NEST+NESTML Tutorial

NICE Conference 2025 – Heidelberg
24.–28. March

Charl Linssen | Sebastian Spreizer | Dennis Terhorst

Our plan for today

1. **Introduction to NEST** (Dennis)
    - Hands-on: NEST Desktop (Sebastian)
2. **Introduction to eprop in NEST** (Charl)
    - Hands-on: Learning a sine wave
3. **Introduction to NESTML** (Charl)
    - Hands-on:
        - create custom neuron model
        - create custom synaptic plasticity rule

(120 min)

NEST Simulator

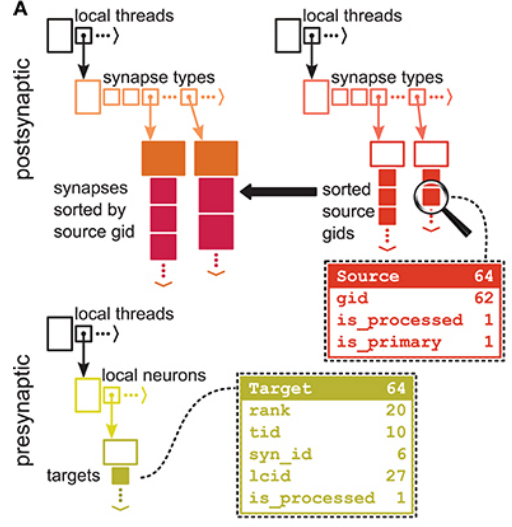
- 25+ years of experience
- 120+ developers
- Model-independent simulation engine
- Python based user interface, C++ kernel
- Multi-threading and multi-processing to use machines efficiently
- Laptop to super computers

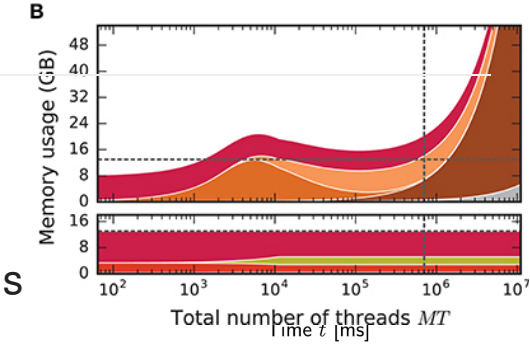
NEST and friends

- NEST Simulator – main engine
  - pyNEST – Python interface
- NESTML – Modeling language
- NEST Desktop – teaching front-end
- PyNN – simulator independant language
- MUSIC – interface to other simulators
- NEAT – bridging to compartmental models
- …

Why is NEST relevant for neuromorphic?

- Hardware–software co-design
- Synergies for solutions
- Competition between technologies
- Verification and validation





Why NEST – flexible preview of neuroscience

- exchangable neuron/synapse/network models
  → learn about explored mechanisms
  - gap junctions
  - astrocytes
  - …
- investigate requirement limits, for example [1]
  - weight resolution
  - update timesteps

[1] Pfeil et al., Is a 4-bit synaptic weight resolution enough? - constraints on enabling spike-timing dependent plasticity in neuromorphic hardware. – 10.3389/fnins.2012.00090

Why is NEST relevant for neuromorphic?

- Hardware–software co-design
- Synergies for solutions
- Competition between technologies
- Verification and validation

👉 “build the thing right” and *“build the right thing”*

Why NEST – flexible preview of neuroscience

- exchangable neuron/synapse/network models
  → learn about explored mechanisms
  - gap junctions
  - astrocytes
  - …
- investigate requirement limits, for example [1]
  - weight resolution
  - update timesteps

[1] Pfeil et al., Is a 4-bit synaptic weight resolution enough? - constraints on enabling spike-timing dependent plasticity in neuromorphic hardware. – 10.3389/fnins.2012.00090

