

CNS*2024 Natal

B Break **K** Keynote **R** reception/party **G** Registration **T** Tutorial

JULY 20 • SATURDAY

8:00am – 8:00am **G** **Registration**

9:00am –
10:15am

T T01: Building mechanistic multiscale models using NEURON and NetPyNE to study brain function and disease

Cedro I

Speakers: Valery Bragin, Salvador Dura Bernal, William W Lytton, Robert McDougal, Adam J.H. Newton

Understanding the brain requires studying its multiscale interactions, from molecules to cells to circuits and networks. Although vast experimental datasets are being generated across scales and modalities, integrating and interpreting this data remains a daunting challenge. This tutorial will highlight recent advances in mechanistic multiscale modeling and how it offers an unparalleled approach to integrate these data and provide insights into brain function and disease. Multiscale models facilitate the interpretation of experimental findings across different brain regions, brain scales (molecular, cellular, circuit, system), brain function (sensory perception, motor behavior, learning, etc), recording/imaging modalities (intracellular voltage, LFP, EEG, fMRI, etc) and disease/disorders (e.g., schizophrenia, epilepsy, ischemia, Parkinson's, etc). As such, it has a broad appeal to experimental, clinical, and computational neuroscientists, students, and educators.

This tutorial will introduce multiscale modeling using two NIH-funded tools: the NEURON 9.0 simulator (<https://neuron.yale.edu/neuron/>), including the Reaction-Diffusion (RxD) module and the NetPyNE tool (<http://netpyne.org>). The tutorial will combine background, examples, and hands-on exercise covering the implementation of models at four key scales: (1) intracellular dynamics (e.g., calcium buffering, protein interactions), (2) single neuron electrophysiology (e.g., action potential propagation), (3) neurons in extracellular space (e.g., spreading depression), and (4) neuronal circuits, including dynamics such as oscillations and simulation of recordings such as local field potentials (LFP) and electroencephalography (EEG). For circuit simulations, we will use NetPyNE, a high-level interface to NEURON supporting programmatic and GUI specifications that facilitate the development, parallel simulation, and analysis of biophysically detailed neuronal circuits. We conclude with an example combining all three tools that link intracellular/extracellular molecular dynamics with network spiking activity and LFP/EEG. The tutorial will incorporate recent developments and new features in the NEURON and NetPyNE tools.

Schedule:

9.00 - 10.15: NEURON
10.15 - 10.45: Coffee Break
10.45 - 12.15: NEURON + RxD
12.15 - 14.00: Lunch
14.00 - 15.40: NetPyNE Web App
15.40 - 16.00: Break
16.00 - 17.15: NetPyNE Programmatic

Detailed NetPyNE Schedule:

14.00-14.30 (30 min): Introduction to multiscale brain circuit modeling in NetPyNE

14.30-15.40 (1h10 min): Graphical Web App Tutorials

- Overview of Web App and OSB environment (10 min)
- Simple network (10 min)
- Network with morphologically detailed neurons (10 min)
- Multiscale cortical network: from RxD to LFP/EEG (20 min)
- Automated network parameter exploration (20 min)

15.40-16.00 (20 min): COFFEE BREAK

16.00-17.15 (1h15 min): Jupyter Notebook Programmatic Tutorials

- OSB JupyterLab environment (15 min)
- Network oscillations tutorial (30 min)
- Other tutorials (importing cells, analysis/plotting) (20 min)

Speakers (in alphabetical order):

Valery Bragin, NetPyNE circuit modeling

Charité – Berlin University Medicine / State University of New York (SUNY) Downstate Health

Sciences University

Salvador Dura-Bernal, NetPyNE circuit modeling

State University of New York (SUNY) Downstate Health Sciences University

William W Lytton, Multiscale Modeling Overview

State University of New York (SUNY) Downstate Health Sciences University

Robert A McDougal, NEURON single cells

Yale University

Adam Newton, NEURON Reaction-Diffusion

State University of New York (SUNY) Downstate Health Sciences University

9:00am –
10:15am

T **T02: From single-cell modeling to large-scale network dynamics with NEST Simulator** Cedro II

Speakers: Iiro Ahokainen, Jasper Albers, Joshua Böttcher

NEST is an established community code for simulating spiking neuronal network models that capture the full details of the structure of biological networks [1]. The simulator runs efficiently on various architectures, from laptops to supercomputers [2]. Over the years, a large body of peer-reviewed neuroscientific studies have been carried out with NEST, and it has become the reference code for research on neuromorphic hardware systems.

This tutorial provides hands-on experience with recent NEST feature additions. First, we explore how an astrocyte-mediated slow inward current impacts typical neural network simulations. Here, we introduce how astrocytes are implemented in NEST and investigate their dynamical behavior. Then, we create small neuron-astrocyte networks and explore their interactions before adding more complexity to the network structure. Second, we develop a functional network that can be trained to solve various tasks using a three-factor learning rule that approximates backpropagation through time: eligibility propagation (e-prop). Specifically, we use e-prop to train a network to solve a supervised regression task to generate temporal patterns and a supervised classification task to accumulate evidence. Third, we investigate how dendritic properties of neurons can be captured by constructing compartmental models in NEST. We import dendritic models from an existing repository and embed them in a network simulation. Finally, we learn to use NESTML, a domain-specific modeling language for neuron and synapse models. We implement a neuron model with an active dendritic compartment and a third-factor STDP synapse defined in NESTML. These models are then used in a network to perform learning, prediction, and replay of sequences of items, such as letters, images, or sounds [3].

[1] Gewaltig M-O & Diesmann M (2007) NEST (Neural Simulation Tool) Scholarpedia 2(4):1430.

[2] Jordan J., Ippen T., Helias M., Kitayama I., Sato M., Igarashi J., Diesmann M., Kunkel S. (2018). Extremely Scalable Spiking Neuronal Network Simulation Code: From Laptops to Exascale Computers. *Frontiers in Neuroinformatics* 12: 2

[3] Bouhadjar Y, Wouters DJ, Diesmann M, Tetzlaff T (2022) Sequence learning, prediction, and replay in networks of spiking neurons. *PLoS Comput Biol* 18(6): e1010233.

Speakers (in alphabetical order):

Iiro Ahokainen, Astrocytes in NEST
Tampere University, Finland

Jasper Albers, E-prop in NEST
Jülich Research Centre, Germany

Joshua Boettcher, Compartmental models in NEST; NESTML
Jülich Research Centre, Germany

Please visit the Tutorial Website for updated information:
<https://clinssen.github.io/OCNS-2024-NEST-workshop/>

9:00am –
10:15am

T **T03: Modeling cortical networks dynamics**

Cedro III

Speakers: Maurizio De Pitta, Pamela Illescas-Maldonado, Scott Rich, Alessandro Sanzeni

The Tutorial aims to provide an essential introduction to modeling biophysically realistic neuronal networks, emphasizing essential circuital components underpinning asynchronous vs. synchronous dynamics. The morning session will introduce classic models of balanced excitatory/inhibitory (E–I) networks with analytical insights into some mechanisms

for the emergence of asynchronous and irregular firing. The afternoon session will shift focus towards network models displaying synchronous dynamics, with hands-on interactive jupyter sessions and practical numerical simulations delving into fundamental theory and clinical applications, including models of epileptiform activity.

MORNING SESSION: Asynchronous dynamics

Emergence of irregular activity in networks of strongly coupled spiking neurons

Alessandro Sanzeni, Bocconi University, Milan, Italy

Introducing glia into cortical network models and the emergence of glial attractors

Maurizio De Pitta, Krembil Research Institute, Toronto, Canada

AFTERNOON SESSION: Synchronous Activity

Oscillations in networks of excitatory and inhibitory neurons: the PING framework

Scott Rich, University of Connecticut, CN, USA

Simulating network models of reactive astrogliosis underpinning epilepsy

Pamela Illescas-Maldonado, and **Vicente Medel**, University of Valparaiso, Valparaiso, Chile

9:00am –
10:15am

T **T04: Standardised, data-driven computational modelling with NeuroML using the Open Source Brian**

Cedro IV

Speakers: Pdraig Gleeson, Boris Marin, Ankur Sinha, Angus Silver

Data-driven models of neurons and circuits are important for understanding how the properties of membrane conductances, synapses, dendrites, and the anatomical connectivity between neurons generate the complex dynamical behaviors of brain circuits in health and disease. However, even though data and models have been made publicly available in recent years and the use of standards such as Neurodata Without Borders (NWB) (<https://nwb.org>)

and NeuroML (<https://neuroml.org>) to promote FAIR (Findable, Accessible, Interoperable, and Reusable) neuroscience is on the rise, but the development of data-driven models remains hampered by the difficulty of finding appropriate data and the inherent complexity involved in their construction.

