CNS*2025 Florence

TYPE - TUTORIALS • JULY 5 • SATURDAY

T Tutorials

JULY 5 • SATURDAY

PINNED 09:00 – 10:10	T	**Tutorial sessions**	Rooms 4, 5, 6, 9, 101, 202, 203, Belvedere, or TBA by tutorials organizers

09:00 – 12:00 T **Computational Life Science in the Cloud: Neurostimulation and Bioelectronic Medicine with o²S²PARC & Sim4Life** Room 5

Speakers: Fariba Karimi

This tutorial introduces **o²S²PARC** and **Sim4Life.web**, two powerful, cloud-based platforms for advanced problem solving in computational life sciences and it targets researchers interested in applying computational neuroscience to real-world biomedical and engineering challenges. Participants will learn how Sim4Life enables image-based and regulatory-grade simulations featuring nerve and brain network models embedded within anatomically detailed representations of the human body. They will also discover how to create, execute, and automate computational pipelines to study realistic neuromodulation scenarios with a high degree of realism, ranging from network responses to non-invasive brain stimulation (NIBS) to the interactions between medical devices and the human body for bioelectronic medicine applications.

o²S²PARC is an open-source, web-accessible platform for collaborative computational modeling, with a focus on neurosciences. It enables seamless integration of diverse computational modeling and data analysis services into pipelines, adhering to FAIR (Findable, Accessible, Interoperable, and Reusable) principles and supporting reproducibility and sustainability. Participants will explore how to use its browser-based GUI, Python API to construct modular and reusable workflows.

Sim4Life.web is a computational life sciences platform with strengths in modeling physical exposure and the resulting physiological responses in complex anatomical environments. It provides advanced multiphysics simulation capabilities, including electromagnetic (EM) and ultrasound solvers, as well as neuronal dynamics modeling. Participants will utilize Sim4Life.web to simulate and analyze the effects of neuromodulation.

Objectives

Participants in this tutorial will:

1. Understand the basics of the o²S²PARC and Sim4Life.web platforms: how to build and execute shareable computational pipelines for neuromodulation studies with the community and how to expand the computational services/functionalities for user-defined applications.
2. Discover how these tools have been applied in cutting-edge applications including:
    1. Modeling the effects of neuromodulation on large scale brain dynamics
    2. Spinal cord stimulation for injury rehabilitation
    3. Peripheral nerve stimulation for cardiac health
3. Explore realistic and advanced neurostimulation-related projects related to EM-neuro interactions in both the CNS and PNS directly within Sim4Life and o²S²PARC.

Tutorial Program

09:00 AM Intro to Image- & Cloud-Based Modeling for Real-World Applications
09:15 AM Getting Started with o²S²PARC & Sim4Life.web
09:45 AM Electromagnetic-Neuro Interactions Across Spatio-Temporal Scales (Brain, Spine, Peripheral Nervous System)
10:25 AM Coffee Break
10:45 AM Hands-On Exercises:
- Modeling the Brain Response to Non-Invasive Brain Stimulation
- Modeling Vagus Nerve Stimulation for Cardiac Health
11:45 AM Q&A and Closing Remarks

Note:

Login details to Sim4Life and o²S²PARC will be provided to all registered participants. Both platforms can be accessed directly through web browser no installation required.

Citations

[1] https://sim4life.swiss/ https://s4l-lite.io/
[2] https://osparc.io/ github
[3] https://tip-lite.science/, user manual
[4] Karimi, F., Steiner, M., Newton, T., Lloyd, B.A., Cassara, A.M., de Fontenay, P., Farcito, S., Triebkorn, J.P., Beanato, E., Wang, H. and Iavarone, E., 2025. Precision non-invasive brain stimulation: an in silico pipeline for personalized control of brain dynamics. *Journal of Neural Engineering*, *22*(2), p.026061.
DIO: https://doi.org/10.1088/1741-2552/adb88f

09:00 – 12:00 T **From single-cell modeling to large-scale network dynamics with NEST Simulator** Room 9

Moderators: Charl Linssen

For more information, please see our tutorial website at: https://clinssen.github.io/NEST-workshop/

NEST is an established, open-source simulator for spiking neuronal networks, which can capture a high degree of detail of biological network structures while retaining high performance and scalability from laptops to HPC [1]. This tutorial offers a hands-on experience in building and simulating neuron, synapse, and network models. It introduces several tools and front-ends to implement modeling ideas most effectively. Participants do not have to install software as all tools are accessible via the cloud. All parts of the tutorial are hands-on, and take place via Jupyter notebooks.
The tutorial consists of four independent parts.

