CNS*2023 Leipzig

TYPE - TUTORIAL • JULY 15 • SATURDAY

T Tutorial

JULY 15 • SATURDAY

09:00 – 10:10 T **T01: Building mechanistic multiscale models with NEURON, NetPyNE, and TREES** Handel

Speakers: Adam Newton, Robert McDougal, William W Lytton, Salvador Dura-Bernal, Hermann Cuntz, Thomas Künzel

Room

Handel

Organizers

Robert A McDougal, Yale University, USA
Hermann Cuntz, Ernst Strüngmann Institute (ESI) for Neuroscience, Germany
Salvador Dura-Bernal, SUNY Downstate, USA
William W Lytton, SUNY Downstate, USA

Description

Brain interactions occur across many temporal and spatial scales. Massive neuroscience-databasing projects throughout the world provide data at all these scales, data that can only be “compiled” and understood using simulation. Mechanistic multiscale modeling provides insights into how changes at molecular, synaptic, morphological, and network + areal scales; produced by development, learning, brain disease, or drugs; affect dynamics and brain function. This tutorial will introduce multiscale modeling using: the NEURON 9.0 simulator, including the Reaction-Diffusion (RxD) module and the NetPyNE tool.

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) networks of neurons. We will use NEURON through Python for single-cell simulations and introduce the new NEURON toolbox for MATLAB*. We will discuss the role of morphology and demonstrate incorporating real morphology information from NeuroMorpho.Org and synthetic morphology from the TREES toolbox. For network simulations, we will use NetPyNE, a high-level interface to NEURON supporting both programmatic and GUI specifications that facilitate 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 tutorial incorporates recent developments and new features in NEURON and NetPyNE. Basic familiarity (e.g. loops, functions, variables, if statements) with Python or MATLAB is recommended. No advanced programming experience or prior knowledge of NEURON or NetPyNE is required.

* MATLAB is a registered trademark of The MathWorks, Inc.

Schedule

- 9:00 - 9:15 - Bill Lytton - Overview: Implementing the Conceptual Model
- 9:15 - 10:10 - Robert McDougal - NEURON scripting basics
- 10:10 - 10:40 - coffee break
- 10:40 - 11:10 - Robert McDougal - Finding and reusing model components
- 11:10 - 11:30 - Robert McDougal and Thomas Künzel - NEURON Toolbox for MATLAB
- 11:30 - 12:10 - Hermann Cuntz - Morphologies and the TREES Toolbox
- 12:10 - 14:00 - lunch break
- 14:00 - 14:45 - Adam Newton - Reaction-Diffusion
- 14:45 - 15:30 - Salvador Dura-Bernal - NetPyNE intro and GUI -- slides
- 15:30 - 16:00 - coffee break
- 16:00 - 17:20 - Salvador Dura-Bernal - NetPyNE scripting

Colab notebooks

- Python intro for NEURON
- Shared colab we'll build together
- Example: current clamp showing propagating wave over space
- Example: loading morphology
- Example: parameter sweeps and temperature

- Example: mod files
- Example: synaptic input and recording spike times
- Example: voltage clamp
- Example: non-pulse current clamp
- Example: integrate and fire cell
- Example: ModelView
- Example: saving a video of a cell spiking

Software Tools

- NEURON
- NetPyNE
- TREES toolbox
- MATLAB

09:00 – 10:10 T **T02: From single-cell modeling to large-scale network dynamics with NEST Simulator** Mahler

Speakers: Sebastian Spreizer, Charl Linssen, Wenqing Wei, Jens Bruchertseifer, Pooja Babu

Tutorial webpage: https://clinssen.github.io/OCNS-2023-NEST-workshop/

Description

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; all tools are accessible via the cloud.

First, we look at NEST Desktop [2], a web-based graphical user interface (GUI) that allows the exploration of essential concepts in computational neuroscience without the need to learn a programming language. This advances both the quality and speed of teaching in computational neuroscience. We will create and analyze a balanced two-population network to get acquainted with the GUI. The tutorial will then turn to Jupyter (Python) notebooks, where we will start by creating a spiking network based on the rodent visual system. This will enable us to study the emergence
of orientation tuning of V1 neurons in the feedforward thalamocortical pathway through a nonlinear input-output transformation. Next, we will observe how homeostatic structural plasticity regulates the activity of individual neurons in a network and leads to activity-dependent self-organization [3].

To demonstrate the ease and flexibility of using NESTML to create custom neuron and synapse models for the NEST Simulator, a functional plasticity rule will be introduced into the balanced E/I network to implement a biologically realistic version of reinforcement learning. This will be done by formulating the learning model in the NESTML language syntax and using the associated toolchain to generate code for NEST [4]. In the custom plasticity rule, spike-timing-dependent plasticity will interact with a global dopamine concentration using NEST's scalable volume transmitter concept [5].

Organizers

Aadhar Sharma, Bernstein Center Freiburg, University of Freiburg, Germany
Charl Linssen, Jülich Research Centre, Germany
Jens Bruchertseifer, Trier University, Germany
Pooja Babu, Jülich Research Centre, Germany
Sebastian Spreitzer, Trier University, Germany
Wenqing Wei, Bernstein Center Freiburg, University of Freiburg, Germany

References

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. DOI: 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 – 10:10 T **T03: Modeling dendritic computations in spiking neural networks with Brian 2 and Dendrify**

Speakers: Spyridon Chavlis, Marcel Stimberg, Michalis Pagkalos Telemann

Organizers

Spyridon Chavlis, Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas,
Heraklion, Greece
Michalis Pagkalos, Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas, and Department of Biology, University of Crete, Heraklion, Greece
Marcel Stimberg, Sorbonne Université, INSERM, CNRS, Institut de la Vision, Paris, France

Description

Dendritic computations arise from the complex interplay between active ion channel conductances, neuronal morphology, and synaptic dynamics. Despite identifying many functional properties of dendrites, their roles in network operations remain to be fully explored. Complementary to experimental work, computational modeling can provide a path toward understanding the relationship between dendritic physiology and function at the network level [1]. To help with this task, the Brian 2 simulator [2] allows scientists to simply and efficiently simulate network models, providing a highly flexible system for describing almost arbitrary models in mathematical language and referring to physical units. The Dendrify package [3] is a complementary module to Brian 2, allowing for efficient and accurate simulation of neural networks consisting of neuronal models with simplified morphologies while retaining all of Brian’s advantages.

This tutorial will first introduce common mathematical models that describe essential neuronal properties, including membrane kinetics, action potential generation, and synaptic dynamics. Next, we will get familiar with Brian 2 by developing and running simulations of networks consisting of point neurons. Finally, we will showcase how Dendrify extends point neurons to reproduce accurately important passive and active dendritic properties. The tutorial will combine theoretical explanations and hands-on computational experiments. While prior experience with Brian 2 or Dendrify is not required, familiarity with Python is highly recommended.

Preliminary schedule

9:00 – 9:15 *Welcome and general introduction*
9:15 – 10:00 *Neural modelling with Brian 2*
10:00 – 10:10 *The “brian2tools” package*
10:10 – 10:40 *Coffee break*
10:40 – 11:40 *Network modelling with Brian 2*
11:40 – 12:10 *Optimizing Brian 2 models for better performance*

12:10 – 14:00 *Lunch break*

14:00 – 15:30 *Dendrify basics and Dendrify–Brian interaction*
15:30 – 16:00 *Coffee break*
16:00 - 17:20 *Model validation and example project*

Software tools

- **Brian2**
- **Dendrify**

References

1. Poirazi P, Papoutsi A (2020). Nat Rev Neurosci. DOI: 10.1038/s41583-020-0301-7
2. Stimberg M, Brette R, Goodman DF (2019). eLife. DOI: 10.7554/eLife.47314
3. Pagkalos M, Chavlis S, Poirazi P (2023). Nat Commun. DOI: 10.1038/s41467-022-35747-8

09:00 – 10:10 T **T04: Modeling astrocytes and neuron-astrocyte interactions** Schumann

Speakers: Giulio Bonifazi, Peter Bork, Audrey Denizot, Kerstin Lenk

For details, please check out the **Tutorial website: https://sites.google.com/view/astrotutorial.**

Speakers

Giulio Bonifazi, University of the Basque Country (UPV/EHU), Leioa, Spain
Audrey Denizot, AIStroSight, INRIA, Hospices Civils de Lyon, Université Claude Bernard Lyon 1, Villeurbanne, France
Peter Bork, Technical University of Denmark, Lyngby, Denmark
Kerstin Lenk, Graz University of Technology, Graz, Austria

Description

Over the past few years, a significant amount of experimental evidence has emerged highlighting the role of astrocytes in several aspects of brain function. Astrocytes are a prominent glial cell type whose processes progressively ramify, generating a dense matrix of elaborate thin protrusions. Such protrusions allow astrocytes to infiltrate the brain tissue and closely associate with the surrounding neuropil and synapses. This morphological association can be regarded as teleological evidence for the functional interaction of astrocytes and synapses. Astrocytes can modulate synaptic plasticity and neurotransmission by responding to neuronal action potentials through several mechanisms that include: transient increases in their intracellular ions, changes in cell volume, and uptake of neurotransmitters from the extracellular space. Several modeling approaches have emerged in this context, spanning multiple spatial resolutions, from the molecular to the network scale. In this tutorial, we will introduce critical aspects of such models, providing participants with the tools to include astrocytes in the modeling of brain function.

Schedule

The detailed schedule of the Tutorial can be found at **https://sites.google.com/view/astrotutorial**.