The Open Source Brain web platform (OSB) (<https://opensourcebrain.org>) combines data, accompanying analysis tools, and computational models in a scalable resource. It indexes repositories from established sources such as the DANDI data archive (<https://dandiarchive.org>), the ModelDB model sharing archive (<https://modeldb.science>), and GitHub to provide easy access to a plethora of experimental data and models, including a large number standardized in NWB and NeuroML formats. OSB also incorporates the NeuroML software ecosystem. NeuroML is an established community standard and software ecosystem that enables the development of detailed biophysical models using a declarative, simulator-independent description. The software ecosystem supports all steps of the model lifecycle and allows users to automatically generate code and run their NeuroML models using some well-established simulation engines (NEURON/NetPyNE).

In this tutorial, attendees will learn about:

- Finding data and models on OSB
- NeuroML and its software ecosystem
- Using NeuroML models on OSB
- Building and simulating new NeuroML models constrained by the data on OSB

We will also assist with advanced tasks and discuss new features to aid researchers further.

Agenda

- **0900 - 1015: Introduction: why NeuroML?**
- **1045 - 1115: Installing NeuroML (hands-on)**
- **1115 - 1215: Introduction to modelling with NeuroML (hands-on)**
- **1400 - 1700: Advanced modelling with NeuroML/Model conversion/Troubleshooting (hands-on)**

Speakers (in alphabetical order):

Pdraig Gleeson, University College London, London, UK

Boris Marin, Universidade Federal do ABC, Brazil

Angus Silver, University College London, London, UK

Ankur Sinha, University College London, London, UK

9:00am –
10:15am

T T05: Understanding motor control through multiscale modeling of spinal cord neuronal circuits

Speakers: Ricardo Gonçalves Molinari, Leonardo Abdala Elias, Renato Naville Watanabe, André Cedro V Fabio Kohn

Circuits of neurons in the spinal cord process sensory and descending information to produce the neural drive to the muscle and the ensuing movement. Dysfunctions in these circuits could generate abnormal movements, for instance, tremors. Models of spinal cord circuits have been used for several years to enlighten our understanding of basic principles of motor control, namely the recruitment and rate coding of motor units that explain force gradation and movement smoothness. More recently, computer simulations of biophysical models of the neuromuscular system provide information on 1) how interneuron circuits could attenuate/cancel tremor signals, 2) how modulating sensory signals could produce intermittent recruitment of motor units in a posture control task, and 3) how axon demyelination could impact force and position control. In this tutorial, we will present an overview of neuromusculoskeletal models and several application examples. Two hands-on sessions will follow the introductory talk. The first session will use a web-based neuromuscular simulator (ReMoto – <http://remoto.leb.usp.br>) that can be easily configured (without coding) to study several aspects of force generation and control. Finally, the attendees will design a spinal cord circuit from scratch using general-purpose simulators of neurons and neuronal networks (NEURON and NetPyNE). In the latter session, a model including a pool of motor neurons with stochastic synaptic inputs will be used to show how the motor pool could reduce the independent synaptic noise and transmit the common input to the motor output.

Speakers

- André Fabio Kohn: *Neuromusculoskeletal models and their applications (remote talk)*
 - Renato Naville Watanabe: *Hands-on session using ReMoto simulator*
 - Leonardo Abdala Elias and Ricardo Gonçalves Molinari: *Hands-on session on designing spinal cord circuits using general-purpose simulators of neurons and neuronal networks (NEURON and NetPyNE)*
-

9:00am –
10:15am

T **T06: Implementing the Gaussian-Linear Hidden Markov model (GLHMM), with a package in Python for brain data analysis**

Cedro VI

Speakers: Cecilia Jarne, Diego Vidaurre

Hidden Markov Models (HMMs) are a type of statistical models used to model sequences of data where the underlying state of the system is not directly observable. They are a powerful tool that has been used in several applications, including speech recognition, natural language processing, and bioinformatics, mainly because of their data-driven approach.

For this tutorial, we introduce the GLHMM model and Python package (<https://github.com/vidaurre/glhmm>). In short, the GLHMM is a general framework where linear regression is used to parameterise a Gaussian state distribution, thereby it can accommodate a wide range of uses -including unsupervised, encoding and decoding models. GLHMM is implemented as a Python toolbox with emphasis on statistical testing and out-of-sample prediction -i.e. aimed to find and characterise brain-behaviour associations. This toolbox uses a stochastic variational inference approach, enabling it to handle large data sets at reasonable computational time. This approach can be applied to several data modalities, including animal recordings or non- brain data, and applied over a broad range of experimental paradigms.

For demonstration in this tutorial, we will show examples with fMRI data. The goal of this tutorial is to provide a step-by-step guide to using the toolbox. It's aimed at Master's and PhD. students (and Postdocs) interested in learning how to implement HMMs mainly for brain data analysis but not exclusively.

Objectives:

- 1) Implement the algorithm with a Python package.
- 2) To use the algorithm to estimate relevant parameters of an HMM with a practical example on HCP (<https://www.humanconnectome.org/>)

Prerequisites:

To complete this tutorial, participants will need to have the following knowledge and skills:

- 1) Basic knowledge of programming in Python.
- 2) Basic knowledge of probability and statistics.

Materials:

The tutorial will be available online in a Collab notebook. The source code for the algorithm will also be available online on github (<https://github.com/vidaurre/glhmm>).

Bibliography:

- Diego Vidaurre, Nick Y. Larsen, Laura Masaracchia, Lenno R.P.T Ruijters, Sonsoles Alonso, Christine Ahrends, Mark W. Woolrich. The Gaussian-Linear Hidden Markov model: a Python package.2023 <https://arxiv.org/abs/2312.07151>
- Diego Vidaurre, Stephen M. Smith, and Mark W. Woolrich. Brain network dynamics are hierarchically organized in time. 2017 <https://www.pnas.org/doi/full/10.1073/pnas.1705120114>
- Diego Vidaurre, Romesh Abeyesuriya, Robert Becker, Andrew J. Quinn, Fidel Alvaro-Almagro, Stephen M. Smith, Mark W. Woolrich, Discovering dynamic brain networks from big data in rest and task, *NeuroImage*, <https://doi.org/10.1016/j.neuroimage.2017.06.077>
- D. Vidaurre, A. Llera, S.M. Smith, M.W. Woolrich, Behavioural relevance of spontaneous, transient brain network interactions in fMRI, *NeuroImage*, <https://doi.org/10.1016/j.neuroimage.2020.117713>

9:00am –
10:15am

T T08: Single cell signal processing and data analysis in Matlab

Cedro VI

Speakers: Cengiz Gunay

Matlab (Mathworks, Natick, MA) is a popular computing environment that offers an alternative to more advanced environments with its simplicity, especially for those less computationally inclined or for collaborating with experimentalists. In this tutorial, we will focus on the following tasks in Matlab: (1) Signal processing of recorded or simulated traces (e.g., filtering noise, spike and burst finding in single-unit intracellular electrophysiology data in current-clamp, and extracting numerical characteristics); (2) analyzing tabular data (e.g. obtained from Excel or from the result of other analyses); and (3) plotting and visualization. For all of these, we will take advantage of the PANDORA toolbox, which is an open-source project that has been proposed for analysis and visualization (RRID: SCR_001831, [1]). PANDORA was initially developed for managing and analyzing brute-force neuronal parameter search databases. However, it has proven useful for various other types of simulation or experimental data analysis [2-7]. PANDORA's original motivation was to offer an object-oriented program for analyzing neuronal data inside the Matlab environment, in particular with a database table-like object, similar to the "dataframe" object offered in the R ecosystem and the pandas Python module. PANDORA offers a similarly convenient syntax for a powerful database querying system. A typical workflow would constitute of generating parameter sets for simulations, and then analyze the resulting simulation output and other recorded data, to find spikes and to measure additional characteristics to construct databases, and finally analyze and visualize these database contents. PANDORA provides objects for loading datasets, controlling simulations, importing/exporting data, and visualization. In this tutorial, we use the toolbox's standard features and show how to customize them for a given project.