We demonstrate how a tripartite connection can be included in neural simulations. We investigate the calcium dynamics in astrocytes and the effects of slow inward current on neural activity. Different connection rules are explored in order to account for spatial properties of astrocytes.

We develop a functional spiking network that can be trained to solve various tasks using an online, bio-inspired learning rule that approximates backpropagation through time: eligibility propagation (e-prop). Specifically, we use e-prop for training a network to solve a supervised classification task in which evidence needs to be accumulated over time, and generate arbitrary temporal patterns in a supervised regression task.

A neuron and synapse model are defined in NESTML that are subsequently used in a network to perform learning, prediction and replay of sequences of items, such as letters, images or sounds. The architecture learns sequences in a continuous manner: the network is exposed to repeated presentations of a given ensemble of sequences (e.g., {A,D,B,E} and {F,D,B,C}). At the beginning of the learning process, all presented sequence elements are unanticipated and do not lead to a prediction. As a consequence, the network generates mismatch signals and adjusts its synaptic strengths to minimise the prediction error.

We investigate how dendritic properties of neurons can be captured by constructing compartmental models in NEST, and using the NEAT toolbox. NEAT is a python library for the study, simulation and simplification of morphological neuron models. NEAT implements a new and powerful method to simplify morphological neuron models into compartmental models with few compartments.

Citations [1] https://nest-simulator.readthedocs.org/
[2] https://nest-desktop.readthedocs.org/
[3] Diaz-Pier S, Naveau M, Butz-Ostendorf M, Morrison A (2016). Automatic Generation of Connectivity for Large-Scale Neuronal Network Models through Structural Plasticity. Frontiers in Neuroanatomy, Vol. 10. https://doi.org/10.3389/fnana.2016.00057
[4] https://nestml.readthedocs.org/
[5] Potjans W, Morrison A, Diesmann M (2010). Enabling functional neural circuit simulations with distributed computing of neuromodulated plasticity. Frontiers in Computational Neuroscience, 4:141. DOI: https://doi.org/10.3389/fncom.2010.00141

09:00 – 12:00 T **Getting started with Cobrawap** Room 101

Speakers: Cosimo Lupo

Session description

Cobrawap (Collaborative Brain Wave Analysis Pipeline) [1,2] is an open-source, modular and customizable data analysis tool developed in the context of HBP/EBRAINS, with the aim of enabling standardized quantitative descriptions of cortical wave dynamics observed in heterogeneous data sources, both experimental and simulated. The tool intercepts the increasing demand expressed by the Neuroscience community for reusability and reproducibility, offering a software framework suitable for collecting generalized implementations of established methods and algorithms, and for embracing innovative procedures.
Inspired by FAIR principles and leveraging the latest findings in software engineering, Cobrawap is structured as a collection of modular Python3 building blocks that can be flexibly arranged along sequential stages, implementing data processing steps and analysis methods, directed by workflow managers (Snakemake or CWL). This “Getting started” tutorial [3] provides an introductory exercise to Cobrawap users, i.e. people interested in applying solutions already implemented in the software, for the exemplary scientific use-cases of i) imaging and ii) ECoG recordings from mouse cortex under anesthesia, and iii) spiking simulations from human brain. The value of this exercise goes beyond obtaining results from this specific use-case, since it allows also for “learning by doing” how the Cobrawap approach has been set up, and thus how it can be exploited in other scientific use-cases. Therefore, hints and clues will be given to people interested in developing new functions, i.e. expanding the pipeline to intercept their investigation goals and fit to their data types.