Morning sessions:
Session 1: ***"Calcium signaling in astrocytes"***
Session 2: "*Astrocyte control of cellular and extracellular volume"*

Afternoon sessions:
Session 3: "*Astrocytic tuning of synaptic activity"*
Session 4: "*Interactions of neurons and astrocytes at the network level"*

09:00 – 10:10 T **T05: A guide to reconstructing Dynamical Systems from neural measurements using Recurrent Neural Networks** Goethe

Speakers: Manuel Brenner, Daniel Durstewitz, Max Thurm

Organizers

Manuel Brenner, *Max Thurm*, *Alena Brändle*, *Lukas Eisenmann*, *Janik Fechtelpeter*, and *Florian Hess*
Department of Theoretical Neuroscience, Central Institute of Mental Health, and Faculty of Physics and Astronomy, Heidelberg University, Germany

Daniel Durstewitz
Department of Theoretical Neuroscience, Central Institute of Mental Health, and Faculty of Physics and Astronomy, Heidelberg University, Germany; Interdisciplinary Center for Scientific Computing, Heidelberg University, Germany

Description

Dynamical system reconstruction (DSR) is a powerful framework for analyzing neural data and understanding the brain's computational processes [1]. DSR is a deep learning methodology, mostly based on recurrent neural networks (RNNs), where we infer a *generative model* from neural and/ or behavioral recordings that behaves in every dynamically relevant aspect like the system underlying the data observed. Thus, after training on neural data, an RNN should be able to generate new data with the same temporal and geometrical signatures and attractor states as the underlying neural substrate.

t thereby becomes a formally accessible surrogate model for the real system which can be analyzed, simulated, and probed further. After a brief overview over various machine learning approaches for DSR, we will introduce a general framework for training a class of mathematically tractable RNN models specifically tailored to DSR. We will also discuss multimodal extensions that allow to integrate various simultaneously observed data modalities, like single-unit recordings, Ca2+ imaging, and behavioral data, into the same DSR model [3]. Practical examples will cover multiple single-unit recordings, fMRI, EEG and behavioral/ smartphone-based data [4].
Besides training RNNs for DSR on such data, we will also discuss their post-training analysis, including procedures for model validation, and for analyzing state spaces, attractor objects like fixed points, cycles, and chaos, and basins of attraction, or Lyapunov spectra.

We will provide Jupyter Notebooks and example data sets on which the methods can be tested, but encourage participants to also bring their own data!
Software tools:

https://github.com/DurstewitzLab/dendPLRNN

https://github.com/DurstewitzLab/GTF-shPLRNN
A Jupyter notebook with code examples will be made available here before the workshop begins.

References

1. Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. Reconstructing computational dynamics from neural measurements with recurrent neural networks, 2022. https://www.biorxiv.org/content/10.1101/2022.10.31.514408v1
2. Brenner, Hess, et al. (2022). Tractable Dendritic RNNs for Reconstructing Nonlinear Dynamical Systems. In International Conference on Machine Learning (pp. 2292-2320). PMLR
3. Manuel Brenner, Georgia Koppe and Daniel Durstewitz, Multimodal Teacher Forcing for Reconstructing Nonlinear Dynamical Systems, AAAI 2023 workshop paper. https://arxiv.org/abs/2212.07892
4. Koppe et al. (2019). Koppe G, Toutounji H, Kirsch P, Lis S, Durstewitz D (2019) Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI. PLOS Computational Biology 15(8): e1007263.

09:00 – 10:10 T **T06: Spectral analysis of experimental neural signals: Fundamentals and phase dynamics** Leibniz

Speakers: Axel Hutt

Organizer

Axel Hutt, INRIA Nancy Grand Est, France

Description

The spectral analysis of observed neural activity is essential in most experimental neuroscience research. To understand the meanwhile large number and types of spectral analysis techniques, it is important to understand in detail the fundamental aspects of sampling theory, Fourier analysis, and non-Fourier analysis.

The tutorial will start with sampling theory and Fourier analysis of regular and non-regular sampled data. Subsequently, it will introduce to Fourier-motivated analysis methods for non-stationary signals, e.g., different spectral power measures and different time-frequency analysis techniques (windowed Fourier transform and continuous wavelet analysis). In this context, the notion of instantaneous phase will be explained and discussed in detail (Hilbert Transform, Phase-Amplitude coupling).

The discussion of the topics will touch on theoretical aspects but will focus on more practical issues. It will be a hands-on tutorial with much information known from practice. As additional support, a large number of Python source code scripts will be provided upon request. These scripts permit the participant to implement herself/himself the different techniques and support their deep understanding.

09:00 – 10:10 T **T07: Hands-on tutorial on PyNN: A simulator-independent language for building neuronal network models** Bach

Speakers: Andrew Davison, Shailesh Appukuttan, Nitin Anisetty, Chitaranjan Mahapatra

Organizers

Chitaranjan Mahapatra, *Nitin Anisetty, Shailesh Appukuttan*, and *Andrew Davison*
Neuroinformatics Group, Paris-Saclay Institute of Neuroscience, CNRS, Université Paris-Saclay, France

Description

Computational neuroscientists have developed a diversity of software platforms for simulating neural network models at different scales. However, each simulator has its own programming or configuration language, leading to considerable difficulty porting models from one simulator to another. A common programming interface to multiple simulators would help to reduce or eliminate the problems of simulator diversity while retaining the benefits. PyNN is such an interface, making it possible to write a simulation script once, using the Python programming language, and run it without modification on any supported simulator (currently including NEURON, NEST, Brian, and the SpiNNaker and BrainScaleS neuromorphic hardware systems).

The PyNN API supports modeling at a high level of abstraction (populations of neurons, layers, columns, and the connections between them) while still allowing access to the details of individual neurons and synapses when required. PyNN provides a library of standard neuron, synapse, and synaptic plasticity models, which have been verified to work in the same way on the different supported simulators. It also provides a set of commonly used connectivity algorithms (e.g., all-to-all, random, distance-dependent, small-world) but makes it easy to provide the user’s own connectivity in a simulator-independent way. PyNN is developed as part of the NeuralEnsemble community, which promotes open-source software development in neuroscience.

In this hands-on session tutorial, participants will learn:

- How to install PyNN and the Brian 2, NEST, and NEURON simulators on Linux.
- How to build simple networks of integrate and fire neurons using PyNN.
- How to run simulation experiments with the networks using the Brian 2, NEST, and NEURON simulators.
- How to plot and visualize simulation data.

Prerequisites

Basic familiarity with the Python programming language

Useful links

- PyNN Website
- PyNN Documentation
- PyNN Download

09:00 – 10:10 T **T08: Using open tools to build efficient workflows for data access, management and analysis**

Speakers: Thomas Wachtler, Michael Denker, Reema Gupta, Moritz Kern Schiller

Tutorial Repository

On GIN - https://gin.g-node.org/CNS2023-Leipzig/CNS2023-Data-Workflows/

Overview **Session I: 9:00 -- 10:10 am**

- **9:00-9:10 Welcome, Introduction**
- **9:10-9:30 Introduction to GIN**
- **9:30-9:40 Introduction to the Dataset**
- **9:40-10:10 Primer on Neo I**

----- 10:10-10:40 COFFEE BREAK -----

Session II: 10:40--12:10 am

- 10:40-11:00 Primer on Neo II
- 11:00-11:30 Data Analysis with Elephant
- 11:30-12:10 Data Organization and Storage with NIX

Prerequisite

The tutorial is open to all registered participants, and assumes only basic experience with the Python programming language.

Preparation

In order to interactively follow the tutorials online, we suggest creating a free EBRAINS account (https://www.ebrains.eu/page/sign-up) in advance.

Tutorial Materials

Tutorial materials will be made available during the session.

Organizers

- *Reema Gupta*, Faculty of Biology, Ludwig-Maximilians-Universität München
- *Moritz Kern,* Institute of Neuroscience and Medicine (INM-6/10), Jülich Research Centre
- *Michael Denker*, Institute of Neuroscience and Medicine (INM-6/10), Jülich Research Centre
- *Thomas Wachtler*, Faculty of Biology, Ludwig-Maximilians-Universität München

Description

Neuroscientists today face challenges in managing the growing volume and complexity of data generated through rapid technological and methodological advancements and sophisticated experimental paradigms. Data management tools and methods provide indispensable solutions for researchers to efficiently handle, organize, and analyze datasets, facilitating model validation, refinement, and simulation, while fostering collaborations. This tutorial presents examples combining multiple tools synergistically into a complete digitized workflow to help researchers manage and control data and analysis processes.

- odML is an open, lightweight, and flexible format that provides a common schema (with implementations in XML, JSON, and YAML) to collect, organize and share metadata in a human- and machine-readable way.
- NIX is a lean data model and file format for storing fully annotated scientific datasets, i.e., the data together with rich metadata (odML) and their relations in a consistent, comprehensive format.
- Neo provides programmatic data objects for working with and representing electrophysiological data and can read data from many proprietary formats. In combination with NIX, Neo makes electrophysiological data interoperable with generic analysis scripts, tools, and services.
- Elephant provides a large portfolio of standard and advanced methods for analyzing data from neuronal spike trains or time series data, such as LFPs. The Neo data model makes them easily accessible to scientists and applications.
- Alpaca enables simple capture of human-readable provenance of the data processing workflow.
- GIN is a platform for version-controlled (git and git-annex) data management and collaboration. It supports any file types and folder structure, provides web and command-line access, provides an option for local installation, and services including format validation and data publication (DOI).