References:

1. Günay et al. 2009 Neuroinformatics, 7(2):93-111. doi: 10.1007/s12021-009-9048-z
2. Doloc-Mihu et al. 2011 Journal of biological physics, 37(3), 263–283. doi:10.1007/s10867-011- 9215-y;
3. Lin et al. 2012 J Neurosci 32(21): 7267–77;
4. Wolfram et al. 2014 J Neurosci, 34(7): 2538–2543; doi: 10.1523/JNEUROSCI.4511-13.2014;
5. Günay et al. 2015 PLoS Comp Bio. doi: 10.1371/journal.pcbi.1004189;
6. Wenning et al. 2018 eLife 2018;7:e31123 doi: 10.7554/eLife.31123;
7. Günay et al. 2019 eNeuro, 6(4), ENEURO.0417-18.2019. doi:10.1523/ENEURO.0417-18.2019

10:15am –
10:45am

B Coffee break

10:45am –
12:15pm

T T01: Building mechanistic multiscale models using NEURON and NetPyNE to study brain function and disease

Cedro I

Speakers: Valery Bragin, Salvador Dura Bernal, William W Lytton, Adam J.H. Newton, Robert McDougal

Understanding the brain requires studying its multiscale interactions, from molecules to cells to circuits and networks. Although vast experimental datasets are being generated across scales and modalities, integrating and interpreting this data remains a daunting challenge. This tutorial will highlight recent advances in mechanistic multiscale modeling and how it offers an unparalleled approach to integrate these data and provide insights into brain function and disease. Multiscale models facilitate the interpretation of experimental findings across different brain regions, brain scales (molecular, cellular, circuit, system), brain function (sensory perception, motor behavior, learning, etc), recording/imaging modalities (intracellular voltage, LFP, EEG, fMRI, etc) and disease/disorders (e.g., schizophrenia, epilepsy, ischemia, Parkinson's, etc). As such, it broadly appeals to experimental, clinical, and computational neuroscientists, students, and educators.

This tutorial will introduce multiscale modeling using two NIH-funded tools: the NEURON 9.0 simulator (<https://neuron.yale.edu/neuron/>), including the Reaction-Diffusion (RxD) module and the NetPyNE tool (<http://netpyne.org>). The tutorial will combine background, examples, and hands-on exercises covering the implementation of models at four key scales: (1) intracellular dynamics (e.g., calcium buffering, protein interactions), (2) single neuron electrophysiology (e.g., action potential propagation), (3) neurons in extracellular space (e.g., spreading depression), and (4) neuronal circuits, including dynamics such as oscillations and simulation of recordings such as local field potentials (LFP) and electroencephalography (EEG). For circuit simulations, we will use NetPyNE, a high-level interface to NEURON supporting programmatic and GUI specifications that facilitate the development, parallel simulation, and analysis of biophysically detailed neuronal circuits. We conclude with an example combining all three tools that link intracellular/extracellular molecular dynamics with network spiking activity and LFP/EEG. The tutorial will incorporate recent developments and new features in the NEURON and NetPyNE tools.

Schedule:

9.00 - 10.15: NEURON
10.15 - 10.45: Coffee Break
10.45 - 12.15: NEURON + RxD
12.15 - 14.00: Lunch
14.00 - 15.40: NetPyNE Web App
15.40 - 16.00: Break
16.00 - 17.15: NetPyNE Programmatic

Detailed NetPyNE Schedule:

14.00-14.30 (30 min): Introduction to multiscale brain circuit modeling in NetPyNE
14.30-15.40 (1h10 min): Graphical Web App Tutorials

- Overview of Web App and OSB environment (10 min)
- Simple network (10 min)
- Network with morphologically detailed neurons (10 min)
- Multiscale cortical network: from RxD to LFP/EEG (20 min)
- Automated network parameter exploration (20 min)

15.40-16.00 (20 min): COFFEE BREAK

16.00-17.15 (1h15 min): Jupyter Notebook Programmatic Tutorials

- OSB JupyterLab environment (15 min)
- Network oscillations tutorial (30 min)
- Other tutorials (importing cells, analysis/plotting) (20 min)

Speakers (in alphabetical order):

Valery Bragin, NetPyNE circuit modeling

Charité – Berlin University Medicine / State University of New York (SUNY) Downstate Health Sciences University

Salvador Dura-Bernal, NetPyNE circuit modeling
State University of New York (SUNY) Downstate Health Sciences University

William W Lytton, Multiscale Modeling Overview
State University of New York (SUNY) Downstate Health Sciences University

Robert A McDougal, NEURON single cells
Yale University

Adam Newton, NEURON Reaction-Diffusion
State University of New York (SUNY) Downstate Health Sciences University

10:45am –
12:15pm

T **T02: From single-cell modeling to large-scale network dynamics with NEST Simulator** Cedro II

Speakers: Iiro Ahokainen, Jasper Albers, Joshua Böttcher

NEST is an established community code for simulating spiking neuronal network models that capture the full details of the structure of biological networks [1]. The simulator runs efficiently on various architectures, from laptops to supercomputers [2]. Over the years, a large body of peer-reviewed neuroscientific studies have been carried out with NEST, and it has become the reference code for research on neuromorphic hardware systems.

This tutorial provides hands-on experience with recent NEST feature additions. First, we explore how an astrocyte-mediated slow inward current impacts typical neural network simulations. Here, we introduce how astrocytes are implemented in NEST and investigate their dynamical behavior. Then, we create small neuron-astrocyte networks and explore their interactions before adding more complexity to the network structure. Second, we develop a functional network that can be trained to solve various tasks using a three-factor learning rule that approximates backpropagation through time: eligibility propagation (e-prop). Specifically, we use e-prop to train a network to solve a supervised regression task to generate temporal patterns and a supervised classification task to accumulate evidence. Third, we investigate how dendritic properties of neurons can be captured by constructing compartmental models in NEST. We import dendritic models from an existing repository and embed them in a network simulation. Finally, we learn to use NESTML, a domain-specific modeling language for neuron and synapse models. We implement a neuron model with an active dendritic compartment and a third-factor STDP synapse defined in NESTML. These models are then used in a network to perform learning, prediction, and replay of sequences of items, such as letters, images, or sounds [3].

[1] Gewaltig M-O & Diesmann M (2007) NEST (Neural Simulation Tool) Scholarpedia 2(4):1430.

[2] Jordan J., Ippen T., Helias M., Kitayama I., Sato M., Igarashi J., Diesmann M., Kunkel S. (2018). Extremely Scalable Spiking Neuronal Network Simulation Code: From Laptops to Exascale Computers. *Frontiers in Neuroinformatics* 12: 2

[3] Bouhadjar Y, Wouters DJ, Diesmann M, Tetzlaff T (2022) Sequence learning, prediction, and replay in networks of spiking neurons. *PLoS Comput Biol* 18(6): e1010233.

Speakers (in alphabetical order):

Iiro Ahokainen, Astrocytes in NEST
Tampere University, Finland

Jasper Albers, E-prop in NEST
Jülich Research Centre, Germany

Joshua Boettcher, Compartmental models in NEST; NESTML
Jülich Research Centre, Germany

Please visit the Tutorial Website for updated information:
<https://clinssen.github.io/OCNS-2024-NEST-workshop/>

10:45am –
12:15pm

T **T03: Modeling cortical networks dynamics**

Cedro III

Speakers: Pamela Illescas-Maldonado, Maurizio De Pitta, Scott Rich, Alessandro Sanzeni

The Tutorial aims to provide an essential introduction to modeling biophysically realistic neuronal networks, emphasizing essential circuital components underpinning asynchronous vs. synchronous dynamics. The morning session will introduce classic models of balanced excitatory/inhibitory (E–I) networks with analytical insights into some mechanisms

for the emergence of asynchronous and irregular firing. The afternoon session will shift focus towards network models displaying synchronous dynamics, with hands-on interactive jupyter sessions and practical numerical simulations delving into fundamental theory and clinical applications, including models of epileptiform activity.

MORNING SESSION: Asynchronous dynamics

Emergence of irregular activity in networks of strongly coupled spiking neurons

Alessandro Sanzeni, Bocconi University, Milan, Italy

Introducing glia into cortical network models and the emergence of glial attractors

Maurizio De Pitta, Krembil Research Institute, Toronto, Canada

AFTERNOON SESSION: Synchronous Activity

Oscillations in networks of excitatory and inhibitory neurons: the PING framework

Scott Rich, University of Connecticut, CN, USA

Simulating network models of reactive astrogliosis underpinning epilepsy

Pamela Illescas-Maldonado, and **Vicente Medel**, University of Valparaiso, Valparaiso, Chile

10:45am –
12:15pm

T **T04: Standardised, data-driven computational modelling with NeuroML using the Open Source Brian** Cedro IV

Speakers: Ankur Sinha, Padraig Gleeson, Boris Marin, Angus Silver

Data-driven models of neurons and circuits are important for understanding how the properties of membrane conductances, synapses, dendrites, and the anatomical connectivity between neurons generate the complex dynamical behaviors of brain circuits in health and disease. However, even though data and models have been made publicly available in recent years and the use of standards such as Neurodata Without Borders (NWB) (<https://nwb.org>)

and NeuroML (<https://neuroml.org>) to promote FAIR (Findable, Accessible, Interoperable, and Reusable) neuroscience is on the rise, but the development of data-driven models remains hampered by the difficulty of finding appropriate data and the inherent complexity involved in their construction.