Why NEST – co-design: different world, similar solutions

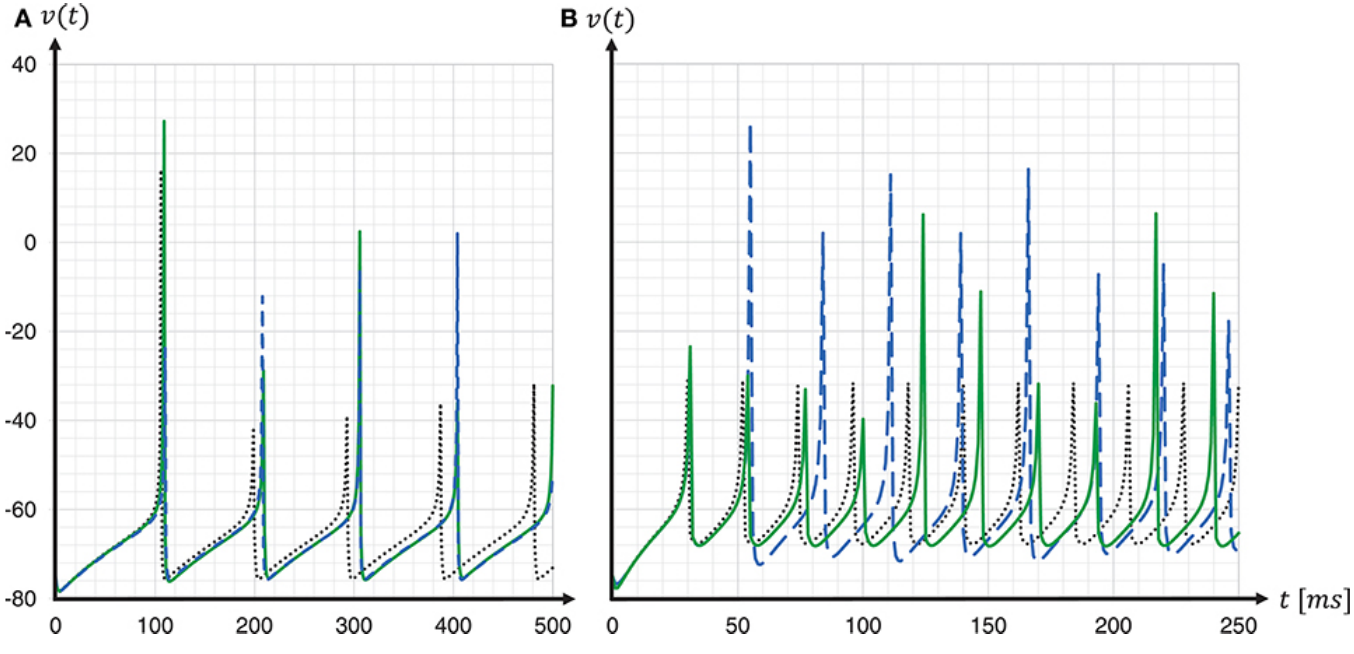
- model implementations
- architectures for handling data
- communication structures
- memory bottlenecks

Why NEST – cross-check from different perspective

- cross-validation and verification
  - mathematical models
  - software
  - hardware

Verification – example 1: Spike timing

With and without fixed-point data type conversion



Trensch et al., Rigorous Neural Network Simulations: A Model Substantiation Methodology for Increasing the Correctness of Simulation Results in the Absence of Experimental Validation Data – 10.3389/fninf.2018.00081

Verification – example 2: Integration method

Comparison of simulation engines

- numerical solver can be biassed, leading to different firing rates for certain types of models

Plesser, Commentary: Accelerating spiking neural network simulations with PymoNNto and PymoNNtorch – 10.3389/fninf.2024.1446620

Memory & Communication – example 3: NEST kernel

Example

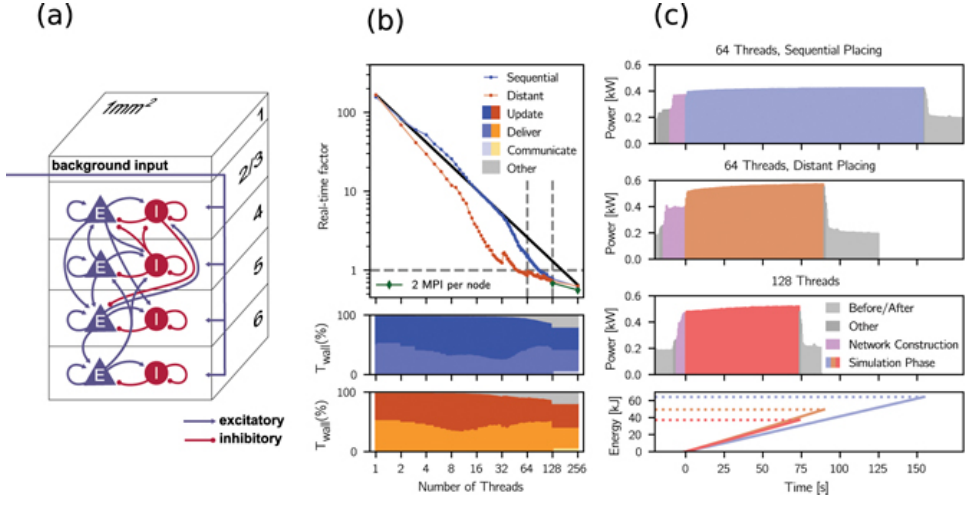
- internal division of communication strucutres
- minimized required synchronization points