Background Reading

1. Grewe, J., Wachtler, T., Benda, J., 2011. A Bottom-up Approach to Data Annotation in Neurophysiology. Frontiers in Neuroinformatics 5, 16. https://doi.org/10.3389/fninf.2011.00016
2. Zehl, L., Jaillet, F., Stoewer, A., Grewe, J., Sobolev, A., Wachtler, T., Brochier, T.G., Riehle, A., Denker, M., Grün, S., 2016. Handling Metadata in a Neurophysiology Laboratory. Frontiers in Neuroinformatics 10, 26. https://doi.org/10.3389/fninf.2016.00026
3. Sprenger, J., Zehl, L., Pick, J., Sonntag, M., Grewe, J., Wachtler, T., Grün, S., Denker, M., 2019. odMLtables: A User-Friendly Approach for Managing Metadata of Neurophysiological Experiments. Front. Neuroinform. 13, 62. https://doi.org/10.3389/fninf.2019.00062
4. Brochier, T., Zehl, L., Hao, Y., Duret, M., Sprenger, J., Denker, M., Grün, S., Riehle, A., 2018. Massively parallel recordings in macaque motor cortex during an instructed delayed reach-to-grasp task. Scientific Data 5, 180055. https://doi.org/10.1038/sdata.2018.55
5. Denker, M., Grün, S., Wachtler, T., Scherberger, H., 2021. Reproducibility and efficiency in handling complex neurophysiological data. Neuroforum 27, 27–34. https://doi.org/10.1515/nf-2020-0041

10:40 – 12:10 T **T01: Building mechanistic multiscale models with NEURON, NetPyNE, and TREES** Handel

Speakers: Adam Newton, Robert McDougal, William W Lytton, Salvador Dura-Bernal, Hermann Cuntz, Thomas Künzel

Organizers

Robert A McDougal, Yale University, USA
Hermann Cuntz, Ernst Strüngmann Institute (ESI) for Neuroscience, Germany
Salvador Dura-Bernal, SUNY Downstate, USA
William W Lytton, SUNY Downstate, USA

Description

Brain interactions occur across many temporal and spatial scales. Massive neuroscience-databasing projects throughout the world provide data at all these scales, data that can only be “compiled” and understood using simulation. Mechanistic multiscale modeling provides insights into how changes at molecular, synaptic, morphological, and network + areal scales; produced by development, learning, brain disease, or drugs; affect dynamics and brain function. This tutorial will introduce multiscale modeling using: the NEURON 9.0 simulator, including the Reaction-Diffusion (RxD) module and the NetPyNE tool.

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) networks of neurons. We will use NEURON through Python for single-cell simulations and introduce the new NEURON toolbox for MATLAB*. We will discuss the role of morphology and demonstrate incorporating real morphology information from NeuroMorpho.Org and synthetic morphology from the TREES toolbox. For network simulations, we will use NetPyNE, a high-level interface to NEURON supporting both programmatic and GUI specifications that facilitate 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 tutorial incorporates recent developments and new features in NEURON and NetPyNE. Basic familiarity (e.g. loops, functions, variables, if statements) with Python or MATLAB is recommended. No advanced programming experience or prior knowledge of NEURON or NetPyNE is required.

* MATLAB is a registered trademark of The MathWorks, Inc.

Software Tools

- NEURON
- NetPyNE
- TREES toolbox
- MATLAB

10:40 – 12:10 T **T02: From single-cell modeling to large-scale network dynamics with NEST Simulator** Mahler

Speakers: Sebastian Spreizer, Charl Linssen, Wenqing Wei, Jens Bruchertseifer, Pooja Babu
Tutorial webpage: https://clinssen.github.io/OCNS-2023-NEST-workshop/

Description

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; all tools are accessible via the cloud.

First, we look at NEST Desktop [2], a web-based graphical user interface (GUI) that allows the exploration of essential concepts in computational neuroscience without the need to learn a programming language. This advances both the quality and speed of teaching in computational neuroscience. We will create and analyze a balanced two-population network to get acquainted with the GUI. The tutorial will then turn to Jupyter (Python) notebooks, where we will start by creating a spiking network based on the rodent visual system. This will enable us to study the emergence
of orientation tuning of V1 neurons in the feedforward thalamocortical pathway through a nonlinear input-output transformation. Next, we will observe how homeostatic structural plasticity regulates the activity of individual neurons in a network and leads to activity-dependent self-organization [3].

To demonstrate the ease and flexibility of using NESTML to create custom neuron and synapse models for the NEST Simulator, a functional plasticity rule will be introduced into the balanced E/I network to implement a biologically realistic version of reinforcement learning. This will be done by formulating the learning model in the NESTML language syntax and using the associated toolchain to generate code for NEST [4]. In the custom plasticity rule, spike-timing-dependent plasticity will interact with a global dopamine concentration using NEST's scalable volume transmitter concept [5].

Organizers
Aadhar Sharma, Bernstein Center Freiburg, University of Freiburg, Germany
Charl Linssen, Jülich Research Centre, Germany
Jens Bruchertseifer, Trier University, Germany
Pooja Babu, Jülich Research Centre, Germany
Sebastian Spreitzer, Trier University, Germany
Wenqing Wei, Bernstein Center Freiburg, University of Freiburg, Germany

References

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. DOI: 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

10:40 – 12:10 T **T03: Modeling dendritic computations in spiking neural networks with Brian 2 and Dendrify**

Speakers: Spyridon Chavlis, Marcel Stimberg, Michalis Pagkalos Telemann

Important note: The afternoon session will take place in Leibniz room, second floor!!

Organizers

Spyridon Chavlis, Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas,
Heraklion, Greece
Michalis Pagkalos, Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas, and Department of Biology, University of Crete, Heraklion, Greece
Marcel Stimberg, Sorbonne Université, INSERM, CNRS, Institut de la Vision, Paris, France

Description

Dendritic computations arise from the complex interplay between active ion channel conductances, neuronal morphology, and synaptic dynamics. Despite identifying many functional properties of dendrites, their roles in network operations remain to be fully explored. Complementary to experimental work, computational modeling can provide a path toward understanding the relationship between dendritic physiology and function at the network level [1]. To help with this task, the Brian 2 simulator [2] allows scientists to simply and efficiently simulate network models, providing a highly flexible system for describing almost arbitrary models in mathematical language and referring to physical units. The Dendrify package [3] is a complementary module to Brian 2, allowing for efficient and accurate simulation of neural networks consisting of neuronal models with simplified morphologies while retaining all of Brian’s advantages.

This tutorial will first introduce common mathematical models that describe essential neuronal properties, including membrane kinetics, action potential generation, and synaptic dynamics. Next, we will get familiar with Brian 2 by developing and running simulations of networks consisting of point neurons. Finally, we will showcase how Dendrify extends point neurons to reproduce accurately important passive and active dendritic properties. The tutorial will combine theoretical explanations and hands-on computational experiments. While prior experience with Brian 2 or Dendrify is not required, familiarity with Python is highly recommended.

Preliminary schedule

9:00 – 9:15	*Welcome and general introduction*
9:15 – 10:00	*Neural modelling with Brian 2*
10:00 – 10:10	*The “brian2tools” package*
10:10 – 10:40	*Coffee break*
10:40 – 11:40	*Network modelling with Brian 2*
11:40 – 12:10	*Optimizing Brian 2 models for better performance*
12:10 – 14:00	*Lunch break*
14:00 – 15:30	*Dendrify basics and Dendrify–Brian interaction*
15:30 – 16:00	*Coffee break*
16:00 - 17:20	*Model validation and example project*

Software tools

- **Brian2**
- **Dendrify**

References

1. Poirazi P, Papoutsi A (2020). Nat Rev Neurosci. DOI: 10.1038/s41583-020-0301-7
2. Stimberg M, Brette R, Goodman DF (2019). eLife. DOI: 10.7554/eLife.47314
3. Pagkalos M, Chavlis S, Poirazi P (2023). Nat Commun. DOI: 10.1038/s41467-022-35747-8

10:40 – 12:10 T **T04: Modeling astrocytes and neuron-astrocyte interactions** Schumann

Speakers: Giulio Bonifazi, Peter Bork, Audrey Denizot, Kerstin Lenk

For details, please check out the **Tutorial website: https://sites.google.com/view/astrotutorial.**

Speakers

Giulio Bonifazi, University of the Basque Country (UPV/EHU), Leioa, Spain
Audrey Denizot, AIStroSight, INRIA, Hospices Civils de Lyon, Université Claude Bernard Lyon 1, Villeurbanne, France
Peter Bork, Technical University of Denmark, Lyngby, Denmark
Kerstin Lenk, Graz University of Technology, Graz, Austria

Description

Over the past few years, a significant amount of experimental evidence has emerged highlighting the role of astrocytes in several aspects of brain function. Astrocytes are a prominent glial cell type whose processes progressively ramify, generating a dense matrix of elaborate thin protrusions. Such protrusions allow astrocytes to infiltrate the brain tissue and closely associate with the surrounding neuropil and synapses. This morphological association can be regarded as teleological evidence for the functional interaction of astrocytes and synapses. Astrocytes can modulate synaptic plasticity and neurotransmission by responding to neuronal action potentials through several mechanisms that include: transient increases in their intracellular ions, changes in cell volume, and uptake of neurotransmitters from the extracellular space. Several modeling approaches have emerged in this context, spanning multiple spatial resolutions, from the molecular to the network scale. In this tutorial, we will introduce critical aspects of such models, providing participants with the tools to include astrocytes in the modeling of brain function.

Schedule

The detailed schedule of the Tutorial can be found at **https://sites.google.com/view/astrotutorial**.

Morning sessions:
Session 1: "*Calcium signaling in astrocytes"*
Session 2: ***"Astrocyte control of cellular and extracellular volume"***

Afternoon sessions:
Session 3: "*Astrocytic tuning of synaptic activity"*
Session 4: "*Interactions of neurons and astrocytes at the network level*"

10:40 – 12:10 T **T05: A guide to reconstructing Dynamical Systems from neural measurements using Recurrent Neural Networks** Goethe

Speakers: Manuel Brenner, Daniel Durstewitz, Max Thurm

Organizers

Manuel Brenner, *Max Thurm*, *Alena Brändle*, *Lukas Eisenmann*, *Janik Fechtelpeter*, and *Florian Hess*
Department of Theoretical Neuroscience, Central Institute of Mental Health, and Faculty of Physics and Astronomy, Heidelberg University, Germany

Daniel Durstewitz
Department of Theoretical Neuroscience, Central Institute of Mental Health, and Faculty of Physics and Astronomy, Heidelberg University, Germany; Interdisciplinary Center for Scientific Computing, Heidelberg University, Germany

Description

Dynamical system reconstruction (DSR) is a powerful framework for analyzing neural data and understanding the brain's computational processes [1]. DSR is a deep learning methodology, mostly based on recurrent neural networks (RNNs), where we infer a *generative model* from neural and/ or behavioral recordings that behaves in every dynamically relevant aspect like the system underlying the data observed. Thus, after training on neural data, an RNN should be able to generate new data with the same temporal and geometrical signatures and attractor states as the underlying neural substrate.

t thereby becomes a formally accessible surrogate model for the real system which can be analyzed, simulated, and probed further. After a brief overview over various machine learning approaches for DSR, we will introduce a general framework for training a class of mathematically tractable RNN models specifically tailored to DSR. We will also discuss multimodal extensions that allow to integrate various simultaneously observed data modalities, like single-unit recordings, Ca2+ imaging, and behavioral data, into the same DSR model [3]. Practical examples will cover multiple single-unit recordings, fMRI, EEG and behavioral/ smartphone-based data [4].
Besides training RNNs for DSR on such data, we will also discuss their post-training analysis, including procedures for model validation, and for analyzing state spaces, attractor objects like fixed points, cycles, and chaos, and basins of attraction, or Lyapunov spectra.

