Parallel simulations of a 3D cerebellar network created with neuroConstruct

Padraig Gleeson¹, Volker Steuber¹,², R. Angus Silver³

¹Department of Physiology, University College London, UK
²Science and Technology Research Institute, University of Hertfordshire, UK

The various configurations of the parallel computing environment in which network simulations can be run are separate from any specific Simulation Configuration. These Parallel Configurations can include: a single processor on the local machine, a number of local processors, or a set of hosts each of which can have multiple processors. At the time of generation, the cells and connections for a particular Simulation Configuration are created first, and then the cells are assigned to the computing nodes as defined in the chosen Parallel Configuration.

To create an executable simulation, a NetworkML file specifying the network structure plus distribution across the nodes is created, along with a generic NetworkML parser written in Python (appropriate for any NetworkML file and any simulator supporting Python) and files specific for the simulator, to create the cells and manage the simulation. Currently Parallel NEURON is supported, but the scheme could be extended for other simulations, potentially accessing them through a standard interface such as PynML.

Cerebellar Granule Cell Layer Model

The ability to execute simulations in parallel environments allows larger scale simulations to be generated. The system we are currently focusing on is the granule cell (GrC) layer of the cerebellum. To this end we are collaborating with Zoltan Nusser’s labs to obtain detailed anatomical data on the densities of GrCs, mossy fiber terminals and Golgi cells in the GrC layer of P30 rats, to age match the electrophysiological data collected in the Silver lab.

The latest version of neuroConstruct has allowed us to extend the initial 3D GrC layer model from Gleeson et al 2007 to reflect a more realistic anatomy and to scale up the size of the simulation. We have constructed a model of a 1mm by 1mm patch of GrC layer, a volume sufficiently large to encompass a number of distinct receptive fields within the fractured somatotopic map (Bower and Kassel, 1990). A smaller scale version of the network with the same cell densities and connectivity can be used when multiple simulation runs are needed to investigate the influence of anatomy and synaptic plasticity on signal processing in this brain region.

Conclusions

- We have extended neuroConstruct to automatically generate code for running simulations on multiple processors using the Python scripting version of Parallel NEURON.
- This allows the generation of networks of hundreds of thousands of cells.
- Simulations can be created on a scale much closer to real biological networks
- neuroConstruct can be freely downloaded from http://www.neuroConstruct.org.

References


Two versions of the cerebellar granule cell layer model incorporating realistic cell densities and connectivity. Left: scale of 10µm containing 2380 cells. Right: from square by 10µm thick patch containing over a quarter of a million cells. Granule cells are in yellow, mossy fiber terminals in blue, Golgi cells in red.

Parallel network simulations

A neuroConstruct project can have multiple Cell Types, the structures of which are specified in a simulator independent format, closely linked to MorphML. Cell Groups consist of a Cell Type, the 3D Region in which they are placed and a specification of the packing pattern to use. Simulation Configurations are sets of Cell Groups, together with Network Connections, electrical inputs and lists of variables to plot and/or save during the simulation. A project can have many of these, each designed to illustrate a different aspect of the behaviour of the cells in the network, e.g. activity of single cells, small networks of a subset of Cell Groups, large scale network activity.

References


