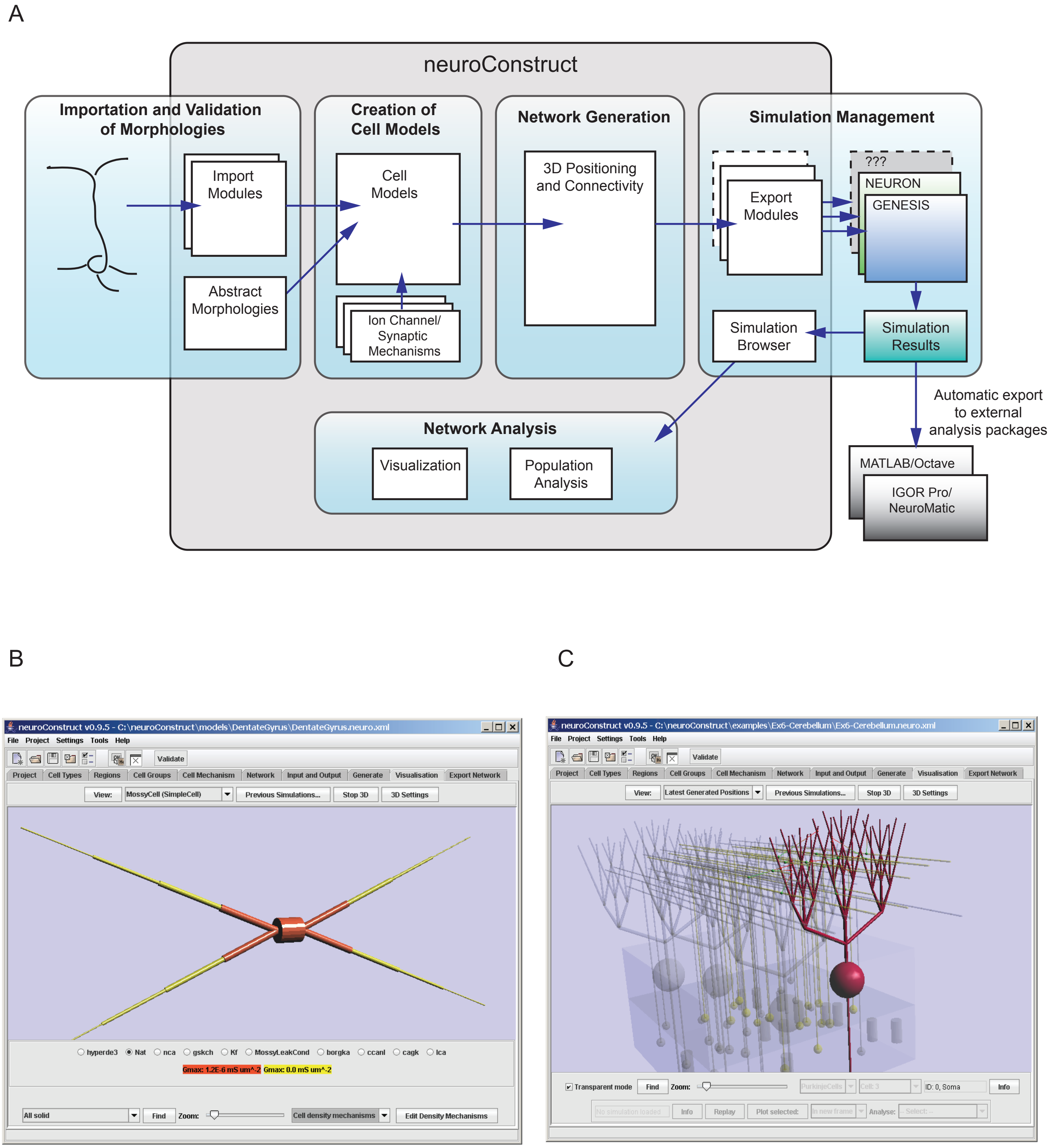

A
neuroConstruct
Importation and Validation of Morphologies
Import Modules
Abstract Morphologies
Creation of Cell Models
Cell Models
Ion Channel/ Synaptic Mechanisms
Network Generation
3D Positioning and Connectivity
Simulation Management
Export Modules
???
NEURON
GENESIS
Simulation Browser
Simulation Results
Network Analysis
Visualization
Population Analysis
Automatic export to external analysis packages
MATLAB/Octave
IGOR Pro/ NeuroMatic
B
neuroConstruct v0.9.5 - C:\neuroConstruct\models\DentateGyrus\DentateGyrus.neuro.xml
File Project Settings Tools Help
Validate
Project Cell Types Regions Cell Groups Cell Mechanism Network Input and Output Generate Visualisation Export Network
View: MossyCell (SimpleCell)
Previous Simulations... Stop 3D 3D Settings
○ hyperde3 ● Nat ○ nca ○ gskch ○ Kf ○ MossyLeakCond ○ borgka ○ ccanl ○ cagk ○ lca
Gmax: 1.2E-6 mS um^-2 Gmax: 0.0 mS um^-2
All solid
Find Zoom:
Cell density mechanisms
Edit Density Mechanisms
C
neuroConstruct v0.9.5 - C:\neuroConstruct\examples\Ex6-Cerebellum\Ex6-Cerebellum.neuro.xml
File Project Settings Tools Help
Validate
Project Cell Types Regions Cell Groups Cell Mechanism Network Input and Output Generate Visualisation Export Network
View: Latest Generated Positions
Previous Simulations... Stop 3D 3D Settings
☑ Transparent mode
Find Zoom:
PurkinjeCells
Cell: 3
ID: 0, Soma
Info
No simulation loaded
Info Replay Plot selected:
In new frame
Analyse: Select: -