We will provide Jupyter Notebooks and example data sets on which the methods can be tested, but encourage participants to also bring their own data!

Software tools:

https://github.com/DurstewitzLab/dendPLRNN

https://github.com/DurstewitzLab/GTF-shPLRNN
A Jupyter notebook with code examples will be made available here before the workshop begins.

References

1. Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. Reconstructing computational dynamics from neural measurements with recurrent neural networks, 2022. https://www.biorxiv.org/content/10.1101/2022.10.31.514408v1
2. Brenner, Hess, et al. (2022). Tractable Dendritic RNNs for Reconstructing Nonlinear Dynamical Systems. In International Conference on Machine Learning (pp. 2292-2320). PMLR
3. Manuel Brenner, Georgia Koppe and Daniel Durstewitz, Multimodal Teacher Forcing for Reconstructing Nonlinear Dynamical Systems, AAAI 2023 workshop paper. https://arxiv.org/abs/2212.07892
4. Koppe et al. (2019). Koppe G, Toutounji H, Kirsch P, Lis S, Durstewitz D (2019) Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI. PLOS Computational Biology 15(8): e1007263.

10:40 – 12:10 T **T06: Spectral analysis of experimental neural signals: Fundamentals and phase dynamics** Leibniz

Speakers: Axel Hutt

Organizer

Axel Hutt, INRIA Nancy Grand Est, France

Description

The spectral analysis of observed neural activity is essential in most experimental neuroscience research. To understand the meanwhile large number and types of spectral analysis techniques, it is important to understand in detail the fundamental aspects of sampling theory, Fourier analysis, and non-Fourier analysis.

The tutorial will start with sampling theory and Fourier analysis of regular and non-regular sampled data. Subsequently, it will introduce to Fourier-motivated analysis methods for non-stationary signals, e.g., different spectral power measures and different time-frequency analysis techniques (windowed Fourier transform and continuous wavelet analysis). In this context, the notion of instantaneous phase will be explained and discussed in detail (Hilbert Transform, Phase-Amplitude coupling).

The discussion of the topics will touch on theoretical aspects but will focus on more practical issues. It will be a hands-on tutorial with much information known from practice. As additional support, a large number of Python source code scripts will be provided upon request. These scripts permit the participant to implement herself/himself the different techniques and support their deep understanding.

10:40 – 12:10 T **T07: Hands-on tutorial on PyNN: A simulator-independent language for building neuronal network models** Bach

Speakers: Andrew Davison, Shailesh Appukuttan, Nitin Anisetty, Chitaranjan Mahapatra

Organizers

Chitaranjan Mahapatra, *Nitin Anisetty, Shailesh Appukuttan*, and *Andrew Davison*
Neuroinformatics Group, Paris-Saclay Institute of Neuroscience, CNRS, Université Paris-Saclay, France

Description

Computational neuroscientists have developed a diversity of software platforms for simulating neural network models at different scales. However, each simulator has its own programming or configuration language, leading to considerable difficulty porting models from one simulator to another. A common programming interface to multiple simulators would help to reduce or eliminate the problems of simulator diversity while retaining the benefits. PyNN is such an interface, making it possible to write a simulation script once, using the Python programming language, and run it without modification on any supported simulator (currently including NEURON, NEST, Brian, and the SpiNNaker and BrainScaleS neuromorphic hardware systems).

The PyNN API supports modeling at a high level of abstraction (populations of neurons, layers, columns, and the connections between them) while still allowing access to the details of individual neurons and synapses when required. PyNN provides a library of standard neuron, synapse, and synaptic plasticity models, which have been verified to work in the same way on the different supported simulators. It also provides a set of commonly used connectivity algorithms (e.g., all-to-all, random, distance-dependent, small-world) but makes it easy to provide the user's own connectivity in a simulator-independent way. PyNN is developed as part of the NeuralEnsemble community, which promotes open-source software development in neuroscience.

In this hands-on session tutorial, participants will learn:

- How to install PyNN and the Brian 2, NEST, and NEURON simulators on Linux.
- How to build simple networks of integrate and fire neurons using PyNN.
- How to run simulation experiments with the networks using the Brian 2, NEST, and NEURON simulators.
- How to plot and visualize simulation data.

Prerequisites

Basic familiarity with the Python programming language

Useful links

- PyNN Website
- PyNN Documentation
- PyNN Download

10:40 – 12:10 T **T08: Using open tools to build efficient workflows for data access, management and analysis**

Speakers: Thomas Wachtler, Michael Denker, Reema Gupta, Moritz Kern Schiller

Tutorial Repository

On GIN - https://gin.g-node.org/CNS2023-Leipzig/CNS2023-Data-Workflows/

Overview Session I: 9:00 -- 10:10 am

- 9:00-9:10 Welcome, Introduction
- 9:10-9:30 Introduction to GIN
- 9:30-9:40 Introduction to the Dataset
- 9:40-10:10 Primer on Neo I

----- 10:10-10:40 COFFEE BREAK -----

Session II: 10:40--12:10 am

- **10:40-11:00 Primer on Neo II**
- **11:00-11:30 Data Analysis with Elephant**
- **11:30-12:10 Data Organization and Storage with NIX**

Prerequisite

The tutorial is open to all registered participants, and assumes only basic experience with the Python programming language.

Preparation

In order to interactively follow the tutorials online, we suggest creating a free EBRAINS account (https://www.ebrains.eu/page/sign-up) in advance.

Tutorial Materials

Tutorial materials will be made available during the session.

Organizers

- *Reema Gupta*, Faculty of Biology, Ludwig-Maximilians-Universität München
- *Moritz Kern,* Institute of Neuroscience and Medicine (INM-6/10), Jülich Research Centre
- *Michael Denker*, Institute of Neuroscience and Medicine (INM-6/10), Jülich Research Centre
- *Thomas Wachtler*, Faculty of Biology, Ludwig-Maximilians-Universität München

Description

Neuroscientists today face challenges in managing the growing volume and complexity of data generated through rapid technological and methodological advancements and sophisticated experimental paradigms. Data management tools and methods provide indispensable solutions for researchers to efficiently handle, organize, and analyze datasets, facilitating model validation, refinement, and simulation, while fostering collaborations. This tutorial presents examples combining multiple tools synergistically into a complete digitized workflow to help researchers manage and control data and analysis processes.

- odML is an open, lightweight, and flexible format that provides a common schema (with implementations in XML, JSON, and YAML) to collect, organize and share metadata in a human- and machine-readable way.
- NIX is a lean data model and file format for storing fully annotated scientific datasets, i.e., the data together with rich metadata (odML) and their relations in a consistent, comprehensive format.
- Neo provides programmatic data objects for working with and representing electrophysiological data and can read data from many proprietary formats. In combination with NIX, Neo makes electrophysiological data interoperable with generic analysis scripts, tools, and services.
- Elephant provides a large portfolio of standard and advanced methods for analyzing data from neuronal spike trains or time series data, such as LFPs. The Neo data model makes them easily accessible to scientists and applications.
- Alpaca enables simple capture of human-readable provenance of the data processing workflow.
- GIN is a platform for version-controlled (git and git-annex) data management and collaboration. It supports any file types and folder structure, provides web and command-line access, provides an option for local installation, and services including format validation and data publication (DOI).

Background Reading

1. Grewe, J., Wachtler, T., Benda, J., 2011. A Bottom-up Approach to Data Annotation in Neurophysiology. Frontiers in Neuroinformatics 5, 16. https://doi.org/10.3389/fninf.2011.00016
2. Zehl, L., Jaillet, F., Stoewer, A., Grewe, J., Sobolev, A., Wachtler, T., Brochier, T.G., Riehle, A., Denker, M., Grün, S., 2016. Handling Metadata in a Neurophysiology Laboratory. Frontiers in Neuroinformatics 10, 26. https://doi.org/10.3389/fninf.2016.00026
3. Sprenger, J., Zehl, L., Pick, J., Sonntag, M., Grewe, J., Wachtler, T., Grün, S., Denker, M., 2019. odMLtables: A User-Friendly Approach for Managing Metadata of Neurophysiological Experiments. Front. Neuroinform. 13, 62. https://doi.org/10.3389/fninf.2019.00062
4. Brochier, T., Zehl, L., Hao, Y., Duret, M., Sprenger, J., Denker, M., Grün, S., Riehle, A., 2018. Massively parallel recordings in macaque motor cortex during an instructed delayed reach-to-grasp task. Scientific Data 5, 180055. https://doi.org/10.1038/sdata.2018.55
5. Denker, M., Grün, S., Wachtler, T., Scherberger, H., 2021. Reproducibility and efficiency in handling complex neurophysiological data. Neuroforum 27, 27–34. https://doi.org/10.1515/nf-2020-0041

14:00 – 15:30 T

T01: Building mechanistic multiscale models with NEURON, NetPyNE, and TREES

Handel

Speakers: Adam Newton, Robert McDougal, William W Lytton, Salvador Dura-Bernal, Hermann Cuntz, Thomas Künzel

Organizers

Robert A McDougal, Yale University, USA
Hermann Cuntz, Ernst Strüngmann Institute (ESI) for Neuroscience, Germany
Salvador Dura-Bernal, SUNY Downstate, USA
William W Lytton, SUNY Downstate, USA

Description

Brain interactions occur across many temporal and spatial scales. Massive neuroscience-databasing projects throughout the world provide data at all these scales, data that can only be “compiled” and understood using simulation. Mechanistic multiscale modeling provides insights into how changes at molecular, synaptic, morphological, and network + areal scales; produced by development, learning, brain disease, or drugs; affect dynamics and brain function. This tutorial will introduce multiscale modeling using: the NEURON 9.0 simulator, including the Reaction-Diffusion (RxD) module and the NetPyNE tool.