References

[1] https://github.com/NeuralEnsemble/cobrawap, https://cobrawap.readthedocs.io
[2] Gutzen, et al. (2024) https://doi.org/10.1016/j.crmeth.2023.100681
[3] https://github.com/APE-group/hands_on_cobrawap

Acknowledgments

Research co-funded by: European Union’s Horizon Europe Programme under Specific Grant Agreement No. 101147319 (EBRAINS 2.0); European Commission NextGeneration EU (PNRR EBRAINS-Italy MUR-CUP-B51E22000150006). This work is presented on behalf of the Cobrawap core team: C. Lupo, R. Gutzen, M. Denker, P. S. Paolucci, G. De Bonis. Scientific applications, that will be the subject of targeted publications in preparation, will be co-authored by additional collaborators and partners, here acknowledged: G. Gaglioti, T. Nieus, A. Pigorini, S. Sarasso, M. Massimini, C. De Luca, E. Montagni, F. Resta, A. L. Allegra Mascaro, C. Lupascu, M. Migliore.

Requirements for attendees

A personal laptop for installing the software and storing test datasets and results. It is strongly suggested to have a UNIX operative system (e.g. a Linux distribution, or Mac OS) with a recent Python distribution (e.g. >= 3.10); if usually working on Windows OS, please consider the installation of a virtual machine. Check this webpage for all the details about the requirements and some previous Cobrawap tutorials.

09:00 – 12:00 T **The Brain Scaffold Builder - reconstructions and simulations: from single cells to circuits at different levels of detail** Room 202

Speakers: Filippo Marchetti

Description:

The Brain Scaffold Builder (BSB) software is a black box component framework designed for multi-paradigm neural modeling, born to reduce the effort necessary to build neural network models.
It also provides an interface for simulate your network with popular simulators.
Throughout the sessions, attendees will learn how to use the BSB to make their own neural circuits at the level of cells. In particular, we will showcase the different tools that the software offers to reconstruct brain tissue, in terms of cell placement and connectivity. We will also demonstrate how to link neuron morphologies to their network during reconstruction, simulate the resulting model with the NEURON simulator, and analyze the results. In a similar way, we will present simulations with the NEST simulator. Finally, we will showcase how the attendees can create their custom BSB tools tailored for their use cases.

Preliminary Schedule:

09:00 – 09:15 General introduction
09:15 – 10:10 Create your first Neural Network with the BSB
10:10 – 10:40 Coffee break
10:40 – 11:20 Simulate your Neural Network with the BSB
11:20 – 12:00 Advanced reconstructions and simulations examples

09:00 – 16:00 T **Building Intuitive and Efficient Biophysical Models with Jaxley and DendroTweaks** Room 203

Speakers: Spyridon Chavlis, Michael Deistler, Kyra Kadhim, Roman Makarov

Organizers

Spyridon Chavlis, Institute of Molecular Biology and Biotechnology (IMBB), Foundation for Research and Technology-Hellas (FORTH), Heraklion, Greece

Michael Deistler, Machine Learning in Science, University of Tübingen, and Tübingen AI Center, Tübingen, Germany

Kyra L Kadhim, Tübingen AI Center, Tübingen, Germany, and Hertie Institute for AI in Brain Health, University of Tübingen, Tübingen, Germany

Roman Makarov, Institute of Molecular Biology and Biotechnology (IMBB), Foundation for Research and Technology-Hellas (FORTH), Heraklion, Greece, and Department of Biology, University of Crete, Heraklion, Greece

Description

Detailed biophysical models have been instrumental for in-depth mechanistic understanding of neuronal activity and function. However, building biophysical models is challenging: the models are complex and have many free parameters, making them hard to interpret, slow to simulate, and difficult to validate against experimental data. In this tutorial, we will showcase methods for making biophysical models more intuitive and computationally efficient, using two new software tools: the DendroTweaks toolbox [1] and the Jaxley simulator [2].