Figure 2

Gleeson et al. Figure 2

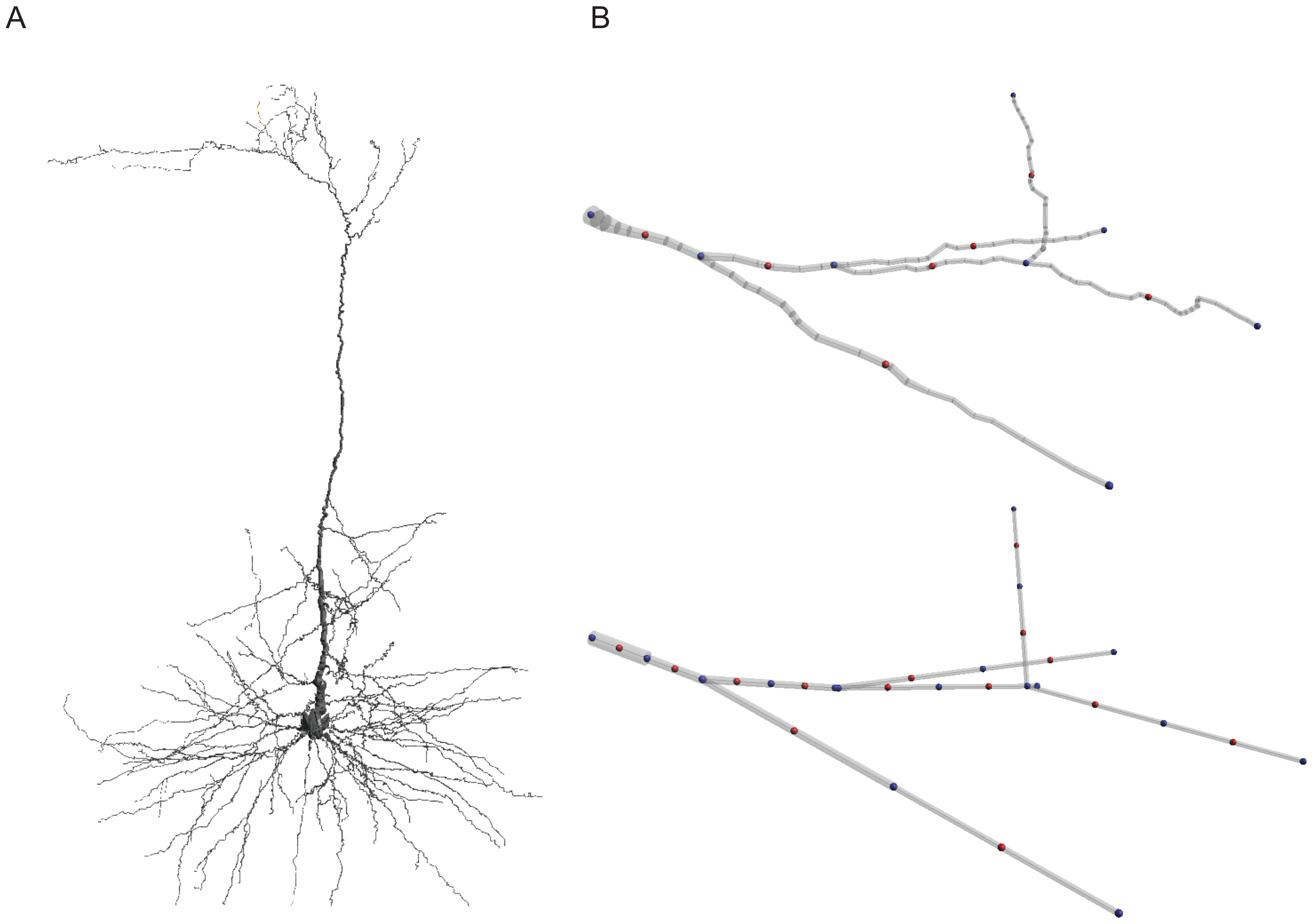
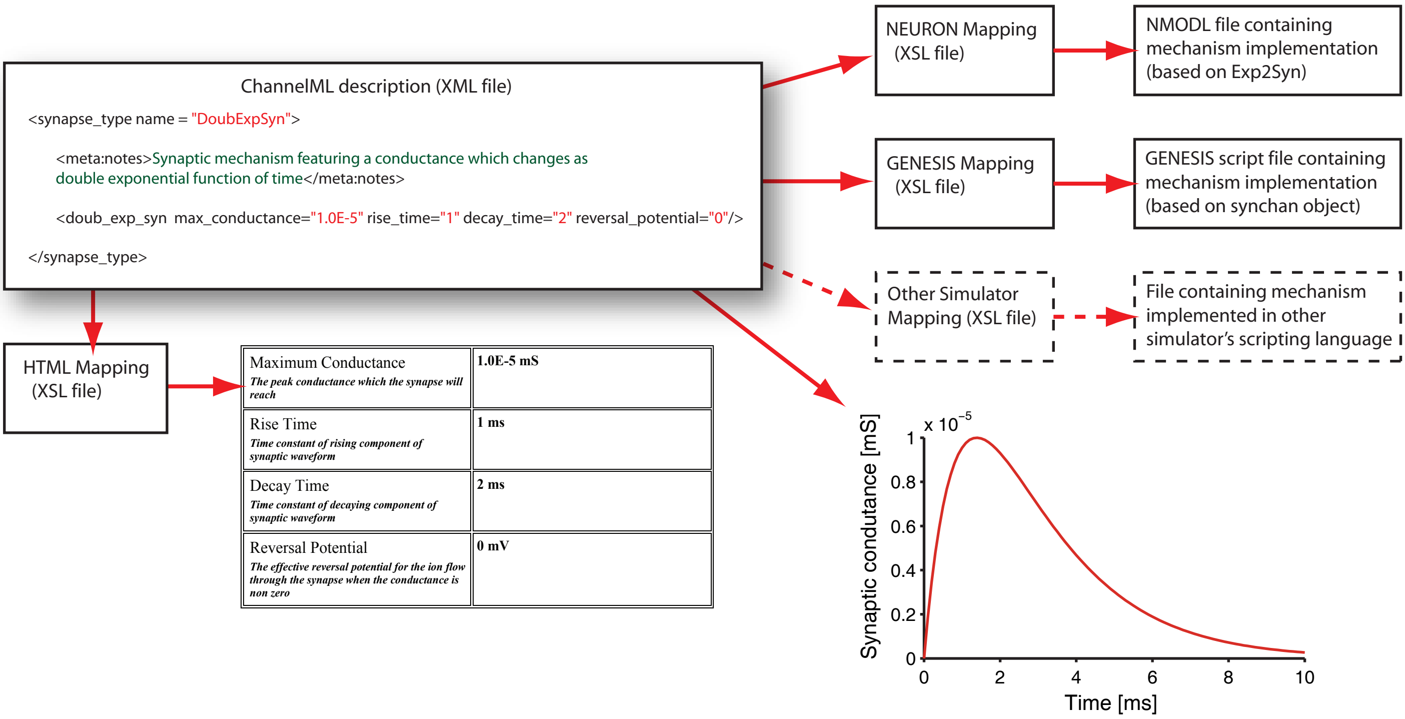