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) networks of neurons. We will use NEURON through Python for single-cell simulations and introduce the new NEURON toolbox for MATLAB*. We will discuss the role of morphology and demonstrate incorporating real morphology information from NeuroMorpho.Org and synthetic morphology from the TREES toolbox. For network simulations, we will use NetPyNE, a high-level interface to NEURON supporting both programmatic and GUI specifications that facilitate 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 tutorial incorporates recent developments and new features in NEURON and NetPyNE. Basic familiarity (e.g. loops, functions, variables, if statements) with Python or MATLAB is recommended. No advanced programming experience or prior knowledge of NEURON or NetPyNE is required.

* MATLAB is a registered trademark of The MathWorks, Inc.

Software Tools

- NEURON
- NetPyNE
- TREES toolbox
- MATLAB

14:00 – 15:30 T **T02: From single-cell modeling to large-scale network dynamics with NEST Simulator** Mahler

Speakers: Sebastian Spreizer, Charl Linssen, Wenqing Wei, Jens Bruchertseifer, Pooja Babu

Tutorial webpage: https://clinssen.github.io/OCNS-2023-NEST-workshop/

Description

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; all tools are accessible via the cloud.

First, we look at NEST Desktop [2], a web-based graphical user interface (GUI) that allows the exploration of essential concepts in computational neuroscience without the need to learn a programming language. This advances both the quality and speed of teaching in computational neuroscience. We will create and analyze a balanced two-population network to get acquainted with the GUI. The tutorial will then turn to Jupyter (Python) notebooks, where we will start by creating a spiking network based on the rodent visual system. This will enable us to study the emergence
of orientation tuning of V1 neurons in the feedforward thalamocortical pathway through a nonlinear input-output transformation. Next, we will observe how homeostatic structural plasticity regulates the activity of individual neurons in a network and leads to activity-dependent self-organization [3].

To demonstrate the ease and flexibility of using NESTML to create custom neuron and synapse models for the NEST Simulator, a functional plasticity rule will be introduced into the balanced E/I network to implement a biologically realistic version of reinforcement learning. This will be done by formulating the learning model in the NESTML language syntax and using the associated toolchain to generate code for NEST [4]. In the custom plasticity rule, spike-timing-dependent plasticity will interact with a global dopamine concentration using NEST's scalable volume transmitter concept [5].

Organizers

Aadhar Sharma, Bernstein Center Freiburg, University of Freiburg, Germany
Charl Linssen, Jülich Research Centre, Germany
Jens Bruchertseifer, Trier University, Germany
Pooja Babu, Jülich Research Centre, Germany
Sebastian Spreitzer, Trier University, Germany
Wenqing Wei, Bernstein Center Freiburg, University of Freiburg, Germany

References

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. DOI: 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

14:00 – 15:30 T

T03: Modeling dendritic computations in spiking neural networks with Brian 2 and Dendrify

Speakers: Spyridon Chavlis, Marcel Stimberg, Michalis Pagkalos

Telemann

Organizers

Spyridon Chavlis, Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas,
Heraklion, Greece
Michalis Pagkalos, Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas, and Department of Biology, University of Crete, Heraklion, Greece
Marcel Stimberg, Sorbonne Université, INSERM, CNRS, Institut de la Vision, Paris, France

Description

Dendritic computations arise from the complex interplay between active ion channel conductances, neuronal morphology, and synaptic dynamics. Despite identifying many functional properties of dendrites, their roles in network operations remain to be fully explored. Complementary to experimental work, computational modeling can provide a path toward understanding the relationship between dendritic physiology and function at the network level [1]. To help with this task, the Brian 2 simulator [2] allows scientists to simply and efficiently simulate network models, providing a highly flexible system for describing almost arbitrary models in mathematical language and referring to physical units. The Dendrify package [3] is a complementary module to Brian 2, allowing for efficient and accurate simulation of neural networks consisting of neuronal models with simplified morphologies while retaining all of Brian’s advantages.

This tutorial will first introduce common mathematical models that describe essential neuronal properties, including membrane kinetics, action potential generation, and synaptic dynamics. Next, we will get familiar with Brian 2 by developing and running simulations of networks consisting of point neurons. Finally, we will showcase how Dendrify extends point neurons to reproduce accurately important passive and active dendritic properties. The tutorial will combine theoretical explanations and hands-on computational experiments. While prior experience with Brian 2 or Dendrify is not required, familiarity with Python is highly recommended.

Preliminary schedule

9:00 – 9:15 *Welcome and general introduction*
9:15 – 10:00 *Neural modelling with Brian 2*
10:00 – 10:10 *The “brian2tools” package*
10:10 – 10:40 *Coffee break*
10:40 – 11:40 *Network modelling with Brian 2*
11:40 – 12:10 *Optimizing Brian 2 models for better performance*

12:10 – 14:00 *Lunch break*

14:00 – 15:30 *Dendrify basics and Dendrify–Brian interaction*
15:30 – 16:00 *Coffee break*
16:00 - 17:20 *Model validation and example project*

Software tools

- **Brian2**
- **Dendrify**

References

1. Poirazi P, Papoutsi A (2020). Nat Rev Neurosci. DOI: 10.1038/s41583-020-0301-7
2. Stimberg M, Brette R, Goodman DF (2019). eLife. DOI: 10.7554/eLife.47314
3. Pagkalos M, Chavlis S, Poirazi P (2023). Nat Commun. DOI: 10.1038/s41467-022-35747-8

14:00 – 15:30 T **T04: Modeling astrocytes and neuron-astrocyte interactions** Schumann

Speakers: Giulio Bonifazi, Peter Bork, Audrey Denizot, Kerstin Lenk

For details, please check out the **Tutorial website: https://sites.google.com/view/astrotutorial.**

Speakers

Giulio Bonifazi, University of the Basque Country (UPV/EHU), Leioa, Spain
Audrey Denizot, AIStroSight, INRIA, Hospices Civils de Lyon, Université Claude Bernard Lyon 1, Villeurbanne, France
Peter Bork, Technical University of Denmark, Lyngby, Denmark
Kerstin Lenk, Graz University of Technology, Graz, Austria

Description

Over the past few years, a significant amount of experimental evidence has emerged highlighting the role of astrocytes in several aspects of brain function. Astrocytes are a prominent glial cell type whose processes progressively ramify, generating a dense matrix of elaborate thin protrusions. Such protrusions allow astrocytes to infiltrate the brain tissue and closely associate with the surrounding neuropil and synapses. This morphological association can be regarded as teleological evidence for the functional interaction of astrocytes and synapses. Astrocytes can modulate synaptic plasticity and neurotransmission by responding to neuronal action potentials through several mechanisms that include: transient increases in their intracellular ions, changes in cell volume, and uptake of neurotransmitters from the extracellular space. Several modeling approaches have emerged in this context, spanning multiple spatial resolutions, from the molecular to the network scale. In this tutorial, we will introduce critical aspects of such models, providing participants with the tools to include astrocytes in the modeling of brain function.

Schedule

The detailed schedule of the Tutorial can be found at **https://sites.google.com/view/astrotutorial**.

Morning sessions:
Session 1: "*Calcium signaling in astrocytes"*
Session 2: "*Astrocyte control of cellular and extracellular volume"*

Afternoon sessions:
Session 3: ***"Astrocytic tuning of synaptic activity"***
Session 4: "*Interactions of neurons and astrocytes at the network level"*

14:00 – 15:30 T **T09: Applied Dynamical Systems Methods in the Neuroscience** Leibniz

Speakers: Richard Gast, Bastian Pietras

Organizers

Richard Gast, Northwestern University, Chicago, USA
Bastian Pietras, Universitat Pompeu Fabra, Barcelona, Spain
Hil G.E. Meijer, University of Twente, Enschede, The Netherlands

Description

Theoretical approaches to neuroscience require mathematical models to study relationships between brain structure, dynamics, and function. As such, typical problems encountered in theoretical neuroscience involve (i) the development of mathematical models of brain dynamics, (ii) finding solutions to these models, and (iii) studying parameter dependencies in these models. In this tutorial, we will give a hands-on introduction to analytical as well as numerical methods for (ii) and (iii) based on well-known examples for (I).

As a first step, we will introduce analytic methods for solving differential equation systems and studying their parameter dependencies via bifurcation theory. Using the quadratic integrate-and-fire (QIF) neuron model, we will investigate single neuron dynamics initially before moving on to network dynamics of recurrently coupled QIF neurons. At the hand of the exact mean-field equations for coupled QIF neurons [1], we introduce the general concepts behind numerical analysis techniques.

In a hands-on session, the tutorial participants will be able to apply these techniques to study the dynamics and bifurcations of networks of QIF neurons. We will provide Jupyter notebooks to showcase how to use the Python-based dynamical systems modeling tools PyRates and PyCoBi for numerical dynamical systems analysis [2].