The Open Source Brain web platform (OSB) (<https://opensourcebrain.org>) combines data, accompanying analysis tools, and computational models in a scalable resource. It indexes repositories from established sources such as the DANDI data archive (<https://dandiarchive.org>), the ModelDB model sharing archive (<https://modeldb.science>), and GitHub to provide easy access to a plethora of experimental data and models, including a large number standardized in NWB and NeuroML formats. OSB also incorporates the NeuroML software ecosystem. NeuroML is an established community standard and software ecosystem that enables the development of detailed biophysical models using a declarative, simulator-independent description. The software ecosystem supports all steps of the model lifecycle and allows users to automatically generate code and run their NeuroML models using some well-established simulation engines (NEURON/NetPyNE).

In this tutorial, attendees will learn about:

- Finding data and models on OSB
- NeuroML and its software ecosystem
- Using NeuroML models on OSB
- Building and simulating new NeuroML models constrained by the data on OSB

We will also assist with advanced tasks and discuss new features to aid researchers further.

Agenda

- **0900 - 1015: Introduction: why NeuroML?**
- **1045 - 1115: Installing NeuroML (hands-on)**
- **1115 - 1215: Introduction to modelling with NeuroML (hands-on)**
- **1400 - 1700: Advanced modelling with NeuroML/Model conversion/Troubleshooting (hands-on)**

Speakers (in alphabetical order):

Padraig Gleeson, University College London, London, UK

Boris Marin, Universidade Federal do ABC, Brazil

Angus Silver, University College London, London, UK

Ankur Sinha, University College London, London, UK

10:45am –
12:15pm

T T05: Understanding motor control through multiscale modeling of spinal cord neuronal circuits
Speakers: Ricardo Gonçalves Molinari, Leonardo Abdala Elias, Renato Naville Watanabe, André Cedro V Fabio Kohn

Circuits of neurons in the spinal cord process sensory and descending information to produce the neural drive to the muscle and the ensuing movement. Dysfunctions in these circuits could generate abnormal movements, for instance, tremors. Models of spinal cord circuits have been used for several years to enlighten our understanding of basic principles of motor control, namely the recruitment and rate coding of motor units that explain force gradation and movement smoothness. More recently, computer simulations of biophysical models of the neuromuscular system provide information on 1) how interneuron circuits could attenuate/cancel tremor signals, 2) how modulating sensory signals could produce intermittent recruitment of motor units in a posture control task, and 3) how axon demyelination could impact force and position control. In this tutorial, we will present an overview of neuromusculoskeletal models and several application examples. Two hands-on sessions will follow the introductory talk. The first session will use a web-based neuromuscular simulator (ReMoto – <http://remoto.leb.usp.br>) that can be easily configured (without coding) to study several aspects of force generation and control. Finally, the attendees will design a spinal cord circuit from scratch using general-purpose simulators of neurons and neuronal networks (NEURON and NetPyNE). In the latter session, a model including a pool of motor neurons with stochastic synaptic inputs will be used to show how the motor pool could reduce the independent synaptic noise and transmit the common input to the motor output.

Speakers

- André Fabio Kohn: *Neuromusculoskeletal models and their applications (remote talk)*
 - Renato Naville Watanabe: *Hands-on session using ReMoto simulator*
 - Leonardo Abdala Elias and Ricardo Gonçalves Molinari: *Hands-on session on designing spinal cord circuits using general-purpose simulators of neurons and neuronal networks (NEURON and NetPyNE)*
-

10:45am –
12:15pm

T **T06: Implementing the Gaussian-Linear Hidden Markov model (GLHMM), with a package in Python for brain data analysis**

Cedro VI

Speakers: Cecilia Jarne, Diego Vidaurre

Hidden Markov Models (HMMs) are a type of statistical models used to model sequences of data where the underlying state of the system is not directly observable. They are a powerful tool that has been used in several applications, including speech recognition, natural language processing, and bioinformatics, mainly because of their data-driven approach.

For this tutorial, we introduce the GLHMM model and Python package (<https://github.com/vidaurre/glhmm>). In short, the GLHMM is a general framework where linear regression is used to parameterise a Gaussian state distribution, thereby it can accommodate a wide range of uses -including unsupervised, encoding and decoding models. GLHMM is implemented as a Python toolbox with emphasis on statistical testing and out-of-sample prediction -i.e. aimed to find and characterise brain-behaviour associations. This toolbox uses a stochastic variational inference approach, enabling it to handle large data sets at reasonable computational time. This approach can be applied to several data modalities, including animal recordings or non- brain data, and applied over a broad range of experimental paradigms.

For demonstration in this tutorial, we will show examples with fMRI data. The goal of this tutorial is to provide a step-by-step guide to using the toolbox. It's aimed at Master's and PhD. students (and Postdocs) interested in learning how to implement HMMs mainly for brain data analysis but not exclusively.

Objectives:

- 1) Implement the algorithm with a Python package.
- 2) To use the algorithm to estimate relevant parameters of an HMM with a practical example on HCP (<https://www.humanconnectome.org/>)

Prerequisites:

To complete this tutorial, participants will need to have the following knowledge and skills:

- 1) Basic knowledge of programming in Python.
- 2) Basic knowledge of probability and statistics.

Materials:

The tutorial will be available online in a Collab notebook. The source code for the algorithm will also be available online on github (<https://github.com/vidaurre/glhmm>).

Bibliography:

- Diego Vidaurre, Nick Y. Larsen, Laura Masaracchia, Lenno R.P.T Ruijters, Sonsoles Alonso, Christine Ahrends, Mark W. Woolrich. The Gaussian-Linear Hidden Markov model: a Python package.2023 <https://arxiv.org/abs/2312.07151>
- Diego Vidaurre, Stephen M. Smith, and Mark W. Woolrich. Brain network dynamics are hierarchically organized in time. 2017 <https://www.pnas.org/doi/full/10.1073/pnas.1705120114>
- Diego Vidaurre, Romesh Abeyesuriya, Robert Becker, Andrew J. Quinn, Fidel Alfaro-Almagro, Stephen M. Smith, Mark W. Woolrich, Discovering dynamic brain networks from big data in rest and task, *NeuroImage*, <https://doi.org/10.1016/j.neuroimage.2017.06.077>
- D. Vidaurre, A. Llera, S.M. Smith, M.W. Woolrich, Behavioural relevance of spontaneous, transient brain network interactions in fMRI, *NeuroImage*, <https://doi.org/10.1016/j.neuroimage.2020.117713>

10:45am –
12:15pm

T T08: Single cell signal processing and data analysis in Matlab

Jacarandá

Speakers: Cengiz Gunay

Matlab (Mathworks, Natick, MA) is a popular computing environment that offers an alternative to more advanced environments with its simplicity, especially for those less computationally inclined or for collaborating with experimentalists. In this tutorial, we will focus on the following tasks in Matlab: (1) Signal processing of recorded or simulated traces (e.g., filtering noise, spike and burst finding in single-unit intracellular electrophysiology data in current-clamp, and extracting numerical characteristics); (2) analyzing tabular data (e.g. obtained from Excel or from the result of other analyses); and (3) plotting and visualization. For all of these, we will take advantage of the PANDORA toolbox, which is an open-source project that has been proposed for analysis and visualization (RRID: SCR_001831, [1]). PANDORA was initially developed for managing and analyzing brute-force neuronal parameter search databases. However, it has proven useful for various other types of simulation or experimental data analysis [2-7]. PANDORA's original motivation was to offer an object-oriented program for analyzing neuronal data inside the Matlab environment, in particular with a database table-like object, similar to the "dataframe" object offered in the R ecosystem and the pandas Python module. PANDORA offers a similarly convenient syntax for a powerful database querying system. A typical workflow would constitute of generating parameter sets for simulations, and then analyze the resulting simulation output and other recorded data, to find spikes and to measure additional characteristics to construct databases, and finally analyze and visualize these database contents. PANDORA provides objects for loading datasets, controlling simulations, importing/exporting data, and visualization. In this tutorial, we use the toolbox's standard features and show how to customize them for a given project.