Figure 3

Gleeson et al. Figure 3



ChannelML description (XML file)

```
<synapse_type name = "DoubExpSyn">

    <meta:notes>Synaptic mechanism featuring a conductance which changes as
    double exponential function of time</meta:notes>

    <doub_exp_syn  max_conductance="1.0E-5" rise_time="1" decay_time="2" reversal_potential="0"/>

</synapse_type>
```

Maximum Conductance *The peak conductance which the synapse will reach*	**1.0E-5 mS**
Rise Time *Time constant of rising component of synaptic waveform*	**1 ms**
Decay Time *Time constant of decaying component of synaptic waveform*	**2 ms**
Reversal Potential *The effective reversal potential for the ion flow through the synapse when the conductance is non zero*	**0 mV**

NEURON Mapping (XSL file)
NMODL file containing mechanism implementation (based on Exp2Syn)
GENESIS Mapping (XSL file)
GENESIS script file containing mechanism implementation (based on synchan object)
Other Simulator Mapping (XSL file)
File containing mechanism implemented in other simulator's scripting language
HTML Mapping (XSL file)
Synaptic condutance [mS]
x 10^-5
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
Time [ms]

Figure 4

Gleeson et al.

Figure 4

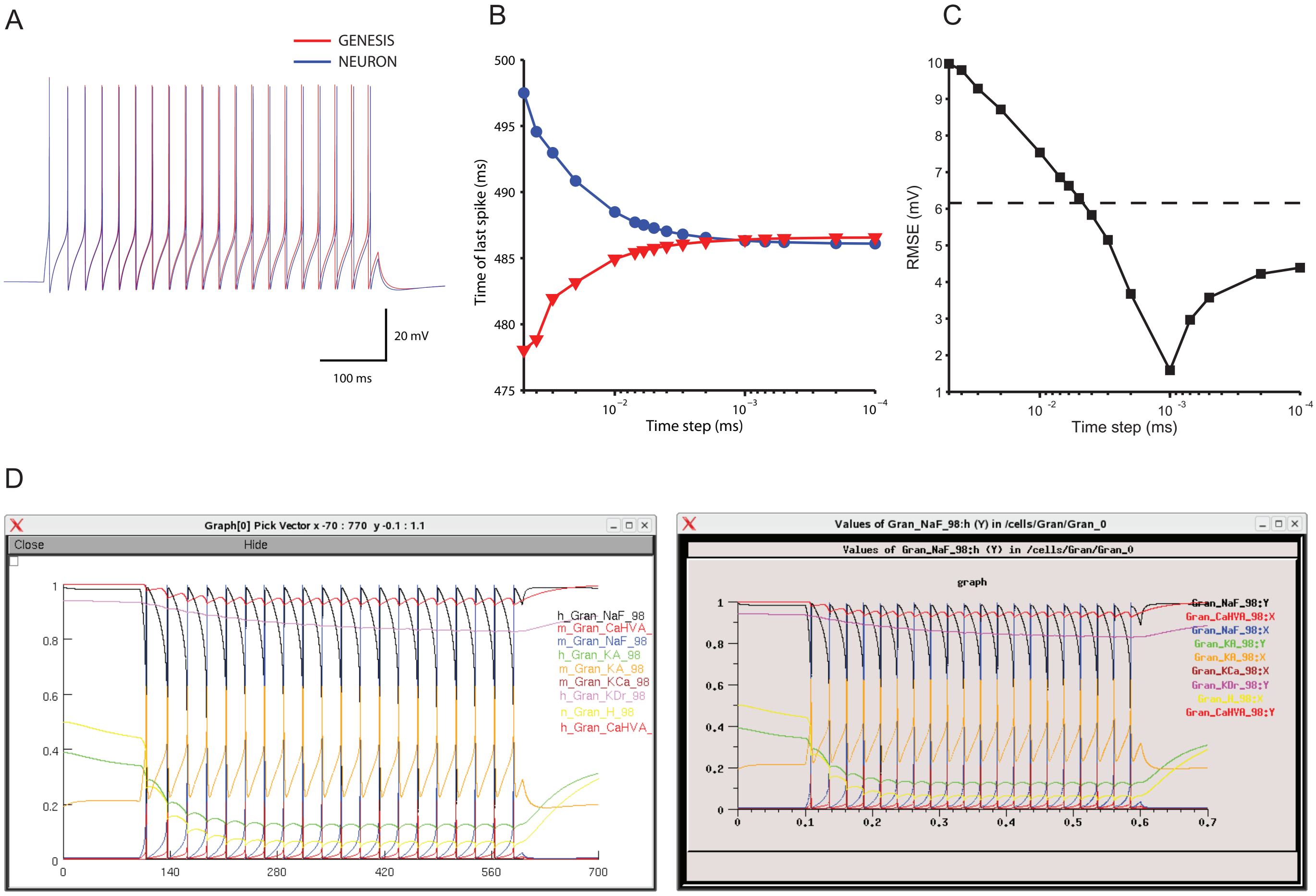


Figure 5

Gleeson et al. Figure 5

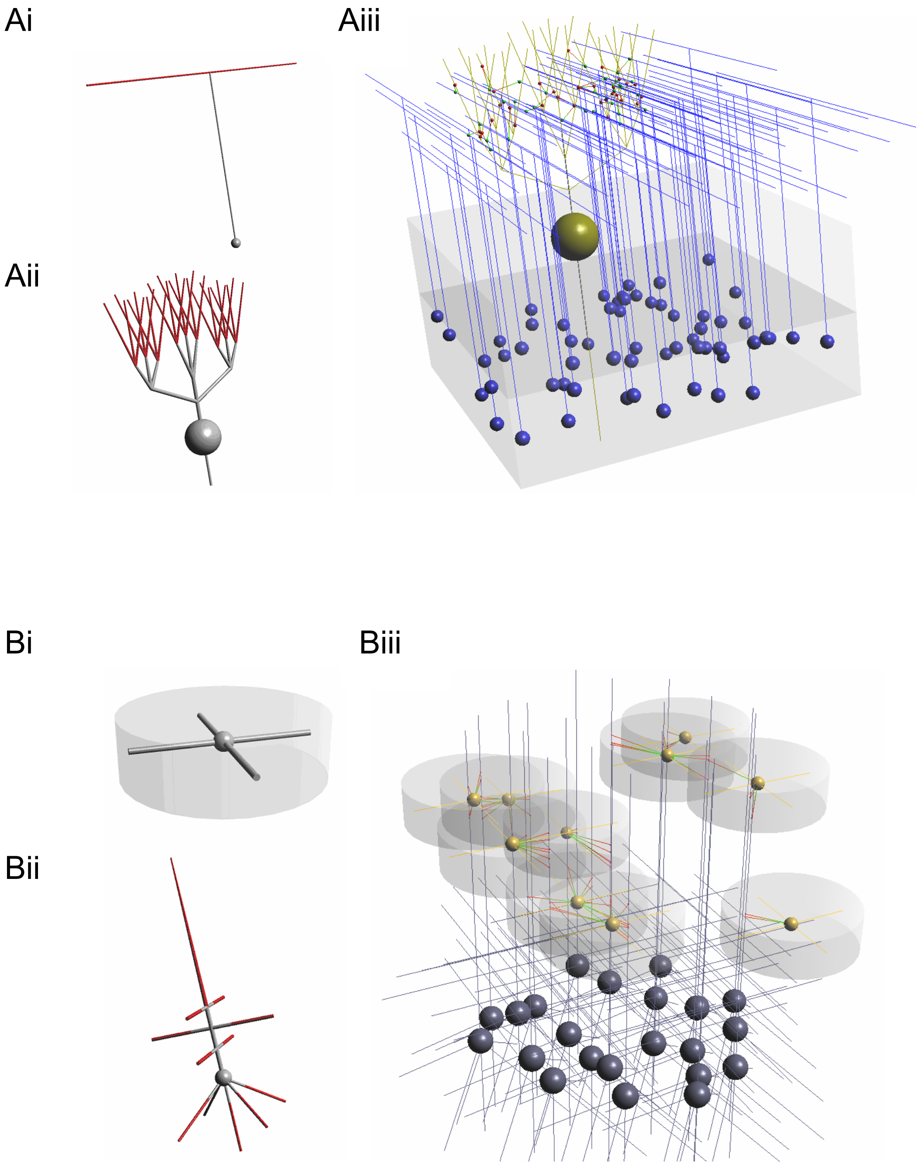
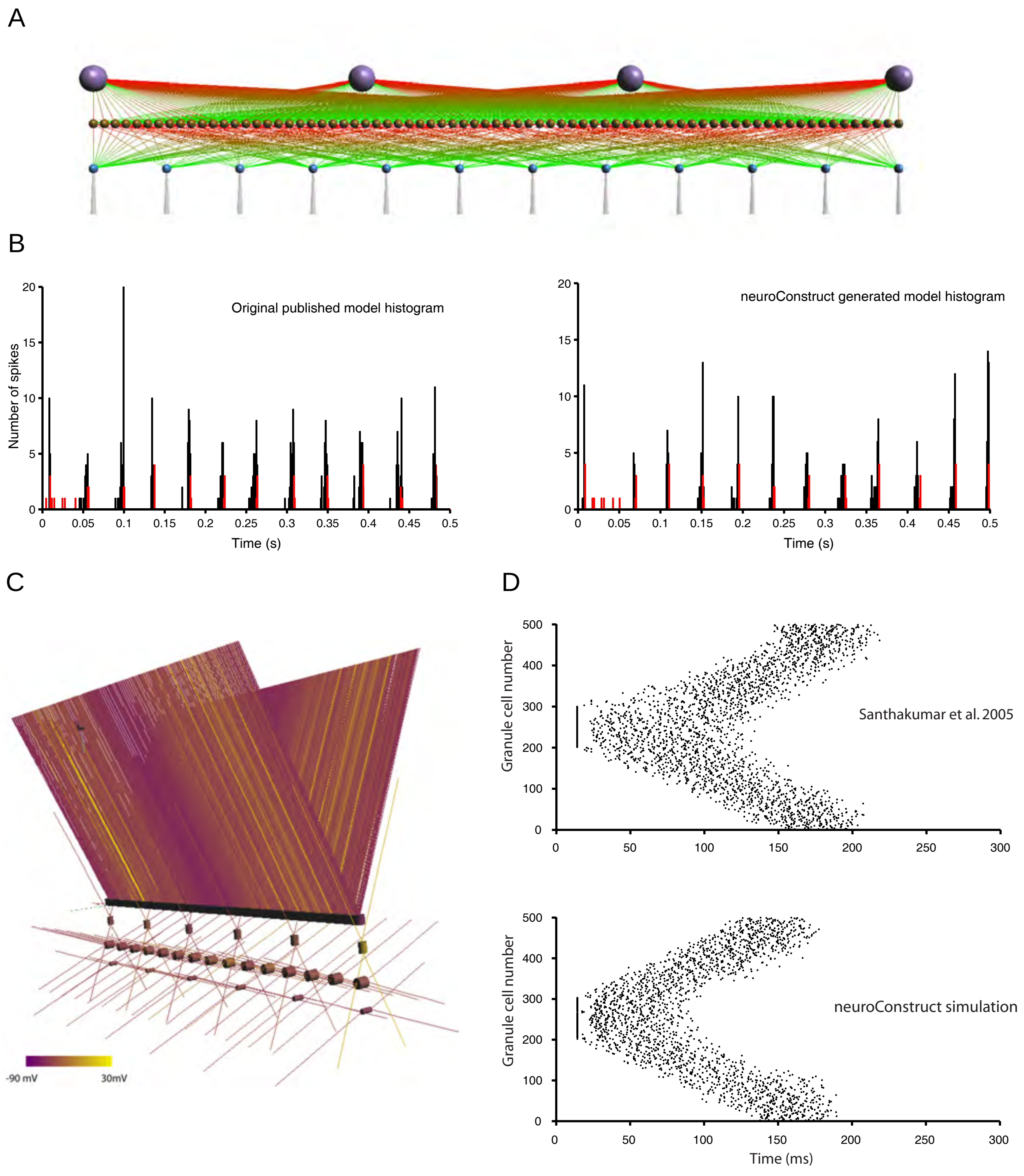
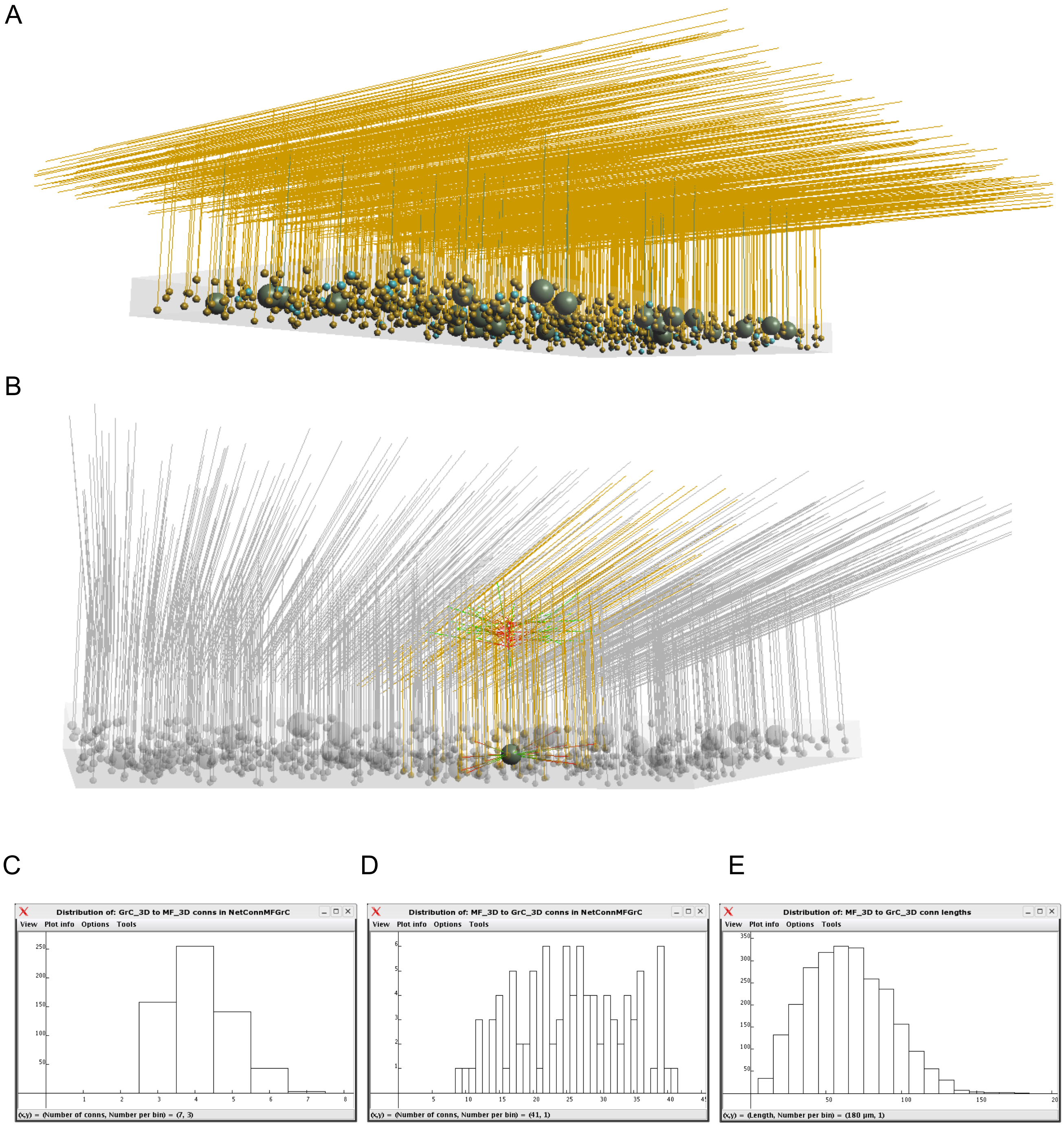