Software tools

- PyRates
- PyCoBi
- RectiPy

Overview

Session I (90 min):

1. Basic Concepts of Dynamical Systems Theory (30 min, Hil Meijer)
2. Hands-on part 1: Simulations of Single Neuron Dynamics and Network Dynamics (30 min, Richard Gast)
3. Phase Transitions and Bifurcations in Neural Networks (30 min, Bastian Pietras)

Session II (80 min):

1. Numerical Methods for Bifurcation Analysis and Parameter Continuation (40 min, Hil Meijer)
2. Hands-on part 2: Bifurcation Analysis of the Mean-Field Equations of a Spiking Neural Network (40 min, Richard Gast)

Instructions/Preparations

Please follow the steps below to prepare your laptop for the hands-on sessions:

1. Install Docker by following the instructions for your respective operating system on: https://docs.docker.com/engine/install (FOR WINDOWS USERS: Docker for Windows tends to be very slow and some of the algorithms we will run will take much longer time on Windows. Too optimize performance, you might want to consider to follow these instructions to install Docker for Windows instead: https://www.createit.com/blog/slow-docker-on-windows-wsl2-fast-and-easy-fix-to-improve-performance/)
2. Download the docker file at https://github.com/pyrates-neuroscience/use_examples/tree/main/ubuntu_image
3. Run the command "*docker build -t tutorial_image .*" in a terminal from the directory where you placed the docker file. (If you downloaded docker desktop, make sure that you have docker desktop running.)

4. Verify that everything worked by running the command "*docker run -it tutorial_image /bin/bash"*
5. If you don’t see any error messages, you can close the terminal and you are all set!

To run the created docker image and access the tutorial material, execute the following command from a terminal:

"*docker run -it -p 8888:8888 tutorial_image /bin/bash*"

Close all your current jupyter notebook sessions.

To start a new jupyter notebook session in the same terminal, execute the following command:

"*jupyter notebook --ip 0.0.0.0 --no-browser --allow-root*"

This command should result in a URL that you can open to access the notebook.

Alternatively, you can open the URL "*localhost:8888/tree*" and copy-paste the authentification token from the provided URL in the terminal.

References

1. Montbrio et al. (Physical Review X, 2015)
2. Gast et al. (2023) arXiv

14:00 – 15:30 T **T10: Introduction to STEPS, and interaction with other computational neuroscience software for multiscale simulations** Lessing

Speakers: Stephanie Battini, Polina Shichkova, Nicola Cantarutti, Christos Kotsalos

Organizers

Christos Kotsalos, Nicola Cantarutti, Stéphanie Lisa Battini, Polina Shichkova, Alessandro Cattabiani, Tristan Carel, Pramod Kumbhar, Daniel Keller, and James Gonzalo King
Blue Brain Project, École Polytechnique Fédérale de Lausanne (EPFL), Geneva, Switzerland

Iain Hepburn, Weiliang Chen, and Jules Lallouette
Okinawa Institute of Science and Technology Graduate University (OIST), Okinawa, Japan

Description

This half-day tutorial aims to introduce users to STEPS (Stochastic Engine for Pathway Simulation), and its complex interaction with other computational neuroscience simulators, such as NEURON/Neurodamus, a blood flow solver (AstroVascPy), and a metabolism solver, in the context of multiscale simulations. STEPS is a package for exact stochastic simulation of reaction-diffusion systems in arbitrarily complex 3D geometries.

The first and theoretical part of the tutorial (~45 minutes) will be on how neuronal firing is connected to metabolism and cerebral blood flow. Blood flow in the brain is regulated by neurons and astrocytes. Increased neuronal activity elevates Ca2+ levels in astrocytes, which in turn release vasoactive metabolites from astrocyte endfeet onto blood vessels. Intracellular metabolism is an intermediate between neuronal signaling-associated energy demand and the blood supply of the energy substrates. We have coupled the neuronal electrophysiology, astrocyte endfeet, blood flow, and metabolism into a combined dynamical model. The NEURON/Neurodamus simulator drives this model implementation at the top level and uses STEPS to track extracellular concentrations of ions and other key species. The presented model and related software provide a multiscale view of the brain function and a tool to assess various biological questions.

In the practical part of the tutorial (~2 hours), we will showcase the multiscale project, where STEPS and the other simulators are coupled together to describe the complex biology of the human brain. Additionally, we will introduce STEPS, its updated, more pythonic API (API2), and its capabilities for High-Performance Computing (STEPS4 with distributed tetrahedral meshes) to cope with challenging and very large-scale simulations. The users will learn how to set up simple reaction-diffusion models, up to whole-cell length scale models of many million tetrahedra.

The participants will have the opportunity to experiment with all the simulators through the provided Docker image.

Before the tutorial, we would advise the participants to follow the instructions provided in this repository, and install all the needed software.

Finally, we will provide access to virtual machines (connect through ssh), which means that the participants can test all the simulators remotely without the need to install anything locally.

Useful links

- STEPS GitHub
- STEPS manual
- STEPS paper
- Multiscale Project Docker Image

14:00 – 15:30 T **T11: Simulation of spiking and BOLD signals using the ANNarchy simulator** Bach

Speakers: Fred Hamker, Helge Dinkelbach, Julien Vitay

Organizers

Helge Ülo Dinkelbach, *Julien Vitay*, and *Fred Hamker*
Department of Computer Science, Chemnitz University of Technology, Germany.

Description

Developing and analyzing neural networks should be as easy as possible for scientists, especially when networks increase in size and complexity. In this tutorial, we will present the neuro-simulator ANNarchy [1] intended for the simulation of rate-coded and spiking models on shared memory systems. Using an equation-based interface, ANNarchy provides a high degree of freedom for the definition of neuron and synapse models. The provided network description is transformed into C++ code for efficient computation on multi-core CPUs and GPUs.

We have recently extended the possibilities of ANNarchy by providing online BOLD monitoring tools [2]. This specialized monitor allows the generation of BOLD signals based on the spiking activity in a neural network, which can then be compared to measured fMRI data obtained from experiments.

The hands-on tutorial will first introduce the ANNarchy simulator using Colab notebooks, introducing how to create neural and synaptic models, create populations, connect them, and record their activity for later evaluation. We will then introduce the BOLD monitor and show how hyper-parameter tuning of the model parameters can be performed.

Software Tools

ANNarchy

Reference

1. Vitay, J., Dinkelbach, H.Ü. and Hamker, F.H. (2015) ANNarchy: a code generation approach to neural simulations on parallel hardware. Frontiers in Neuroinformatics 9:19. DOI:10.3389/fninf.2015.00019
2. Maith, O., Dinkelbach, H.U., Baladron, J., Vitay, J., Hamker, F.H. (2022) BOLD monitoring in the neural simulator ANNarchy. Frontiers in Neuroinformatics, 16. DOI:10.3389/fninf.2022.790966

14:00 – 15:30 T **T12: Vast parameter space exploration using L2L on EBRAINS** Goethe

Speakers: Alper Yegenoglu, Aaron Perez Martin, Sandra Diaz, Michiel van der Vlag

Organizers

- *Michiel van der Vlag*, SDL Neuroscience, Institute for Advanced Simulation, Jülich Supercomputing Centre (JSC) Jülich Research Center and JARA
- *Alper Yegenoglu*, SDL Neuroscience, Institute for Advanced Simulation, Jülich Supercomputing Centre (JSC) Jülich Research Center and JARA; and Institute of Geometry and Applied Mathematics, Department of Mathematics, RWTH Aachen University, Aachen, Germany
- *Sandra Diaz*, SDL Neuroscience, Institute for Advanced Simulation, Jülich Supercomputing Centre (JSC), Jülich Research Center and JARA

Description

This tutorial features a session on a hyper-parameter optimization framework called L2L [1], implementing the concept of learning to learn. The framework provides a selection of different optimization algorithms and makes use of multiple high-performance computing back-ends (e.g., multi-nodes, GPUs) to do vast parameter space explorations in an automated and parallel fashion (Yegenoglu et al. 2022). During this session, you will learn about the installation and use of the framework within the EBRAINS' Collaboratory [2].

Two use cases will be explored during the tutorial: 1) A task learning example in which spiking neural networks implemented in the neural spiking simulator NEST [3] are optimized to solve the mountain car problem from the OpenAI Gym environment [4]. The task encompasses a car that is placed at the bottom of a valley. The networks must find a configuration to accelerate the car to reach the goal on top of the right hill. 2) The optimization of a TVB [5] (Sanz Leon et al. 2013) simulation, which has been used in a study for the scale-integrated understanding of conscious and unconscious brain states and their mechanisms (Goldman et al. 2021). In this use case, a set of five model variables will be automatically explored to find an optimal parametrization for synchronous and asynchronous brain states.

Participants will learn how to launch the simulations on FENIX's High Performing Computing infrastructure [6]. Additionally, participants will be able to deploy jobs on CPUs and GPUs, making use of UNICORE [7] from the EBRAINS' Collaboratory.

Registration

Before course start, it is recommended to already register at the project spaces by executing these steps:

- Register an eBrains account: ebrains.eu/register/
- Register at judoor.fz-juelich.de/register
- Register project: judoor.fz-juelich.de/projects/join/training2323

Reference

1. https://github.com/Meta-optimization/L2L
2. https://wiki.ebrains.eu/
3. https://www.nest-simulator.org/
4. https://github.com/openai/gym
5. https://www.thevirtualbrain.org/
6. https://fenix.eu/
7. https://www.unicore.eu

14:00 – 15:30 T **T13: Introduction to Optical Neuroimage Studio (OptiNiSt)** Schiller

Speakers: Kenji Doya, Yukako Yamane

Caution!! Because of a flight delay, this tutorial will take place only in the last session from 16:00. Sorry for any inconvenience...