References:

1. Günay et al. 2009 Neuroinformatics, 7(2):93-111. doi: 10.1007/s12021-009-9048-z
2. Doloc-Mihu et al. 2011 Journal of biological physics, 37(3), 263–283. doi:10.1007/s10867-011- 9215-y;
3. Lin et al. 2012 J Neurosci 32(21): 7267–77;
4. Wolfram et al. 2014 J Neurosci, 34(7): 2538–2543; doi: 10.1523/JNEUROSCI.4511-13.2014;
5. Günay et al. 2015 PLoS Comp Bio. doi: 10.1371/journal.pcbi.1004189;
6. Wenning et al. 2018 eLife 2018;7:e31123 doi: 10.7554/eLife.31123;
7. Günay et al. 2019 eNeuro, 6(4), ENEURO.0417-18.2019. doi:10.1523/ENEURO.0417-18.2019

12:15pm –
2:00pm

B Lunch

2:00pm – 3:40pm

T T01: Building mechanistic multiscale models using NEURON and NetPyNE to study brain function and disease

Cedro I

Speakers: Valery Bragin, Salvador Dura Bernal, William W Lytton, Robert McDougal, Adam J.H. Newton

Understanding the brain requires studying its multiscale interactions, from molecules to cells to circuits and networks. Although vast experimental datasets are being generated across scales and modalities, integrating and interpreting this data remains a daunting challenge. This tutorial will highlight recent advances in mechanistic multiscale modeling and how it offers an unparalleled approach to integrate these data and provide insights into brain function and disease. Multiscale models facilitate the interpretation of experimental findings across different brain regions, brain scales (molecular, cellular, circuit, system), brain function (sensory perception, motor behavior, learning, etc), recording/imaging modalities (intracellular voltage, LFP, EEG, fMRI, etc) and disease/disorders (e.g., schizophrenia, epilepsy, ischemia, Parkinson's, etc). As such, it broadly appeals to experimental, clinical, and computational neuroscientists, students, and educators.

This tutorial will introduce multiscale modeling using two NIH-funded tools: the NEURON 9.0 simulator (<https://neuron.yale.edu/neuron/>), including the Reaction-Diffusion (RxD) module and the NetPyNE tool (<http://netpyne.org>). The tutorial will combine background, examples, and hands-on exercises covering the implementation of models at four key scales: (1) intracellular dynamics (e.g., calcium buffering, protein interactions), (2) single neuron electrophysiology (e.g., action potential propagation), (3) neurons in extracellular space (e.g., spreading depression), and (4) neuronal circuits, including dynamics such as oscillations and simulation of recordings such as local field potentials (LFP) and electroencephalography (EEG). For circuit simulations, we will use NetPyNE, a high-level interface to NEURON supporting programmatic and GUI specifications that facilitate the development, parallel simulation, and analysis of biophysically detailed neuronal circuits. We conclude with an example combining all three tools that link intracellular/extracellular molecular dynamics with network spiking activity and LFP/EEG. The tutorial will incorporate recent developments and new features in the NEURON and NetPyNE tools.

Schedule:

9.00 - 10.15: NEURON
 10.15 - 10.45: Coffee Break
 10.45 - 12.15: NEURON + RxD
 12.15 - 14.00: Lunch
 14.00 - 15.40: NetPyNE Web App
 15.40 - 16.00: Break
 16.00 - 17.15: NetPyNE Programmatic

Detailed NetPyNE Schedule:

14.00-14.30 (30 min): Introduction to multiscale brain circuit modeling in NetPyNE
 14.30-15.40 (1h10 min): Graphical Web App Tutorials

- Overview of Web App and OSB environment (10 min)
- Simple network (10 min)
- Network with morphologically detailed neurons (10 min)
- Multiscale cortical network: from RxD to LFP/EEG (20 min)
- Automated network parameter exploration (20 min)

15.40-16.00 (20 min): COFFEE BREAK

16.00-17.15 (1h15 min): Jupyter Notebook Programmatic Tutorials

- OSB JupyterLab environment (15 min)
- Network oscillations tutorial (30 min)
- Other tutorials (importing cells, analysis/plotting) (20 min)

Speakers (in alphabetical order):

Valery Bragin, NetPyNE circuit modeling

Charité – Berlin University Medicine / State University of New York (SUNY) Downstate Health Sciences University

Salvador Dura-Bernal, NetPyNE circuit modeling
State University of New York (SUNY) Downstate Health Sciences University

William W Lytton, Multiscale Modeling Overview
State University of New York (SUNY) Downstate Health Sciences University

Robert A McDougal, NEURON single cells
Yale University

Adam Newton, NEURON Reaction-Diffusion
State University of New York (SUNY) Downstate Health Sciences University

2:00pm – 3:40pm T **T02: From single-cell modeling to large-scale network dynamics with NEST Simulator** Cedro II

Speakers: Iiro Ahokainen, Jasper Albers, Joshua Böttcher

NEST is an established community code for simulating spiking neuronal network models that capture the full details of the structure of biological networks [1]. The simulator runs efficiently on various architectures, from laptops to supercomputers [2]. Over the years, a large body of peer-reviewed neuroscientific studies have been carried out with NEST, and it has become the reference code for research on neuromorphic hardware systems.

This tutorial provides hands-on experience with recent NEST feature additions. First, we explore how an astrocyte-mediated slow inward current impacts typical neural network simulations. Here, we introduce how astrocytes are implemented in NEST and investigate their dynamical behavior. Then, we create small neuron-astrocyte networks and explore their interactions before adding more complexity to the network structure. Second, we develop a functional network that can be trained to solve various tasks using a three-factor learning rule that approximates backpropagation through time: eligibility propagation (e-prop). Specifically, we use e-prop to train a network to solve a supervised regression task to generate temporal patterns and a supervised classification task to accumulate evidence. Third, we investigate how dendritic properties of neurons can be captured by constructing compartmental models in NEST. We import dendritic models from an existing repository and embed them in a network simulation. Finally, we learn to use NESTML, a domain-specific modeling language for neuron and synapse models. We implement a neuron model with an active dendritic compartment and a third-factor STDP synapse defined in NESTML. These models are then used in a network to perform learning, prediction, and replay of sequences of items, such as letters, images, or sounds [3].

[1] Gewaltig M-O & Diesmann M (2007) NEST (Neural Simulation Tool) Scholarpedia 2(4):1430.

[2] Jordan J., Ippen T., Helias M., Kitayama I., Sato M., Igarashi J., Diesmann M., Kunkel S. (2018). Extremely Scalable Spiking Neuronal Network Simulation Code: From Laptops to Exascale Computers. *Frontiers in Neuroinformatics* 12: 2

[3] Bouhadjar Y, Wouters DJ, Diesmann M, Tetzlaff T (2022) Sequence learning, prediction, and replay in networks of spiking neurons. *PLoS Comput Biol* 18(6): e1010233.

Speakers (in alphabetical order):

Iiro Ahokainen, Astrocytes in NEST
Tampere University, Finland

Jasper Albers, E-prop in NEST
Jülich Research Centre, Germany

Joshua Boettcher, Compartmental models in NEST; NESTML
Jülich Research Centre, Germany

Please visit the Tutorial Website for updated information:
<https://clinssen.github.io/OCNS-2024-NEST-workshop/>

2:00pm – 3:40pm

T **T03: Modeling cortical networks dynamics**

Cedro III

Speakers: Pamela Illescas-Maldonado, Maurizio De Pitta, Scott Rich, Alessandro Sanzeni

The Tutorial aims to provide an essential introduction to modeling biophysically realistic neuronal networks, emphasizing essential circuital components underpinning asynchronous vs. synchronous dynamics. The morning session will introduce classic models of balanced excitatory/inhibitory (E–I) networks with analytical insights into some mechanisms

for the emergence of asynchronous and irregular firing. The afternoon session will shift focus towards network models displaying synchronous dynamics, with hands-on interactive jupyter sessions and practical numerical simulations delving into fundamental theory and clinical applications, including models of epileptiform activity.