Figure 6

Gleeson et al. Figure 6



Gleeson et al. Figure 7



Distribution of: GrC_3D to MF_3D conns in NetConnMFGrC

(x,y) = (Number of conns, Number per bin) = (7, 3)

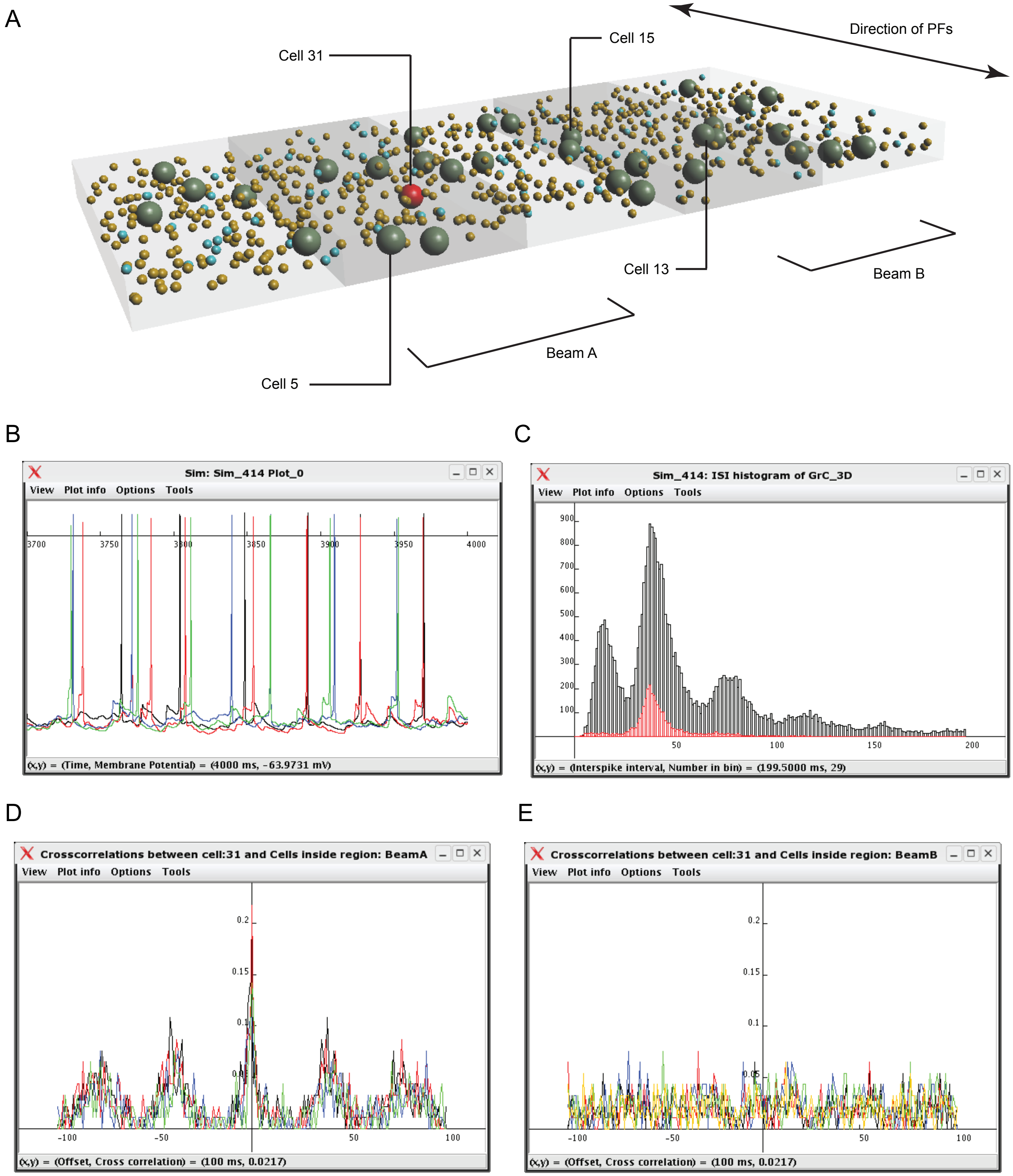
Distribution of: MF_3D to GrC_3D conns in NetConnMFGrC

(x,y) = (Number of conns, Number per bin) = (41, 1)

Distribution of: MF_3D to GrC_3D conn lengths

(x,y) = (Length, Number per bin) = (180 µm, 1)

Figure 8



(x,y) = (Time, Membrane Potential) = (4000 ms, -63.9731 mV)

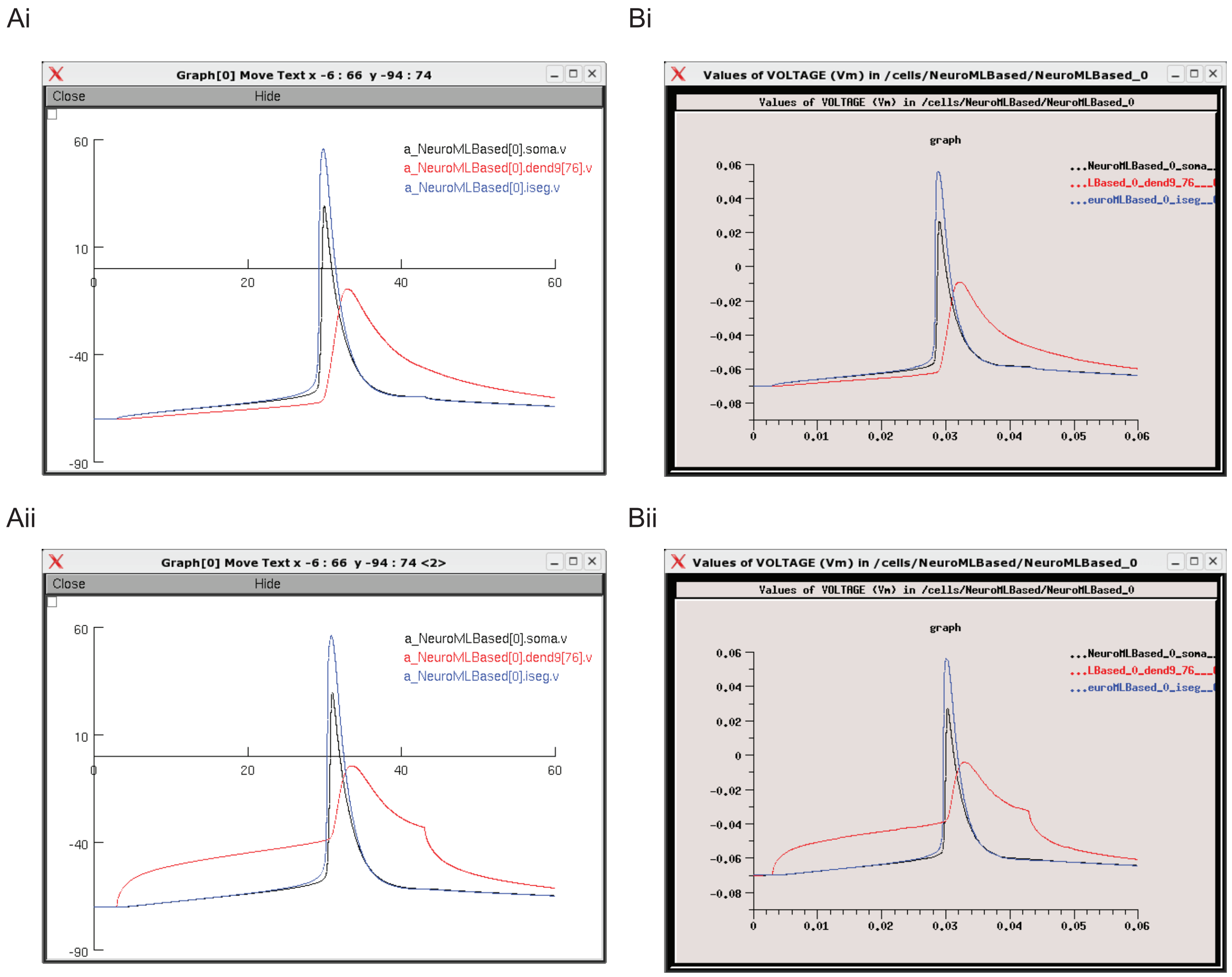
(x,y) = (Interspike interval, Number in bin) = (199.5000 ms, 29)