Organizers

Yukako Yamane, *Yuzhe Li*, and *Kenji Doya*
Neural Computation Unit, Okinawa Institute of Science and Technology Graduate University, Okinawa, Japan

Description

As calcium imaging continues to gain popularity, various optical imaging data processing tools for extracting cell activities from imaging data have been developed and made available as open source. Numerous methods for analyzing population-level neural activities have been published, many of which are also accessible as open source. However, selecting, installing, and testing appropriate tools to set up a data analysis pipeline is laborious for a researcher. Here we introduce OptiNiSt (Optical Neuroimage Studio) that pre-installs popular tools, including CaImAn, Suite2p, and LCCD, and allows selecting, testing, and combining multiple tools through an intuitive graphical user interface. Users can experiment with various analysis methods and parameters without writing codes or setting up different software environments. OptiNiSt is built on the standard NWB data format and APIs. Users can add original algorithms in Python as a processing module and save a constructed analysis pipeline with appropriate parameters as a Snakemake script for parallel processing and reproducibility. OptiNiSt can be installed by pip on Linux, MacOS, and Windows and is also available as a Docker container.

This tutorial showcases how to install OptiNiSt and use it for analyzing calcium imaging data, from ROI detection and neural signal extraction to the downstream analysis of neural population activities.

For this tutorial, we will use a Docker container for easy installation. Please install Docker to your computer:
https://www.docker.com/

The materials for the tutorial is available from DropBox:
https://www.dropbox.com/sh/z0awh38hpwrjchs/AABjmXTvNjrHnsO9lh-jh9fma?dl=0

Useful Links

- OptiNiSt Documentation
- OptiNiSt Download
- OptiNiSt Docker

16:00 – 17:20 T **T01: Building mechanistic multiscale models with NEURON, NetPyNE, and TREES** Handel

Speakers: Adam Newton, Robert McDougal, William W Lytton, Salvador Dura-Bernal, Hermann Cuntz, Thomas Künzel

Organizers

Robert A McDougal, Yale University, USA
Hermann Cuntz, Ernst Strüngmann Institute (ESI) for Neuroscience, Germany
Salvador Dura-Bernal, SUNY Downstate, USA
William W Lytton, SUNY Downstate, USA

Description

Brain interactions occur across many temporal and spatial scales. Massive neuroscience-databasing projects throughout the world provide data at all these scales, data that can only be “compiled” and understood using simulation. Mechanistic multiscale modeling provides insights into how changes at molecular, synaptic, morphological, and network + areal scales; produced by development, learning, brain disease, or drugs; affect dynamics and brain function. This tutorial will introduce multiscale modeling using: the NEURON 9.0 simulator, including the Reaction-Diffusion (RxD) module and the NetPyNE tool.

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) networks of neurons. We will use NEURON through Python for single-cell simulations and introduce the new NEURON toolbox for MATLAB*. We will discuss the role of morphology and demonstrate incorporating real morphology information from NeuroMorpho.Org and synthetic morphology from the TREES toolbox. For network simulations, we will use NetPyNE, a high-level interface to NEURON supporting both programmatic and GUI specifications that facilitate 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 tutorial incorporates recent developments and new features in NEURON and NetPyNE. Basic familiarity (e.g. loops, functions, variables, if statements) with Python or MATLAB is recommended. No advanced programming experience or prior knowledge of NEURON or NetPyNE is required.

* MATLAB is a registered trademark of The MathWorks, Inc.

Software Tools

- NEURON
- NetPyNE
- TREES toolbox
- MATLAB

16:00 – 17:20 T **T02: From single-cell modeling to large-scale network dynamics with NEST Simulator** Mahler

Speakers: Sebastian Spreizer, Charl Linssen, Wenqing Wei, Jens Bruchertseifer, Pooja Babu

Tutorial webpage: https://clinssen.github.io/OCNS-2023-NEST-workshop/

Description

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; all tools are accessible via the cloud.

First, we look at NEST Desktop [2], a web-based graphical user interface (GUI) that allows the exploration of essential concepts in computational neuroscience without the need to learn a programming language. This advances both the quality and speed of teaching in computational neuroscience. We will create and analyze a balanced two-population network to get acquainted with the GUI. The tutorial will then turn to Jupyter (Python) notebooks, where we will start by creating a spiking network based on the rodent visual system. This will enable us to study the emergence
of orientation tuning of V1 neurons in the feedforward thalamocortical pathway through a nonlinear input-output transformation. Next, we will observe how homeostatic structural plasticity regulates the activity of individual neurons in a network and leads to activity-dependent self-organization [3].

To demonstrate the ease and flexibility of using NESTML to create custom neuron and synapse models for the NEST Simulator, a functional plasticity rule will be introduced into the balanced E/I network to implement a biologically realistic version of reinforcement learning. This will be done by formulating the learning model in the NESTML language syntax and using the associated toolchain to generate code for NEST [4]. In the custom plasticity rule, spike-timing-dependent plasticity will interact with a global dopamine concentration using NEST's scalable volume transmitter concept [5].

Organizers

Aadhar Sharma, Bernstein Center Freiburg, University of Freiburg, Germany
Charl Linssen, Jülich Research Centre, Germany
Jens Bruchertseifer, Trier University, Germany
Pooja Babu, Jülich Research Centre, Germany
Sebastian Spreitzer, Trier University, Germany
Wenqing Wei, Bernstein Center Freiburg, University of Freiburg, Germany

References

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. DOI: 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

16:00 – 17:20 T **T03: Modeling dendritic computations in spiking neural networks with Brian 2 and Dendrify**

Speakers: Spyridon Chavlis, Marcel Stimberg, Michalis Pagkalos Telemann

Organizers

Spyridon Chavlis, Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas,
Heraklion, Greece
Michalis Pagkalos, Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas, and Department of Biology, University of Crete, Heraklion, Greece
Marcel Stimberg, Sorbonne Université, INSERM, CNRS, Institut de la Vision, Paris, France

Description

Dendritic computations arise from the complex interplay between active ion channel conductances, neuronal morphology, and synaptic dynamics. Despite identifying many functional properties of dendrites, their roles in network operations remain to be fully explored. Complementary to experimental work, computational modeling can provide a path toward understanding the relationship between dendritic physiology and function at the network level [1]. To help with this task, the Brian 2 simulator [2] allows scientists to simply and efficiently simulate network models, providing a highly flexible system for describing almost arbitrary models in mathematical language and referring to physical units. The Dendrify package [3] is a complementary module to Brian 2, allowing for efficient and accurate simulation of neural networks consisting of neuronal models with simplified morphologies while retaining all of Brian's advantages.

This tutorial will first introduce common mathematical models that describe essential neuronal properties, including membrane kinetics, action potential generation, and synaptic dynamics. Next, we will get familiar with Brian 2 by developing and running simulations of networks consisting of point neurons. Finally, we will showcase how Dendrify extends point neurons to reproduce accurately important passive and active dendritic properties. The tutorial will combine theoretical explanations and hands-on computational experiments. While prior experience with Brian 2 or Dendrify is not required, familiarity with Python is highly recommended.

Preliminary schedule

9:00 – 9:15 *Welcome and general introduction*
9:15 – 10:00 *Neural modelling with Brian 2*
10:00 – 10:10 *The "brian2tools" package*
10:10 – 10:40 *Coffee break*
10:40 – 11:40 *Network modelling with Brian 2*
11:40 – 12:10 *Optimizing Brian 2 models for better performance*

12:10 – 14:00 *Lunch break*

14:00 – 15:30 *Dendrify basics and Dendrify–Brian interaction*
15:30 – 16:00 *Coffee break*
16:00 - 17:20 *Model validation and example project*

Software tools

- **Brian2**
- **Dendrify**

References

1. Poirazi P, Papoutsi A (2020). Nat Rev Neurosci. DOI: 10.1038/s41583-020-0301-7
2. Stimberg M, Brette R, Goodman DF (2019). eLife. DOI: 10.7554/eLife.47314
3. Pagkalos M, Chavlis S, Poirazi P (2023). Nat Commun. DOI: 10.1038/s41467-022-35747-8

16:00 – 17:20 T **T04: Modeling astrocytes and neuron-astrocyte interactions** Schumann

Speakers: Giulio Bonifazi, Peter Bork, Audrey Denizot, Kerstin Lenk

For details, please check out the **Tutorial website: https://sites.google.com/view/astrotutorial.**

Speakers

Giulio Bonifazi, University of the Basque Country (UPV/EHU), Leioa, Spain
Audrey Denizot, AIStroSight, INRIA, Hospices Civils de Lyon, Université Claude Bernard Lyon 1, Villeurbanne, France
Peter Bork, Technical University of Denmark, Lyngby, Denmark
Kerstin Lenk, Graz University of Technology, Graz, Austria

Description

Over the past few years, a significant amount of experimental evidence has emerged highlighting the role of astrocytes in several aspects of brain function. Astrocytes are a prominent glial cell type whose processes progressively ramify, generating a dense matrix of elaborate thin protrusions. Such protrusions allow astrocytes to infiltrate the brain tissue and closely associate with the surrounding neuropil and synapses. This morphological association can be regarded as teleological evidence for the functional interaction of astrocytes and synapses. Astrocytes can modulate synaptic plasticity and neurotransmission by responding to neuronal action potentials through several mechanisms that include: transient increases in their intracellular ions, changes in cell volume, and uptake of neurotransmitters from the extracellular space. Several modeling approaches have emerged in this context, spanning multiple spatial resolutions, from the molecular to the network scale. In this tutorial, we will introduce critical aspects of such models, providing participants with the tools to include astrocytes in the modeling of brain function.

Schedule

The detailed schedule of the Tutorial can be found at **https://sites.google.com/view/astrotutorial**.

Morning sessions:
Session 1: "*Calcium signaling in astrocytes"*
Session 2: "*Astrocyte control of cellular and extracellular volume"*

Afternoon sessions:
Session 3: "*Astrocytic tuning of synaptic activity"*
Session 4: ***"Interactions of neurons and astrocytes at the network level"***

16:00 – 17:20 T **T09: Applied Dynamical Systems Methods in the Neuroscience** Leibniz

Speakers: Richard Gast, Bastian Pietras

Organizers

Richard Gast, Northwestern University, Chicago,USA
Bastian Pietras, University Pompeu Fabra, Barcelona, Spain
Hil G.E. Meijer, University of Twente, Enschede, The Netherlands

Description

Theoretical approaches to neuroscience require mathematical models to study relationships between brain structure, dynamics, and function. As such, typical problems encountered in theoretical neuroscience involve (i) the development of mathematical models of brain dynamics, (ii) finding solutions to these models, and (iii) studying parameter dependencies in these models. In this tutorial, we will give a hands-on introduction to analytical as well as numerical methods for (ii) and (iii) based on well-known examples for (I).