We will first learn how to create single-neuron models with detailed dendritic morphology and rich ion channel composition using the DendroTweaks toolbox in a Jupyter notebook. We will then leverage the DendroTweaks GUI to explore example models interactively and gain an intuitive understanding of how changing model parameters influences neuronal activity. In the second part of the tutorial, we will focus on how to efficiently run and optimize biophysical simulations with the GPU-based and differentiable Jaxley simulator. Participants will learn how Jaxley enables gradient-based optimization of parameters in biophysical cell and network models. They will learn how to integrate biophysical modeling with artificial intelligence to create large-scale, data-driven models that can be trained to perform tasks.

The tutorial welcomes both researchers and students interested in computational neuronal modeling, regardless of their prior experience. A basic understanding of Python programming is useful but not required.

Preliminary Schedule:

09:00 – 09:15 Welcome and general introduction
09:15 – 10:10 Building single-neuron models with DendroTweaks
10:10 – 10:40 Coffee break
10:40 – 12:00 Exploring models in DendroTweaks GUI

12:00 – 13:00 Lunch break

13:00 – 14:30 Building single-cell and network models in Jaxley
14:30 – 15:00 Coffee break
15:00 – 16:00 Optimizing biophysical models with Jaxley

Software tools:

Jaxley
DendroTweaks

References:

1. Makarov R, Chavlis S, Poirazi P (2024). eLife. DOI: https://doi.org/10.7554/eLife.103324.1.sa3
2. Deistler M, Kadhim KL, Pals M, Beck J, Huang Z, Gloeckler M, Lappalainen JK, Schröder C, Berens P, Gonçalves PJ, Macke JH DOI: https://doi.org/10.1101/2024.08.21.608979

09:00 – 16:00 T **Building mechanistic multiscale models with NEURON and NetPyNE** Room 6

Moderators: Robert Andrew McDougal
Speakers: Salvador Dura-Bernal, William W Lytton, Adam J. H. Newton
Organizers:

Robert A. McDougal • Assistant Professor • Yale University • robert.mcdougal@yale.edu
Salvador Dura-Bernal • Assistant Professor • SUNY Downstate • salvador.dura-bernal@downstate.edu
William W. Lytton • Distinguished Professor • SUNY Downstate • billl@neurosim.downstate.edu

Description:

Brain interactions occur across many temporal and spatial scales. Massive neuroscience-databasing projects throughout the world provide data at all these scales; integrating this data into a coherent picture of the brain often requires constructing models. Mechanistic multiscale modeling offers profound insights into how changes at the molecular, synaptic, morphological, and network scales—driven by development, learning, brain diseases, or pharmaceuticals—impact brain dynamics and function.

This tutorial will introduce multiscale modeling using NEURON 9.0 and NetPyNE. The tutorial will combine background, examples and hands on exercises covering the implementation of models at four key scales:

1. single neuron electrophysiology (e.g. action potential propagation),
2. intracellular dynamics (e.g. calcium buffering, protein interactions),
3. neurons in extracellular space (e.g. spreading depression), and
4. networks of neurons.

For single cell simulations, we will utilize NEURON through Python, simulating point cells and incorporating realistic morphology information from NeuroMorpho.Org.

For network simulations, we will use NetPyNE, a high-level interface to NEURON supporting both programmatic and GUI specification that facilitates the development, parallel simulation, and analysis of biophysically detailed neuronal circuits. We conclude with an example that links intracellular molecular dynamics with network spiking activity and local field potentials.

The NetPyNE tutorial will be carried out online using the Open Source Brain web platform (http://v2.opensourcebrain.org), and will include an introduction to NeuroML, a standardized computational neuroscience language to describe detailed models of neurons and neural networks.

The tutorial highlights recent substantial developments and new features in both NEURON and NetPyNE. Basic familiarity (e.g. loops, functions, variables, if) with Python is recommended. No advanced programming experience or prior knowledge of NEURON or NetPyNE is required.