(x,y) = (Offset, Cross correlation) = (100 ms, 0.0217)

(x,y) = (Offset, Cross correlation) = (100 ms, 0.0217)

Supplementary Figure 1

Gleeson et al. Supplementary Figure 1



Comparison of the properties of a detailed model of a layer 5 pyramidal cell implemented in NeuroML and run on NEURON and GENESIS simulators.

A model of a rat layer 5 pyramidal cell from Mainen et al. (1995; illustrated Figure 2A) was converted from the original NEURON script (obtained from http://senselab.med.yale.edu/senselab/modeldb/ShowModel.asp?model=8210) to *neuroConstruct's* NeuroML-based internal morphological representation with channels that were specified in ChannelML. Minor changes were made to fix zero length sections, etc. This allowed the model to be run on both NEURON and GENESIS simulators.

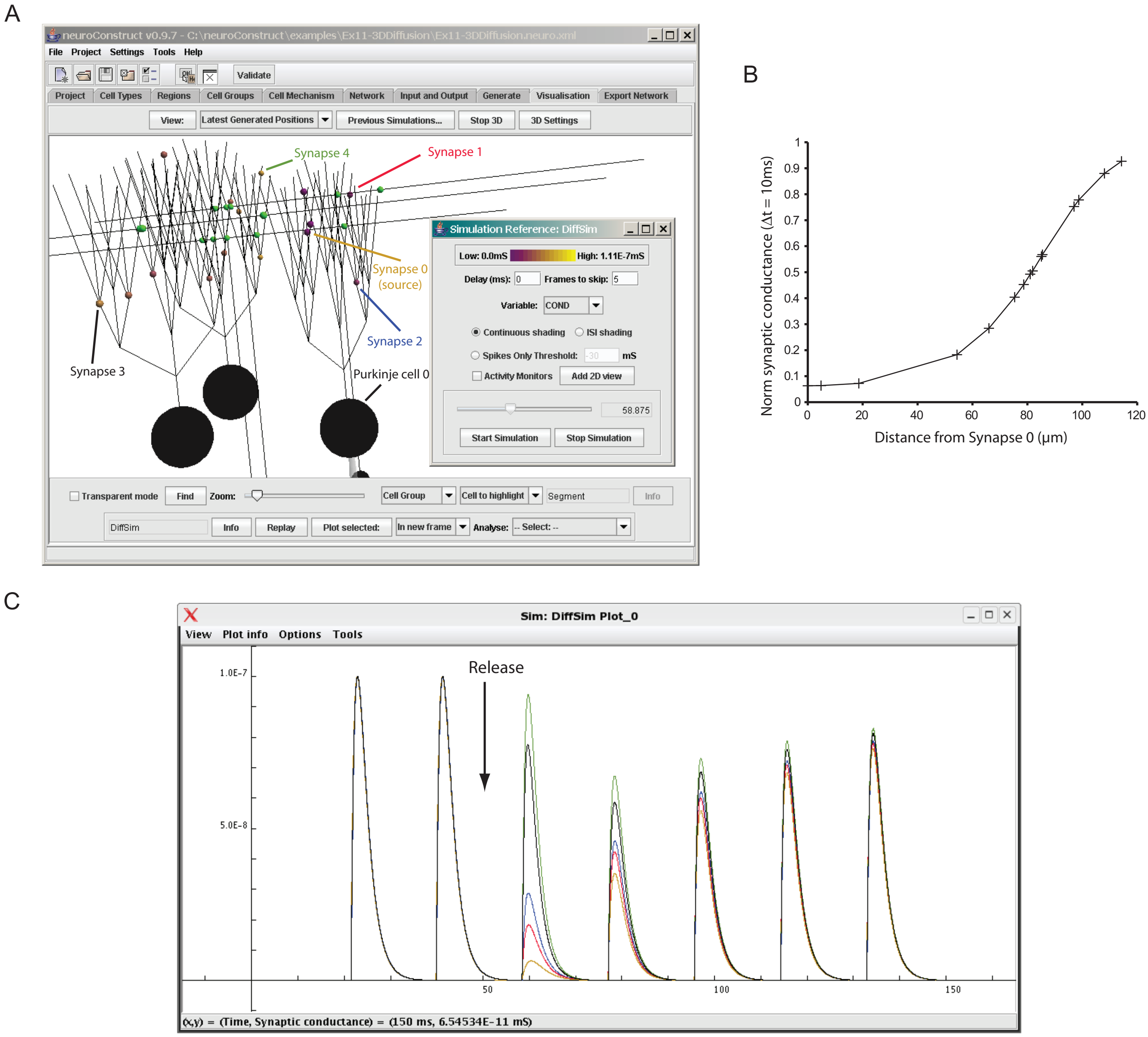
A) Voltage traces from the layer 5 pyramidal cell model implemented in *neuroConstruct* and run on the NEURON simulator. Ai) The cell response to a somatic current step at the soma (black trace), initial segment (blue) and a point along the main apical dendrite at 416 μm from the soma (red). Axes are voltage (mV) and time (ms). Aii) shows the responses, at the same locations, to stimulation at the dendritic location. These results reproduced those in Figure 3A in Mainen et al. (1995).