MORNING SESSION: Asynchronous dynamics

Emergence of irregular activity in networks of strongly coupled spiking neurons

Alessandro Sanzeni, Bocconi University, Milan, Italy

Introducing glia into cortical network models and the emergence of glial attractors

Maurizio De Pitta, Krembil Research Institute, Toronto, Canada

AFTERNOON SESSION: Synchronous Activity

Oscillations in networks of excitatory and inhibitory neurons: the PING framework

Scott Rich, University of Connecticut, CN, USA

Simulating network models of reactive astrogliosis underpinning epilepsy

Pamela Illescas-Maldonado, and **Vicente Medel**, University of Valparaiso, Valparaiso, Chile

2:00pm – 3:40pm

T T04: Standardised, data-driven computational modelling with NeuroML using the Open Source Brian Cedro IV*Speakers: Boris Marin, Padraig Gleeson, Angus Silver, Ankur Sinha*

Data-driven models of neurons and circuits are important for understanding how the properties of membrane conductances, synapses, dendrites, and the anatomical connectivity between neurons generate the complex dynamical behaviors of brain circuits in health and disease. However, even though data and models have been made publicly available in recent years and the use of standards such as Neurodata Without Borders (NWB) (<https://nwb.org>)

and NeuroML (<https://neuroml.org>) to promote FAIR (Findable, Accessible, Interoperable, and Reusable) neuroscience is on the rise, but the development of data-driven models remains hampered by the difficulty of finding appropriate data and the inherent complexity involved in their construction.

The Open Source Brain web platform (OSB) (<https://opensourcebrain.org>) combines data, accompanying analysis tools, and computational models in a scalable resource. It indexes repositories from established sources such as the DANDI data archive (<https://dandiarchive.org>), the ModelDB model sharing archive (<https://modeldb.science>), and GitHub to provide easy access to a plethora of experimental data and models, including a large number standardized in NWB and NeuroML formats. OSB also incorporates the NeuroML software ecosystem. NeuroML is an established community standard and software ecosystem that enables the development of detailed biophysical models using a declarative, simulator-independent description. The software ecosystem supports all steps of the model lifecycle and allows users to automatically generate code and run their NeuroML models using some well-established simulation engines (NEURON/NetPyNE).

In this tutorial, attendees will learn about:

- Finding data and models on OSB
- NeuroML and its software ecosystem
- Using NeuroML models on OSB
- Building and simulating new NeuroML models constrained by the data on OSB

We will also assist with advanced tasks and discuss new features to aid researchers further.

Agenda

- **0900 - 1015: Introduction: why NeuroML?**
- **1045 - 1115: Installing NeuroML (hands-on)**
- **1115 - 1215: Introduction to modelling with NeuroML (hands-on)**
- **1400 - 1700: Advanced modelling with NeuroML/Model conversion/Troubleshooting (hands-on)**

Speakers (in alphabetical order):

Padraig Gleeson, University College London, London, UK

Boris Marin, Universidade Federal do ABC, Brazil

Angus Silver, University College London, London, UK

Ankur Sinha, University College London, London, UK

2:00pm – 3:40pm

T **T07: Unraveling Dynamics and Connectivity from Spiking Time Series of In-Vitro Neuronal Cultures**

Cedro V

Speakers: Leonardo Novelli, Moein Khajehnejad, Forough Habibollahi

This tutorial will equip participants with comprehensive skills for analyzing spiking time series from DishBrain, a pioneering system demonstrating rudimentary biological intelligence by leveraging the adaptive properties of neurons [1,2]. DishBrain integrates in-vitro neuronal networks with in-silico computing through high-density multi-electrode arrays (HD-MEAs). These cultured neuronal networks exhibit biologically-based adaptive intelligence within a simulated gameplay environment of 'Pong' in real-time, facilitated by closed-loop stimulation and recordings [1].

We will introduce this unique dataset and provide samples of these time series data during both gameplay and spontaneous activity, their behavioral labels in the game environment, and custom Python scripts in Jupyter notebook to analyse spiking data from 1024 channels on the HD-MEA.

The tutorial will emphasize methods for extracting meaningful insights from time series of spiking data. We will address the challenges associated with sparse, high-dimensional data and introduce custom-designed pipelines to address them. These include lower-dimensional embedding algorithms such as UMAP [3], t-SNE [4], and CEBRA [5]. Additionally, we will demonstrate the application of Gaussian kernels for data smoothing followed by community detection techniques to identify influential neurons and reduce computational complexity. These techniques will enable visualization and interpretation of complex neural activity patterns in lower-dimensional spaces, facilitating insights into network dynamics.

Finally, we will introduce methods to quantify statistical dependence and infer connectivity networks from time series, with a particular focus on transfer entropy analysis. Transfer entropy is a powerful tool for quantifying the directed information flow in complex systems [6]. Through hands-on exercises and practical demonstrations, participants will learn how to apply transfer entropy analysis using spiking or continuous data to unravel patterns of information flow in neuronal ensembles.

By the end of the tutorial, participants will have a robust toolkit for analyzing and interpreting high-dimensional, sparse spiking time series and extracting insights into neural network dynamics.

[1] Kagan BJ, Kitchen AC, Tran NT, Habibollahi F, Khajehnejad M, Parker BJ, Bhat A, Rollo B, Razi A, Friston KJ. (2022) In vitro neurons learn and exhibit sentience when embodied in a simulated game-world. *Neuron* 110(23):3952-69.

[2] Cortical Labs: <https://corticallabs.com/>

[3] McInnes, L., Healy, J., Saul, N. & Großberger, L. (2018) UMAP: Uniform Manifold Approximation and Projection for dimension reduction. *J. Open Source Softw.* 3, 86.

[4] Maaten, L. V., Postma, E. O. & Herik, J. V. (2009) Dimensionality reduction: a comparative review. *J. Mach. Learn. Res.* 10, 13.

[5] Schneider S, Lee JH, Mathis MW. (2023) Learnable latent embeddings for joint behavioural and neural analysis. *Nature* 617, 360–368.

[6] Bossomaier, T., Barnett, L., Harré, M., & Lizier, J. T. (2016). *An Introduction to Transfer Entropy*. Springer International Publishing.

Speakers:

- Forough Habibollahi: *Introduction to DishBrain: In-vitro spiking neuronal cultures in an embodied game environment*
- Moein Khajehnejad: *Lower dimensional embedding approaches to study complex network dynamics of neuronal systems*
- Leonardo Novelli: *Inferring directed statistical dependence using Transfer Entropy*

2:00pm – 3:40pm T **T09: Interactive Data Visualization Techniques** Cedro VI

Speakers: Anca Doloc-Mihu, Cengiz Gunay

Nowadays, the Web has become very prominent in computing. As a result, web standards and web programming have seen a tremendous improvement with a large number of tools becoming available at lower or no costs. Trends are visible when observing that traditional desktop applications are moving to web versions one by one. For computational neuroscience, Jupyter notebooks already provided a way to share research results on the web. However, Jupyter still requires a backend server running Python and visualizations are static and not dynamic as one would get from using web-native languages. It is possible to use some Python plugins to generate interactive visualizations, but the real power is achieved when using JavaScript directly via several expansive visualization libraries (Plotly, Vega, D3, Three.js, etc). In this tutorial, we will focus on D3.js for creating interactive visualizations for computational neuroscience applications running directly on the browser. We will review the basics for using online JavaScript notebooks (ObservableHQ) and then show more advanced examples via a hands-on tutorial format.

2:00pm – 3:40pm T **T10: Training recurrent spiking neural networks to generate experimentally recorded neural activities** Jacarandá

Speakers: Christopher Kim

Recent advances in machine learning methods enable training recurrent neural networks (RNNs) to perform highly complex and sophisticated tasks. One of the tasks, particularly interesting to neuroscientists, is to generate experimentally recorded neural activities in recurrent neural networks and study the dynamics of trained networks to investigate the underlying neural mechanism. Here, we showcase how a widely-used training method, known as recursive least squares (or FORCE), can be adopted to train spiking RNNs to reproduce spike recordings of cortical neurons. First, we give an overview of the original FORCE learning, which trains the outputs of rate-based RNNs to perform tasks, and we show how it can be modified to generate arbitrarily complex activity patterns in spiking RNNs. Using this method, we show only a subset of neurons embedded in a network of randomly connected excitatory and inhibitory spiking neurons can be trained to produce cortical activities. We demonstrate GPU implementation of the training algorithm, which enables fast training of large-scale networks, and show that the spiking activities of > 60k neurons recorded with NeuroPixel Probe can be reproduced by spiking RNNs.