Preliminary Schedule:

9:00 - 9:20 Introduction to multiscale modeling (WWL)
9:20 - 10:10 NEURON scripting (RAM)
10:10 - 10:40 Coffee break
10:40 - 12:00 Reaction-diffusion simulations (AJHN)
12:00 - 13:00 Lunch break
13:00 - 14:30 NetPyNE graphical web application (SD)
14:30 - 15:00 Coffee break
15:00 - 16:00 NetPyNE scripting (SD)

09:00 – 16:00 T **Exploring Open Brain Platform: A Collaborative Platform for Computational Neuroscience, from single cells to circuits** Room 4

Moderators: Lida Kanari

Exploring Open Brain Platform: A Collaborative Platform for Computational Neuroscience, from single cells to circuits

Open Brain Institute

Brief Description

Computational neuroscience increasingly relies on open-access data, tools, and workflows to build, simulate, and analyze brain models. The Open Brain Institute (OBI) offers a state-of-the-art platform (https://www.openbraininstitute.org) that provides a collaborative environment to drive scientific discovery, enabling researchers to work together seamlessly across disciplines. OBI streamlines computational neuroscience research by enabling standardized experiments within a controlled and consistent cloud-based infrastructure, fostering reproducibility and efficiency across the community. This tutorial will provide participants with a hands-on introduction to the core functionalities of the Open Brain Platform (OBP), including the exploration and analysis of single-cell morphological and electrophysiological datasets, the construction of biophysically detailed neuron models, and the simulation and analysis of small neural circuits. Attendees will gain practical experience navigating OBP's diverse features at both the single-cell and network levels, through powerful computational tools to accelerate their research through standardized and reproducible workflows. Finally, we will highlight the platform's expanding capabilities, including region-to-region brain interactions, AI-driven discovery for collaborative exploration, and the potential to accelerate breakthroughs in computational neuroscience.

Detailed program

9:00–9:10 **Exploration of the Open Brain Platform** (Lida Kanari)
9:10–9:55 **What makes human brains unique? Interactive topological analysis of neuronal morphologies** (Lida Kanari)
9:55–10:10 **Introduction and interactive demonstration to Single Cell Biophysical Modeling** (Darshan Mandge)
Coffee break
10:40–11:10 **The Age of Biophysics and Connectomics** (Idan Segev)
11:10–11:40 **From Spines to Synapses: Simulating the First Full EM Neuron in the Open Brain Platform** (Sapir Shapira)
11:40–11:50 **Inside Human Wiring Networks: Analyzing the Harvard EM Connectom**e (Lida Kanari)
11:50–12:20 **Interactive Demonstration: Investigating Proximal vs. Distal Dendritic Inhibition using the Open Brain Platform** (Darshan Mandge)
Lunch break
14:00–14:30 **General Biophysically-detailed Brain Models: predictions in cortical plasticity, neural coding, and spike sorting in a single model** (James Isbister)
14:30–15:00 **Modeling Human Cortical Microcircuits in Depression and Aging** (Alexandre Guet-McCreight)
15:00–15:40 **Interactive demonstration: Analysis and simulation of atlas-based cortical and hippocampal circuits on the Open Brain Platform** (James Isbister and Armando Romani)
Coffee break
16:00 Discussion

List of speakers

Alexandre Guet-McCreight, Postdoctoral Research Fellow, Krembil Centre for Neuroinformatics, CAMH
James Isbister, Scientist, Open Brain Institute, Lausanne, Switzerland
Lida Kanari, Early-Career Investigator, EPFL, Open Brain Institute, Lausanne, Switzerland
Darshan Mandge, Scientist, Open Brain Institute, Lausanne, Switzerland
Armando Romani, Early-Career Investigator, Open Brain Institute, Lausanne, Switzerland
Idan Segev, Professor, The Edmond and Lily Safra Center for Brain Sciences, the Hebrew University of Jerusalem, Israel.
Sapir Shapira, PhD Student, Hebrew University, Edmond J. Safra Campus, Jerusalem, Israel

Requirements to attend

Please bring your laptop!

Related publications

Kanari et al. 2024
Reimann et al. 2024
Isbister et al. 2023

09:00 – 16:00 T **Multiscale modeling with MOOSE** Belvedere room

Speakers: Subhasis Ray

Description

The objective of this tutorial is to familiarize participants with MOOSE (https://mooseneuro.github.io/), a comprehensive simulation platform designed for computational modeling in neuroscience and systems biology.