B) The same layer 5 pyramidal cell model implemented in *neuroConstruct* and run on the GENESIS simulator. The colored traces correspond to those described in (A). Apart from a minor difference in the timing of the somatic spike (<1 ms) the properties of the model closely reproduce those obtained in with NEURON. Axes are voltage (V) and time (s).

We also tested a version of this model when the original morphology of 5726 segments was recompartmentalized in *neuroConstruct* to 800 segments, while maintaining the overall segment length, total surface area and total axial resistance (Experimental Procedures). Simulation in GENESIS using the recompartmentalized morphology produced similar results to those obtained in B (with a small temporal shift ~1 ms) and speeded up the simulation by a factor of 10.

Supplementary Figure 2

Gleeson et al. Supplementary Figure 2


A
neuroConstruct v0.9.7 - C:\neuroConstruct\examples\Ex11-3DDiffusion\Ex11-3DDiffusion.neuro.xml
File Project Settings Tools Help
Validate
Project Cell Types Regions Cell Groups Cell Mechanism Network Input and Output Generate Visualisation Export Network
Previous Simulations... Stop 3D 3D Settings
Synapse 4
Synapse 1
Synapse 0 (source)
Synapse 2
Synapse 3
Purkinje cell 0
Simulation Reference: DiffSim
Continuous shading
ISI shading
Spikes Only Threshold:
Activity Monitors
Add 2D view
Start Simulation
Stop Simulation
Transparent mode
Find
Zoom:
Info
Replay
Plot selected:
B
Norm synaptic conductance (Δt = 10ms)
Distance from Synapse 0 (μm)
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 20 40 60 80 100 120
C
Sim: DiffSim Plot_0
View Plot info Options Tools
Release
1.0E-7
5.0E-8
50 100 150

View: Latest Generated Positions

Low: 0.0mS High: 1.11E-7mS

Delay (ms): 0 Frames to skip: 5

Variable: COND

● Continuous shading ○ ISI shading

○ Spikes Only Threshold: -30 mS

□ Activity Monitors

58.875

□ Transparent mode

Cell Group | Cell to highlight | Segment | Info

DiffSim | Info | Replay | Plot selected: | In new frame | Analyse: -- Select: --

(x,y) = (Time, Synaptic conductance) = (150 ms, 6.54534E-11 mS)

Example of a simple diffusion mechanism implemented in a 3D network model

To illustrate how 3D information in *neuroConstruct* network models can be used we have created a simple 3D diffusion model of substance that transiently inhibits synapses on Purkinje cells (PCs). This was implemented by writing a new synaptic mechanism in NEURON NMODL script and inserting it into the *neuroConstruct* model via the GUI interface. In the simulation shown above a diffusible substance was released from an individual synapse. All other synapses registered this event (via the setpointer call in NEURON). The concentration of the diffusible substance was calculated at each synaptic location with an analytical solution for a point source in an infinite medium (Equation 3.5; Crank (1975) The Mathematics of Diffusion). The inhibitory effect of the diffusing substance on the synaptic conductance was implemented with a normalized scale factor that was calculated using a Hill expression for a first order reaction.

A) A *neuroConstruct* screenshot of three parallel fibres (PFs; granule cell axons) and three Purkinje cells (PCs) arranged in 3D. Green spheres show presynaptic locations and yellow to purple colour coding shows the conductance values of the postsynaptic locations as indicated in the simulation replay panel. For illustrative purposes we allowed PFs to make multiple synaptic contacts on an individual PC in this model.
B) A plot of the peak synaptic conductance for all 15 synapses at 10 ms after release (60 ms on panel C), normalized to pre release peak conductance. At this early time the inhibitory substance had not diffused very far and distant synapses are only slightly affected.
C) A plot of PF-PC synaptic conductances as a function of time for the five highlighted synapses shown in A during regular stimulation of the granule cells. At 50 ms, a pulse input was applied to Purkinje Cell 0 causing it to fire. This triggered the release of an inhibitory substance from Synapse 0, which had been active within a short time interval. By the subsequent synaptic activation (10 ms later) the synaptic conductances were inhibited with the greatest effect at the closest synapses. At later times the conductance amplitudes converged and then recovered back to control.

This simple, albeit rather unphysiological model, demonstrates that diffusion mechanisms can be implemented in 3D network models created by *neuroConstruct*, but at present this requires the insertion of custom NEURON code.

Installation instructions

Click here to download Supplemental Movies and Spreadsheets: Install.pdf

neuroConstruct_install_win

Click here to download Supplemental Movies and Spreadsheets: neuroConstruct_windows_0_9_7.exe

neuroConstruct_install_MacOS

Click here to download Supplemental Movies and Spreadsheets: neuroConstruct_macos_0_9_7.dmg

neuroConstruct_install_Linux

Click here to download Supplemental Movies and Spreadsheets: neuroConstruct_unix_0_9_7.sh