Presenter:
Christopher Kim
Staff Scientist
 Laboratory of Biological Modeling, NIDDK, National Institutes of Health, Bethesda, MD, USA

References

1. Kim, C. M., & Chow, C. C. (2018). Learning recurrent dynamics in spiking networks. *Elife*, 7, e37124.
2. Kim, C. M., Finkelstein, A., Chow, C. C., Svoboda, K., & Darshan, R. (2023). Distributing task-related neural activity across a cortical network through task-independent connections. *Nature Communications*, 14(1), 2851.
3. Arthur, B. J., Kim, C. M., Chen, S., Preibisch, S., & Darshan, R. (2023). A scalable implementation of the recursive least-squares algorithm for training spiking neural networks. *Frontiers in Neuroinformatics*, 17.
4. Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4), 544-557.

3:40pm – 4:00pm B **Break**

Tutorial's end and preparation for the inaugural Keynote Lecture

4:00pm – 5:15pm

T T01: Building mechanistic multiscale models using NEURON and NetPyNE to study brain function and disease

Cedro I

Speakers: Valery Bragin, Salvador Dura Bernal, William W Lytton, Robert McDougal, Adam J.H. Newton

Understanding the brain requires studying its multiscale interactions, from molecules to cells to circuits and networks. Although vast experimental datasets are being generated across scales and modalities, integrating and interpreting this data remains a daunting challenge. This tutorial will highlight recent advances in mechanistic multiscale modeling and how it offers an unparalleled approach to integrate these data and provide insights into brain function and disease. Multiscale models facilitate the interpretation of experimental findings across different brain regions, brain scales (molecular, cellular, circuit, system), brain function (sensory perception, motor behavior, learning, etc), recording/imaging modalities (intracellular voltage, LFP, EEG, fMRI, etc) and disease/disorders (e.g., schizophrenia, epilepsy, ischemia, Parkinson's, etc). As such, it broadly appeals to experimental, clinical, and computational neuroscientists, students, and educators.

This tutorial will introduce multiscale modeling using two NIH-funded tools: the NEURON 9.0 simulator (<https://neuron.yale.edu/neuron/>), including the Reaction-Diffusion (RxD) module and the NetPyNE tool (<http://netpyne.org>). The tutorial will combine background, examples, and hands-on exercises covering the implementation of models at four key scales: (1) intracellular dynamics (e.g., calcium buffering, protein interactions), (2) single neuron electrophysiology (e.g., action potential propagation), (3) neurons in extracellular space (e.g., spreading depression), and (4) neuronal circuits, including dynamics such as oscillations and simulation of recordings such as local field potentials (LFP) and electroencephalography (EEG). For circuit simulations, we will use NetPyNE, a high-level interface to NEURON supporting programmatic and GUI specifications that facilitate the development, parallel simulation, and analysis of biophysically detailed neuronal circuits. We conclude with an example combining all three tools that link intracellular/extracellular molecular dynamics with network spiking activity and LFP/EEG. The tutorial will incorporate recent developments and new features in the NEURON and NetPyNE tools.

Schedule:

9.00 - 10.15: NEURON
 10.15 - 10.45: Coffee Break
 10.45 - 12.15: NEURON + RxD
 12.15 - 14.00: Lunch
 14.00 - 15.40: NetPyNE Web App
 15.40 - 16.00: Break
 16.00 - 17.15: NetPyNE Programmatic

Detailed NetPyNE Schedule:

14.00-14.30 (30 min): Introduction to multiscale brain circuit modeling in NetPyNE
 14.30-15.40 (1h10 min): Graphical Web App Tutorials

- Overview of Web App and OSB environment (10 min)
- Simple network (10 min)
- Network with morphologically detailed neurons (10 min)
- Multiscale cortical network: from RxD to LFP/EEG (20 min)
- Automated network parameter exploration (20 min)

15.40-16.00 (20 min): COFFEE BREAK
 16.00-17.15 (1h15 min): Jupyter Notebook Programmatic Tutorials

- OSB JupyterLab environment (15 min)
- Network oscillations tutorial (30 min)
- Other tutorials (importing cells, analysis/plotting) (20 min)

Speakers (in alphabetical order):

Valery Bragin, NetPyNE circuit modeling

Charité – Berlin University Medicine / State University of New York (SUNY) Downstate Health Sciences University

Salvador Dura-Bernal, NetPyNE circuit modeling
State University of New York (SUNY) Downstate Health Sciences University

William W Lytton, Multiscale Modeling Overview
State University of New York (SUNY) Downstate Health Sciences University

Robert A McDougal, NEURON single cells
Yale University

Adam Newton, NEURON Reaction-Diffusion
State University of New York (SUNY) Downstate Health Sciences University

4:00pm – 5:15pm T **T02: From single-cell modeling to large-scale network dynamics with NEST Simulator** Cedro II

Speakers: Iiro Ahokainen, Jasper Albers, Joshua Böttcher

NEST is an established community code for simulating spiking neuronal network models that capture the full details of the structure of biological networks [1]. The simulator runs efficiently on various architectures, from laptops to supercomputers [2]. Over the years, a large body of peer-reviewed neuroscientific studies have been carried out with NEST, and it has become the reference code for research on neuromorphic hardware systems.

This tutorial provides hands-on experience with recent NEST feature additions. First, we explore how an astrocyte-mediated slow inward current impacts typical neural network simulations. Here, we introduce how astrocytes are implemented in NEST and investigate their dynamical behavior. Then, we create small neuron-astrocyte networks and explore their interactions before adding more complexity to the network structure. Second, we develop a functional network that can be trained to solve various tasks using a three-factor learning rule that approximates backpropagation through time: eligibility propagation (e-prop). Specifically, we use e-prop to train a network to solve a supervised regression task to generate temporal patterns and a supervised classification task to accumulate evidence. Third, we investigate how dendritic properties of neurons can be captured by constructing compartmental models in NEST. We import dendritic models from an existing repository and embed them in a network simulation. Finally, we learn to use NESTML, a domain-specific modeling language for neuron and synapse models. We implement a neuron model with an active dendritic compartment and a third-factor STDP synapse defined in NESTML. These models are then used in a network to perform learning, prediction, and replay of sequences of items, such as letters, images, or sounds [3].

[1] Gewaltig M-O & Diesmann M (2007) NEST (Neural Simulation Tool) Scholarpedia 2(4):1430.

[2] Jordan J., Ippen T., Helias M., Kitayama I., Sato M., Igarashi J., Diesmann M., Kunkel S. (2018). Extremely Scalable Spiking Neuronal Network Simulation Code: From Laptops to Exascale Computers. *Frontiers in Neuroinformatics* 12: 2

[3] Bouhadjar Y, Wouters DJ, Diesmann M, Tetzlaff T (2022) Sequence learning, prediction, and replay in networks of spiking neurons. *PLoS Comput Biol* 18(6): e1010233.

Speakers (in alphabetical order):

Iiro Ahokainen, Astrocytes in NEST
Tampere University, Finland

Jasper Albers, E-prop in NEST
Jülich Research Centre, Germany

Joshua Boettcher, Compartmental models in NEST; NESTML
Jülich Research Centre, Germany

Please visit the Tutorial Website for updated information:
<https://clinssen.github.io/OCNS-2024-NEST-workshop/>

4:00pm – 5:15pm T **T03: Modeling cortical networks dynamics** Cedro III

Speakers: Pamela Illescas-Maldonado, Maurizio De Pitta, Scott Rich, Alessandro Sanzeni

The Tutorial aims to provide an essential introduction to modeling biophysically realistic neuronal networks, emphasizing essential circuital components underpinning asynchronous vs. synchronous dynamics. The morning session will introduce classic models of balanced excitatory/inhibitory (E–I) networks with analytical insights into some mechanisms

for the emergence of asynchronous and irregular firing. The afternoon session will shift focus towards network models displaying synchronous dynamics, with hands-on interactive jupyter sessions and practical numerical simulations delving into fundamental theory and clinical applications, including models of epileptiform activity.

MORNING SESSION: Asynchronous dynamics

Emergence of irregular activity in networks of strongly coupled spiking neurons

Alessandro Sanzeni, Bocconi University, Milan, Italy

Introducing glia into cortical network models and the emergence of glial attractors

Maurizio De Pitta, Krembil Research Institute, Toronto, Canada

AFTERNOON SESSION: Synchronous Activity

Oscillations in networks of excitatory and inhibitory neurons: the PING framework

Scott Rich, University of Connecticut, CN, USA

Simulating network models of reactive astrogliosis underpinning epilepsy

Pamela Illescas-Maldonado, and **Vicente Medel**, University of Valparaiso, Valparaiso, Chile

4:00pm – 5:15pm

T T04: Standardised, data-driven computational modelling with NeuroML using the Open Source Brian Cedro IV*Speakers: Boris Marin, Angus Silver, Padraig Gleeson, Ankur Sinha*

Data-driven models of neurons and circuits are important for understanding how the properties of membrane conductances, synapses, dendrites, and the anatomical connectivity between neurons generate the complex dynamical behaviors of brain circuits in health and disease. However, even though data and models have been made publicly available in recent years and the use of standards such as Neurodata Without Borders (NWB) (<https://nwb.org>)

and NeuroML (<https://neuroml.org>) to promote FAIR (Findable, Accessible, Interoperable, and Reusable) neuroscience is on the rise, but the development of data-driven models remains hampered by the difficulty of finding appropriate data and the inherent complexity involved in their construction.