In addition to tutorials on biophysical and biochemical models, we shall showcase new features of MOOSE, including "jardesigner", a new web-based frontend for multiscale modeling. We will illustrate the use of jardesigner and MOOSE to build and run spiking neuron models incorporating chemical signaling for neuromodulation and synaptic plasticity on spines.

Prerequisites

- Some familiarity with basic Python programming and common system commands to run programs from the command-line is expected.

- As most of the topics will be hands-on, participants should bring their laptops.

- They should install a Python environment like Anaconda on the laptop. Participants are encouraged to create a Python environment as described in MOOSE installation instructions: https://github.com/MooseNeuro/moose-core/blob/development/docs/source/install/INSTALL.md

Tentative Schedule

09:00 - 09:40 **Multiple scales in Neuroscience**, Upinder S. Bhalla

09:40 - 10:10 **Overview of MOOSE**, Subhasis Ray

10:10 - 10:40 **Coffee Break**

10:40 - 11:25 **Setting up the software environment**, Bhanu Priya S and Subhasis Ray

11:25 - 12:10 **Abstract neuron models**, Bhanu Priya S

12:10 - 14:00 **Lunch Break**

14:00 - 14:45 **Biophysical neuron model with ion channels**, Subhasis Ray

14:45 - 15:20 **Modeling biochemical systems**, Upinder S. Bhalla

15:20 - 16:00 **Multiscale modelling and Jardesigner**, Upinder S. Bhalla

PINNED T **Tutorial sessions** Rooms 4, 5, 6, 9, 101, 202, 203, Belvedere, or TBA by tutorials organizers

10:40 – 12:00

PINNED
13:00 – 14:30

T **Tutorial sessions**

Rooms 4, 5, 6, 9, 101, 202, 203, Belvedere, or TBA by tutorials organizers

13:00 – 16:00

T **Efficient simulations of spiking neural networks using NEST GPU**

Room 9

Speakers: José Villamar

Efficient simulation of large-scale spiking neuronal networks is important for neuroscientific research, and both the simulation speed and the time it takes to instantiate the network in computer memory are key factors. In recent years, hardware acceleration through highly parallel GPUs has become increasingly popular. NEST GPU is a GPU-based simulator under the NEST Initiative written in CUDA-C++ that demonstrates high simulation speeds with models of various network sizes on single-GPU and multi-GPU systems [1,2,3].

Using a single NVIDIA RTX4090 GPU we have simulated networks on the magnitude of 80 thousand neurons and 200 million synapses with a real time factor of 0.4; and using 12000 NVIDIA A100 GPUs on the LEONARDO cluster we have managed to simulate networks on the magnitude of 3.3 billion neurons and 37 trillion synapses with a real time factor of 20.

In this showcase, we will demonstrate the capabilities of the GPU simulator and present our roadmap to integrate this technology into the ecosystem of the CPU-based simulator NEST [4].

For this, we will focus on three aspects of the simulation across model scales, namely network construction speed, state propagation speed, and energy efficiency.

Furthermore, we will present our efforts to statistically validate our simulation results against those of NEST (CPU) using established network models.

You can follow our progress through our GitHub page.

[1] Golosio et al. Front. Comput. Neurosci. 15:627620, 2021.
[2] Tiddia et al. Front. Neuroinform. 16:883333, 2022.
[3] Golosio et al. Appl. Sci. 13, 9598, 2023.
[4] Graber, S., et al. NEST 3.8 (3.8). Zenodo. 10.5281/zenodo.12624784

13:00 – 16:00 T **Graph-Based Insights into Neuronal Populations: From Traditional to Machine Learning Enhanced Causal Discovery**

Room 202

Speakers: Forough Habibollahi, Moein Khajehnejad

Speakers: **Moein Khajenejad** & **Forough Habibollahi**

The study of neuronal populations through graph and network theory provides powerful insights into the organizational principles of the brain. This half-day tutorial will focus on leveraging computational tools and methodologies to analyze neuronal activity as networks, uncovering patterns of functional and effective connectivity, and exploring causal relationships in brain circuits.