Figure form: Jordan et al., Extremely Scalable Spiking Neuronal Network Simulation Code: From Laptops to Exascale Computers – 10.3389/fninf.2018.00002

Validation – example 4: Time and energy

- Is NC really faster?
- Is NC really more energy-efficient?

Microcircuit (PD14) as standard benchmark leading to comparability and constructive competition.



RTF	$E_{\mathrm{syn-event}}$ $(\mu\mathrm{J})$	References
6.29	4.39	2018, NEST [2]
2.47	9.35	2018, NEST [2]
26.08	0.30	2018, GeNN [3]
1.84	0.47 [a]	2018, GeNN [3]
1.00	0.60	2019, SpiNNaker [8]
1.06	—	2021, NeuronGPU [9]
0.70	—	2021, GeNN [10]
0.67	0.33	NEST, AMD EPYC Rome (one node, 2 MPI)
0.53	0.48	NEST, AMD EPYC Rome (two nodes, 4 MPI)

Kurth et al., Sub-realtime simulation of a neuronal network of natural density – 10.1088/2634-4386/ac55fc

Why NEST – Summary

- offers a lot of opportunities and experience in an active and interested community
- established scalable platform also for reference implementations of future hardware

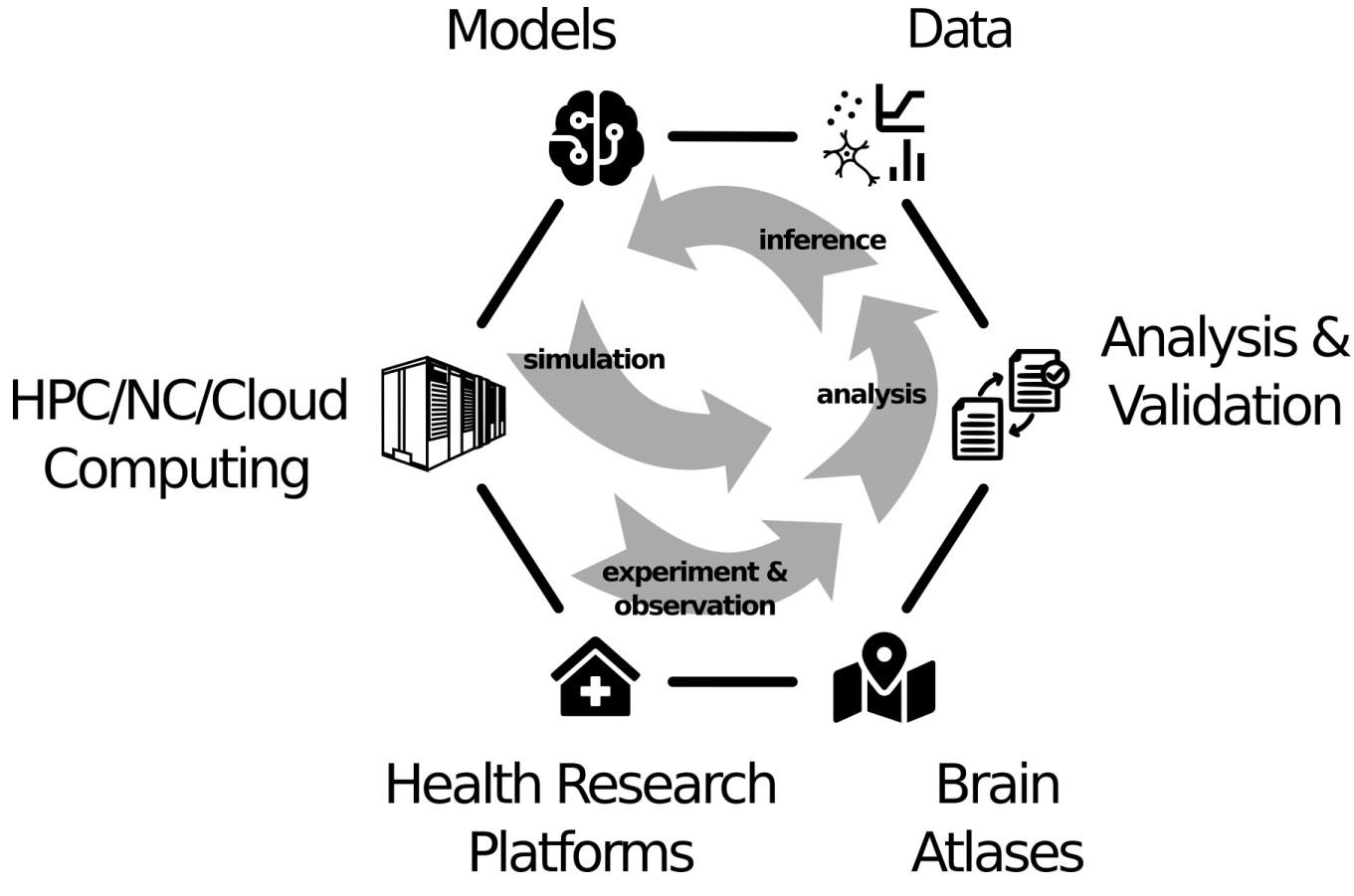
EBRAINS

https://ebrains.eu/register

EBRAINS

- successor of HBP
- digital research infrastructure
- provides interoperable tools, services and resources
- curated ecosystem
- https://ebrains.eu/register

EBRAINS


Models
Data
inference
simulation
analysis
HPC/NC/Cloud Computing
Analysis & Validation
experiment & observation
Health Research Platforms
Brain Atlases