The Open Source Brain web platform (OSB) (<https://opensourcebrain.org>) combines data, accompanying analysis tools, and computational models in a scalable resource. It indexes repositories from established sources such as the DANDI data archive (<https://dandiarchive.org>), the ModelDB model sharing archive (<https://modeldb.science>), and GitHub to provide easy access to a plethora of experimental data and models, including a large number standardized in NWB and NeuroML formats. OSB also incorporates the NeuroML software ecosystem. NeuroML is an established community standard and software ecosystem that enables the development of detailed biophysical models using a declarative, simulator-independent description. The software ecosystem supports all steps of the model lifecycle and allows users to automatically generate code and run their NeuroML models using some well-established simulation engines (NEURON/NetPyNE).

In this tutorial, attendees will learn about:

- Finding data and models on OSB
- NeuroML and its software ecosystem
- Using NeuroML models on OSB
- Building and simulating new NeuroML models constrained by the data on OSB

We will also assist with advanced tasks and discuss new features to aid researchers further.

Agenda

- **0900 - 1015: Introduction: why NeuroML?**
- **1045 - 1115: Installing NeuroML (hands-on)**
- **1115 - 1215: Introduction to modelling with NeuroML (hands-on)**
- **1400 - 1700: Advanced modelling with NeuroML/Model conversion/Troubleshooting (hands-on)**

Speakers (in alphabetical order):

Padraig Gleeson, University College London, London, UK

Boris Marin, Universidade Federal do ABC, Brazil

Angus Silver, University College London, London, UK

Ankur Sinha, University College London, London, UK

4:00pm – 5:15pm

T **T07: Unraveling Dynamics and Connectivity from Spiking Time Series of In-Vitro Neuronal Cultures**

Cedro V

Speakers: Leonardo Novelli, Moein Khajehnejad, Forough Habibollahi

This tutorial will equip participants with comprehensive skills for analyzing spiking time series from DishBrain, a pioneering system demonstrating rudimentary biological intelligence by leveraging the adaptive properties of neurons [1,2]. DishBrain integrates in-vitro neuronal networks with in-silico computing through high-density multi-electrode arrays (HD-MEAs). These cultured neuronal networks exhibit biologically-based adaptive intelligence within a simulated gameplay environment of 'Pong' in real-time, facilitated by closed-loop stimulation and recordings [1].

We will introduce this unique dataset and provide samples of these time series data during both gameplay and spontaneous activity, their behavioral labels in the game environment, and custom Python scripts in Jupyter notebook to analyse spiking data from 1024 channels on the HD-MEA.

The tutorial will emphasize methods for extracting meaningful insights from time series of spiking data. We will address the challenges associated with sparse, high-dimensional data and introduce custom-designed pipelines to address them. These include lower-dimensional embedding algorithms such as UMAP [3], t-SNE [4], and CEBRA [5]. Additionally, we will demonstrate the application of Gaussian kernels for data smoothing followed by community detection techniques to identify influential neurons and reduce computational complexity. These techniques will enable visualization and interpretation of complex neural activity patterns in lower-dimensional spaces, facilitating insights into network dynamics.

Finally, we will introduce methods to quantify statistical dependence and infer connectivity networks from time series, with a particular focus on transfer entropy analysis. Transfer entropy is a powerful tool for quantifying the directed information flow in complex systems [6]. Through hands-on exercises and practical demonstrations, participants will learn how to apply transfer entropy analysis using spiking or continuous data to unravel patterns of information flow in neuronal ensembles.

By the end of the tutorial, participants will have a robust toolkit for analyzing and interpreting high-dimensional, sparse spiking time series and extracting insights into neural network dynamics.

[1] Kagan BJ, Kitchen AC, Tran NT, Habibollahi F, Khajehnejad M, Parker BJ, Bhat A, Rollo B, Razi A, Friston KJ. (2022) In vitro neurons learn and exhibit sentience when embodied in a simulated game-world. *Neuron* 110(23):3952-69.

[2] Cortical Labs: <https://corticallabs.com/>

[3] McInnes, L., Healy, J., Saul, N. & Großberger, L. (2018) UMAP: Uniform Manifold Approximation and Projection for dimension reduction. *J. Open Source Softw.* 3, 86.

[4] Maaten, L. V., Postma, E. O. & Herik, J. V. (2009) Dimensionality reduction: a comparative review. *J. Mach. Learn. Res.* 10, 13.

[5] Schneider S, Lee JH, Mathis MW. (2023) Learnable latent embeddings for joint behavioural and neural analysis. *Nature* 617, 360–368.

[6] Bossomaier, T., Barnett, L., Harré, M., & Lizier, J. T. (2016). *An Introduction to Transfer Entropy*. Springer International Publishing.

Speakers:

- Forough Habibollahi: *Introduction to DishBrain: In-vitro spiking neuronal cultures in an embodied game environment*
- Moein Khajehnejad: *Lower dimensional embedding approaches to study complex network dynamics of neuronal systems*
- Leonardo Novelli: *Inferring directed statistical dependence using Transfer Entropy*

4:00pm – 5:15pm T **T09: Interactive Data Visualization Techniques** Jacarandá

Speakers: Anca Doloc-Mihu, Cengiz Gunay

Nowadays, the Web has become very prominent in computing. As a result, web standards and web programming have seen a tremendous improvement with a large number of tools becoming available at lower or no costs. Trends are visible when observing that traditional desktop applications are moving to web versions one by one. For computational neuroscience, Jupyter notebooks already provided a way to share research results on the web. However, Jupyter still requires a backend server running Python and visualizations are static and not dynamic as one would get from using web-native languages. It is possible to use some Python plugins to generate interactive visualizations, but the real power is achieved when using JavaScript directly via several expansive visualization libraries (Plotly, Vega, D3, Three.js, etc). In this tutorial, we will focus on D3.js for creating interactive visualizations for computational neuroscience applications running directly on the browser. We will review the basics for using online JavaScript notebooks (ObservableHQ) and then show more advanced examples via a hands-on tutorial format.

4:00pm – 5:15pm T **T10: Training recurrent spiking neural networks to generate experimentally recorded neural activities** Jacarandá

Speakers: Christopher Kim

Recent advances in machine learning methods enable training recurrent neural networks (RNNs) to perform highly complex and sophisticated tasks. One of the tasks, particularly interesting to neuroscientists, is to generate experimentally recorded neural activities in recurrent neural networks and study the dynamics of trained networks to investigate the underlying neural mechanism. Here, we showcase how a widely-used training method, known as recursive least squares (or FORCE), can be adopted to train spiking RNNs to reproduce spike recordings of cortical neurons. First, we give an overview of the original FORCE learning, which trains the outputs of rate-based RNNs to perform tasks, and we show how it can be modified to generate arbitrarily complex activity patterns in spiking RNNs. Using this method, we show only a subset of neurons embedded in a network of randomly connected excitatory and inhibitory spiking neurons can be trained to produce cortical activities. We demonstrate GPU implementation of the training algorithm, which enables fast training of large-scale networks, and show that the spiking activities of > 60k neurons recorded with NeuroPixel Probe can be reproduced by spiking RNNs.

Presenter:

Christopher Kim

Staff Scientist

Laboratory of Biological Modeling, NIDDK, National Institutes of Health, Bethesda, MD, USA

References

1. Kim, C. M., & Chow, C. C. (2018). Learning recurrent dynamics in spiking networks. *Elife*, 7, e37124.
2. Kim, C. M., Finkelstein, A., Chow, C. C., Svoboda, K., & Darshan, R. (2023). Distributing task-related neural activity across a cortical network through task-independent connections. *Nature Communications*, 14(1), 2851.
3. Arthur, B. J., Kim, C. M., Chen, S., Preibisch, S., & Darshan, R. (2023). A scalable implementation of the recursive least-squares algorithm for training spiking neural networks. *Frontiers in Neuroinformatics*, 17.
4. Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4), 544-557.