As a first step, we will introduce analytic methods for solving differential equation systems and studying their parameter dependencies via bifurcation theory. Using the quadratic integrate-and-fire (QIF) neuron model, we will investigate single neuron dynamics initially before moving on to network dynamics of recurrently coupled QIF neurons. At the hand of the exact mean-field equations for coupled QIF neurons [1], we introduce the general concepts behind numerical analysis techniques.

In a hands-on session, the tutorial participants will be able to apply these techniques to study the dynamics and bifurcations of networks of QIF neurons. We will provide Jupyter notebooks to showcase how to use the Python-based dynamical systems modeling tools PyRates and PyCoBi for numerical dynamical systems analysis [2].

Software tools

- PyRates
- PyCoBi

References

1. Montbrio etal. (2015) PRX
2. Gast et al. (2023) arXiv

16:00 – 17:20 T **T10: Introduction to STEPS, and interaction with other computational neuroscience software for multiscale simulations** Lessing

Speakers: Stephanie Battini, Polina Shichkova, Nicola Cantarutti, Christos Kotsalos

Organizers

Christos Kotsalos, Nicola Cantarutti, Stéphanie Lisa Battini, Polina Shichkova, Alessandro Cattabiani, Tristan Carel, Pramod Kumbhar, Daniel Keller, and James Gonzalo King
Blue Brain Project, École Polytechnique Fédérale de Lausanne (EPFL), Geneva, Switzerland

Iain Hepburn, Weiliang Chen, and Jules Lallouette
Okinawa Institute of Science and Technology Graduate University (OIST), Okinawa, Japan

Description

This half-day tutorial aims to introduce users to STEPS (Stochastic Engine for Pathway Simulation), and its complex interaction with other computational neuroscience simulators, such as NEURON/Neurodamus, a blood flow solver (AstroVascPy), and a metabolism solver, in the context of multiscale simulations. STEPS is a package for exact stochastic simulation of reaction-diffusion systems in arbitrarily complex 3D geometries.

The first and theoretical part of the tutorial (~45 minutes) will be on how neuronal firing is connected to metabolism and cerebral blood flow. Blood flow in the brain is regulated by neurons and astrocytes. Increased neuronal activity elevates Ca2+ levels in astrocytes, which in turn release vasoactive metabolites from astrocyte endfeet onto blood vessels. Intracellular metabolism is an intermediate between neuronal signaling-associated energy demand and the blood supply of the energy substrates. We have coupled the neuronal electrophysiology, astrocyte endfeet, blood flow, and metabolism into a combined dynamical model. The NEURON/Neurodamus simulator drives this model implementation at the top level and uses STEPS to track extracellular concentrations of ions and other key species. The presented model and related software provide a multiscale view of the brain function and a tool to assess various biological questions.

In the practical part of the tutorial (~2 hours), we will showcase the multiscale project, where STEPS and the other simulators are coupled together to describe the complex biology of the human brain. Additionally, we will introduce STEPS, its updated, more pythonic API (API2), and its capabilities for High-Performance Computing (STEPS4 with distributed tetrahedral meshes) to cope with challenging and very large-scale simulations. The users will learn how to set up simple reaction-diffusion models, up to whole-cell length scale models of many million tetrahedra.

The participants will have the opportunity to experiment with all the simulators through the provided Docker image.

Before the tutorial, we would advise the participants to follow the instructions provided in this repository, and install all the needed software.

Finally, we will provide access to virtual machines (connect through ssh), which means that the participants can test all the simulators remotely without the need to install anything locally.

Useful links

- STEPS GitHub
- STEPS manual
- STEPS paper
- Multiscale Project Docker Image

16:00 – 17:20 T **T11: Simulation of spiking and BOLD signals using the ANNarchy simulator** Bach

Speakers: Fred Hamker, Helge Dinkelbach, Julien Vitay

Organizers

Helge Ülo Dinkelbach, *Julien Vitay*, and *Fred Hamker*
Department of Computer Science, Chemnitz University of Technology, Germany.

Description

Developing and analyzing neural networks should be as easy as possible for scientists, especially when networks increase in size and complexity. In this tutorial, we will present the neuro-simulator ANNarchy [1] intended for the simulation of rate-coded and spiking models on shared memory systems. Using an equation-based interface, ANNarchy provides a high degree of freedom for the definition of neuron and synapse models. The provided network description is transformed into C++ code for efficient computation on multi-core CPUs and GPUs.

We have recently extended the possibilities of ANNarchy by providing online BOLD monitoring tools [2]. This specialized monitor allows the generation of BOLD signals based on the spiking activity in a neural network, which can then be compared to measured fMRI data obtained from experiments.

The hands-on tutorial will first introduce the ANNarchy simulator using Colab notebooks, introducing how to create neural and synaptic models, create populations, connect them, and record their activity for later evaluation. We will then introduce the BOLD monitor and show how hyper-parameter tuning of the model parameters can be performed.

Software Tools

ANNarchy

Reference

1. Vitay, J., Dinkelbach, H.Ü. and Hamker, F.H. (2015) ANNarchy: a code generation approach to neural simulations on parallel hardware. Frontiers in Neuroinformatics 9:19. DOI:10.3389/fninf.2015.00019
2. Maith, O., Dinkelbach, H.U., Baladron, J., Vitay, J., Hamker, F.H. (2022) BOLD monitoring in the neural simulator ANNarchy. Frontiers in Neuroinformatics, 16. DOI:10.3389/fninf.2022.790966

16:00 – 17:20 T **T12: Vast parameter space exploration using L2L on EBRAINS** Goethe

Speakers: Alper Yegenoglu, Aaron Perez Martin, Sandra Diaz, Michiel van der Vlag

Organizers

- *Michiel van der Vlag*, SDL Neuroscience, Institute for Advanced Simulation, Jülich Supercomputing Centre (JSC) Jülich Research Center and JARA
- *Alper Yegenoglu*, SDL Neuroscience, Institute for Advanced Simulation, Jülich Supercomputing Centre (JSC) Jülich Research Center and JARA; and Institute of Geometry and Applied Mathematics, Department of Mathematics, RWTH Aachen University, Aachen, Germany
- *Sandra Diaz*, SDL Neuroscience, Institute for Advanced Simulation, Jülich Supercomputing Centre (JSC), Jülich Research Center and JARA

Description

This tutorial features a session on a hyper-parameter optimization framework called L2L [1], implementing the concept of learning to learn. The framework provides a selection of different optimization algorithms and makes use of multiple high-performance computing back-ends (e.g., multi-nodes, GPUs) to do vast parameter space explorations in an automated and parallel fashion (Yegenoglu et al. 2022). During this session, you will learn about the installation and use of the framework within the EBRAINS' Collaboratory [2].

Two use cases will be explored during the tutorial: 1) A task learning example in which spiking neural networks implemented in the neural spiking simulator NEST [3] are optimized to solve the mountain car problem from the OpenAI Gym environment [4]. The task encompasses a car that is placed at the bottom of a valley. The networks must find a configuration to accelerate the car to reach the goal on top of the right hill. 2) The optimization of a TVB [5] (Sanz Leon et al. 2013) simulation, which has been used in a study for the scale-integrated understanding of conscious and unconscious brain states and their mechanisms (Goldman et al. 2021). In this use case, a set of five model variables will be automatically explored to find an optimal parametrization for synchronous and asynchronous brain states.

Participants will learn how to launch the simulations on FENIX's High Performing Computing infrastructure [6]. Additionally, participants will be able to deploy jobs on CPUs and GPUs, making use of UNICORE [7] from the EBRAINS' Collaboratory.

Reference

1. https://github.com/Meta-optimization/L2L
2. https://wiki.ebrains.eu/
3. https://www.nest-simulator.org/
4. https://github.com/openai/gym
5. https://www.thevirtualbrain.org/
6. https://fenix.eu/
7. https://www.unicore.eu

16:00 – 17:20 T **T13: Introduction to Optical Neuroimage Studio (OptiNiSt)** Schiller

Speakers: Kenji Doya, Yukako Yamane

Organizers

Yukako Yamane, *Yuzhe Li*, and *Kenji Doya*
Neural Computation Unit, Okinawa Institute of Science and Technology Graduate University, Okinawa, Japan

Description

As calcium imaging continues to gain popularity, various optical imaging data processing tools for extracting cell activities from imaging data have been developed and made available as open source. Numerous methods for analyzing population-level neural activities have been published, many of which are also accessible as open source. However, selecting, installing, and testing appropriate tools to set up a data analysis pipeline is laborious for a researcher. Here we introduce OptiNiSt (Optical Neuroimage Studio) that pre-installs popular tools, including CaImAn, Suite2p, and LCCD, and allows selecting, testing, and combining multiple tools through an intuitive graphical user interface. Users can experiment with various analysis methods and parameters without writing codes or setting up different software environments. OptiNiSt is built on the standard NWB data format and APIs. Users can add original algorithms in Python as a processing module and save a constructed analysis pipeline with appropriate parameters as a Snakemake script for parallel processing and reproducibility. OptiNiSt can be installed by pip on Linux, MacOS, and Windows and is also available as a Docker container.

This tutorial showcases how to install OptiNiSt and use it for analyzing calcium imaging data, from ROI detection and neural signal extraction to the downstream analysis of neural population activities.

For this tutorial, we will use a Docker container for easy installation. Please install Docker to your computer:
https://www.docker.com/

The materials for the tutorial is available from DropBox:
https://www.dropbox.com/sh/z0awh38hpwrjchs/AABjmXTvNjrHnsO9lh-jh9fma?dl=0

Useful Links

- OptiNiSt Documentation
- OptiNiSt Download
- OptiNiSt Docker