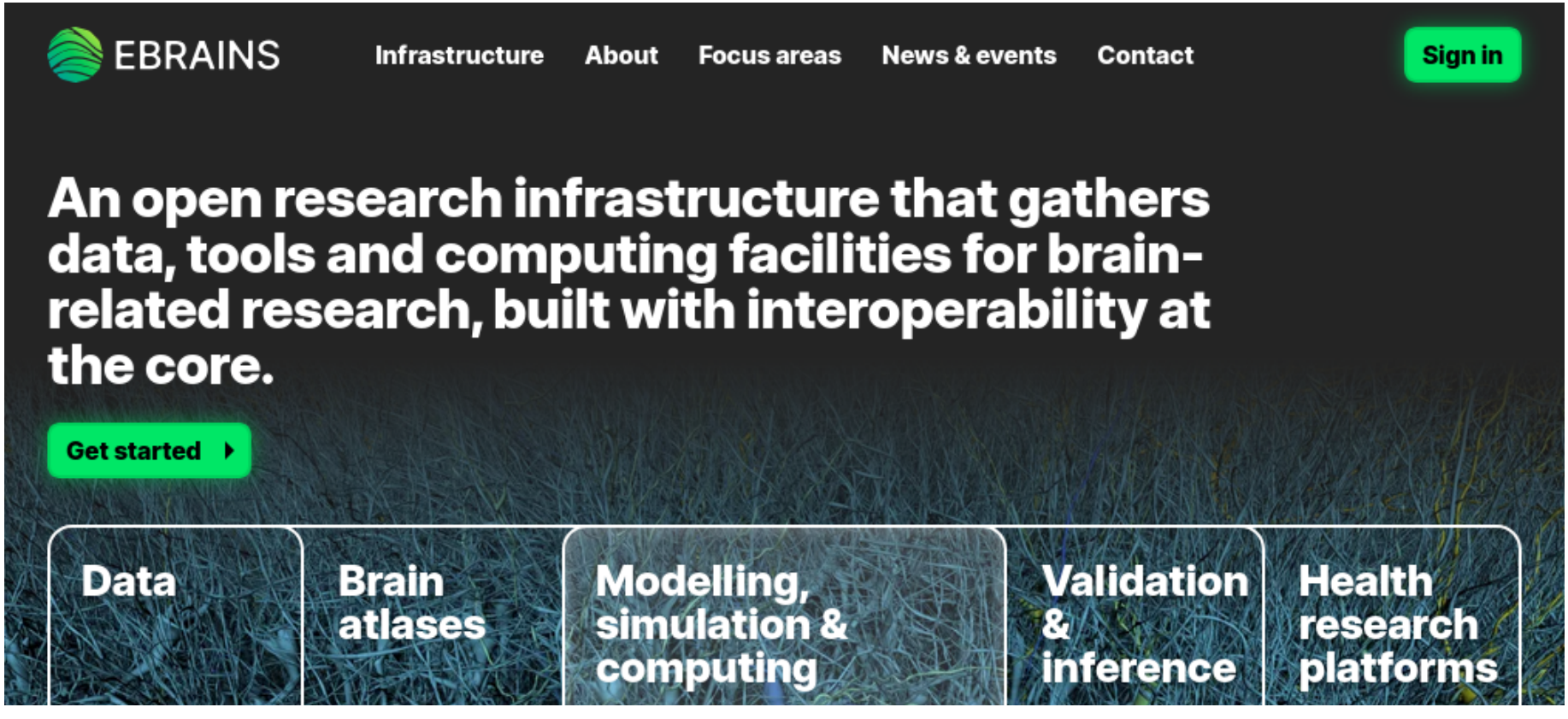
EBRAINS

Technical components

- Collaboration: knowledge graph, drive, wiki, …
- Compute: Jupyter lab, HPC, cloud
- Storage: data, atlas, …
- Software: EBRAINS Software Disistribution
- Services: gitlab, notes, office, …

Single sign-on.

