TITLE :

Simulating Multiple Interacting Neural Populations using Population Density Techniques

ORGANISERS :

Hugh Osborne (University of Leeds, UK) - sc16ho@leeds.ac.uk
Marc de Kamps (University of Leeds, UK) - m.dekamps@leeds.ac.uk
Lukas Deutz (University of Leeds, UK) - scld@leeds.ac.uk

FORMAT :

Introductory Full Day Course

DESCRIPTION :

Neural behaviour at the largest of scales is often modelled using mean rate based techniques which use a small number of variables and fitted parameters to capture the dynamics of a population of neurons. This is both time efficient and an appropriate level of complexity to answer many questions about brain dynamics. However, sometimes it is desirable to be able to relate the population behaviour to the behaviour of its constituent neurons and most rate based techniques do not do this rigorously. Population Density Techniques [1-3] are a rigorous method for taking an individual point neuron model and simulating the dynamics of a population, without the need to simulate individual cells. These methods have been shown to replicate firing rates accurately, compared to direct spike-based simulations, even for small populations [1,2]. In this tutorial we start with presenting the theory of PDTs, their strengths and weaknesses.

Then we present MIIND [4], a neural simulation platform designed for modelling the interactions between multiple populations of neurons. Unlike other PDT systems like DIPDE [5], MIIND uses a two dimensional geometric population density technique [6,7]. We will introduce this technique and guide participants through setting up individual neural populations. To solidify their understanding, they may then familiarise themselves with the simulation and analysis workflow and produce movies of each population as it develops over time in the neuron model's state space. This foundation will pave the way towards modelling a large scale network of populations using MIIND's XML style language. Such simulations can run on a single PC for smaller networks and for larger ones on a GPGPU device or a cluster. We will make an Ubuntu Docker available so that participants can follow the demonstration on their own laptop.

If you have a simulation you wish to try to adapt to a population density approach, we would be happy to give advice.

Tutorial Resources

MIIND source code and documentation : https://github.com/dekamps/miind
MIIND prepackaged download and written tutorial : http://miind.sourceforge.net/
MIIND Docker Image (dockerhub) : hughosborne/miind

Example population movie :
https://drive.google.com/open?id=1CxDcXZWgBZysZDNMQVSzL_02PvsxetY4

[1] Omurtag, A., Knight, B.W. and Sirovich, L., 2000. On the simulation of large populations of neurons. *Journal of computational neuroscience*, *8*(1), pp.51-63.
[2] de Kamps M (2003) A simple and stable numerical solution for the population density equation. *Neural Computation*, **15**(9), pp. 2129-2146.
[3] Baladron J, Fasoli D, Faugeras O, Touboul J. Mean-field description and propagation of chaos in networks of Hodgkin-Huxley and FitzHugh-Nagumo neurons. The Journal of Mathematical Neuroscience. 2012;2(1):10.
[4] de Kamps M; Baier V; Drever J; Dietz M; Mosenlechner L; van der Velde F (2008) The state of MIIND. *NEURAL NETWORKS*, **21**(8), pp. 1164-1181.
[5] Website: © 2015 Allen Institute for Brain Science. DiPDE Simulator [Internet]. Available from: https://github.com/AllenInstitute/dipde.
[6]de Kamps, M., 2013. A generic approach to solving jump diffusion equations with applications to neural populations. *arXiv preprint arXiv:1309.1654*.
[7] de Kamps, M., Elle Lepperød, M. and Lai, Y. M. 2019 Computational Geometry for Modeling Neural Populations: from Visualization to Simulation 2019, *Accepted for publication by Plos CB,* https://www.biorxiv.org/content/10.1101/275412v5