The tutorial will feature two main components:

1. Graph Analysis of Neuronal Populations

Participants will learn how to construct and analyze networks from neuronal activity data using graph-theoretic approaches. Topics will include the creation of adjacency matrices, modularity detection, and the use of metrics like clustering coefficients, centrality, and small-worldness to understand neuronal communication and information flow. Practical examples will be drawn from experimental datasets, such as in vitro spike trains from DishBrain, a pioneering system demonstrating rudimentary biological intelligence by leveraging the adaptive properties of neurons, and also from fMRI time series. This section will also discuss common pitfalls in interpreting functional connectivity data.

1. Causal Graph Discovery in Neural Systems

This segment will introduce techniques for inferring causal relationships in neuronal populations, emphasizing methods such as Granger causality and machine learning-based approaches, including Graph Neural Networks and Large Language Models. Participants will gain hands-on experience with Python tools for implementing these methods, identifying directed functional connections, and understanding the limitations and assumptions of causal inference in neuroscience.

By the end of the session, attendees will have gained practical knowledge and exposure to state-of-the-art methodologies for analyzing neuronal networks and inferring connectivity maps, equipping them to apply these tools to their research.

13:00 – 16:00 T **Modelling hippocampal neurons of animals navigating in VR with Recurrent Neural Networks** Room 5

Moderators: Marco P. Abrate

TL;DR

This tutorial demonstrates how to simulate a navigating agent (**RatInABox**), collect its visual data (**Blender**), build a Recurrent Neural Network (**RNN**) to model hippocampal-like spatial representations (**Pytorch**), and analyze its latent representations.

INTRODUCTION

The intricate interplay between sensory perception and the "cognitive map" of space (O'Keefe & Nadel, 1979) allows animals to navigate complex environments, interact with them in purposeful ways, and adapt to new situations. In the hippocampal formation, various classes of spatially modulated neurons support navigation by integrating sensory inputs to construct flexible internal models of the world (Behrens et al., 2018). Vision, in particular, plays a crucial role in guiding movement and shaping neural representations of space. However, since time is linear and irreversible, animals must rely on sequential observations to infer environmental structure. These experiences are then transformed into internal models that capture the relationships between elements in the world (Buzsáki & Tingley, 2018; Stachenfeld et al., 2017).

DESCRIPTION

This tutorial introduces computational methods for studying these processes. First, we demonstrate how to simulate the trajectories of virtual agents resembling animals (such as rodents and ferrets) with **RatInABox** and how to automatically collect visual information while agents navigate a highly customisable virtual environment using the **Blender Python API**. Next, we explore how shallow **RNNs** can be trained (using **Pytroch**) to develop biologically-plausible allocentric tuning curves in their **latent space representations**, resembling the activity of spatially selective neurons in the hippocampal formation. These models enable us to predict how specific experiences or stimuli may facilitate or hinder the emergence of spatial neurons in the hippocampus, for example in the context of development. Moreover, they can serve as surrogate models for evaluating navigational performance under different sensory conditions or targeted neuronal lesions.

13:00 – 16:00 T **Training recurrent spiking neural networks to generate experimentally recorded neural activities** Room 101

Speakers: Christopher Kim

Recent advances in machine learning methods make it possible to train 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. We first give an overview of the original FORCE learning, which trains the outputs of rate-based RNNs to perform tasks, and 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 reproduce cortical neural activities.

References:

- Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, *63*(4), 544-557.
- Kim, C. M., & Chow, C. C. (2018). Learning recurrent dynamics in spiking networks. *Elife*, *7*, e37124.
- Kim, C. M., & Chow, C. C. (2021). Training spiking neural networks in the strong coupling regime. *Neural computation*, *33*(5), 1199-1233.
- 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.

PINNED
15:00 – 16:00 T **Tutorial sessions** Rooms 4, 5, 6, 9, 101, 202, 203, Belvedere, or TBA by tutorials organizers