https://ebrains.eu

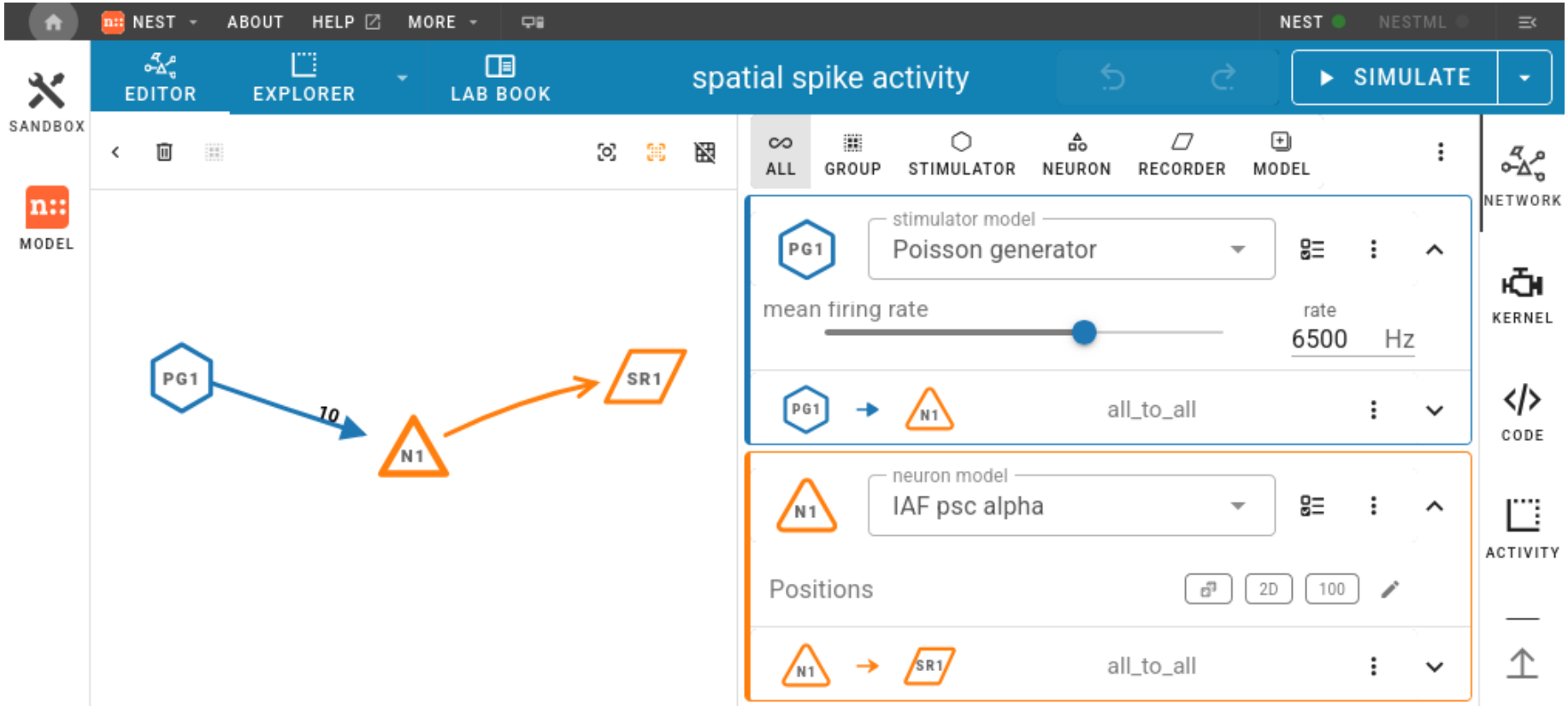


NEST Desktop

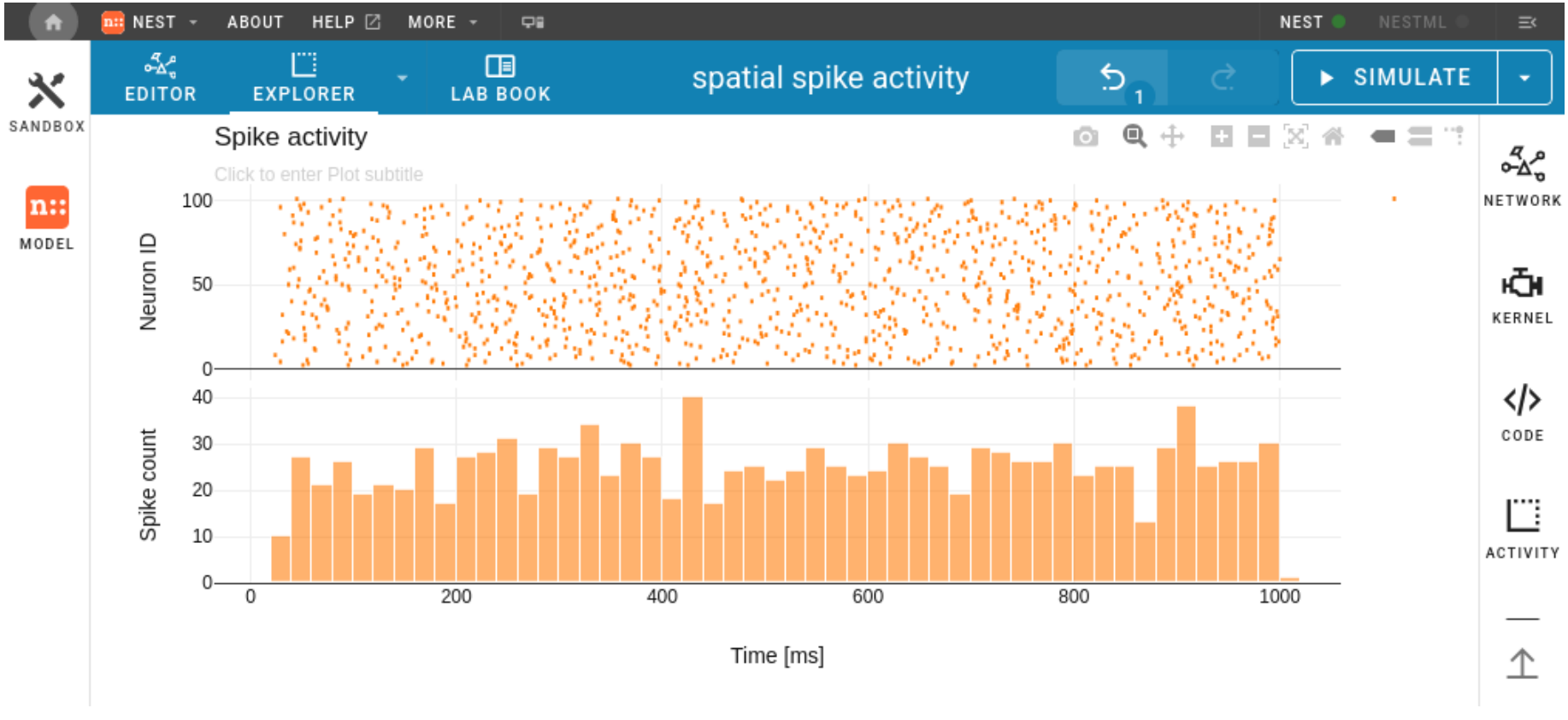
- originally for teaching
- low threshold, graphical, web-based
- quick prototyping
- simple analyses

https://nest-desktop.apps.ebrains.eu

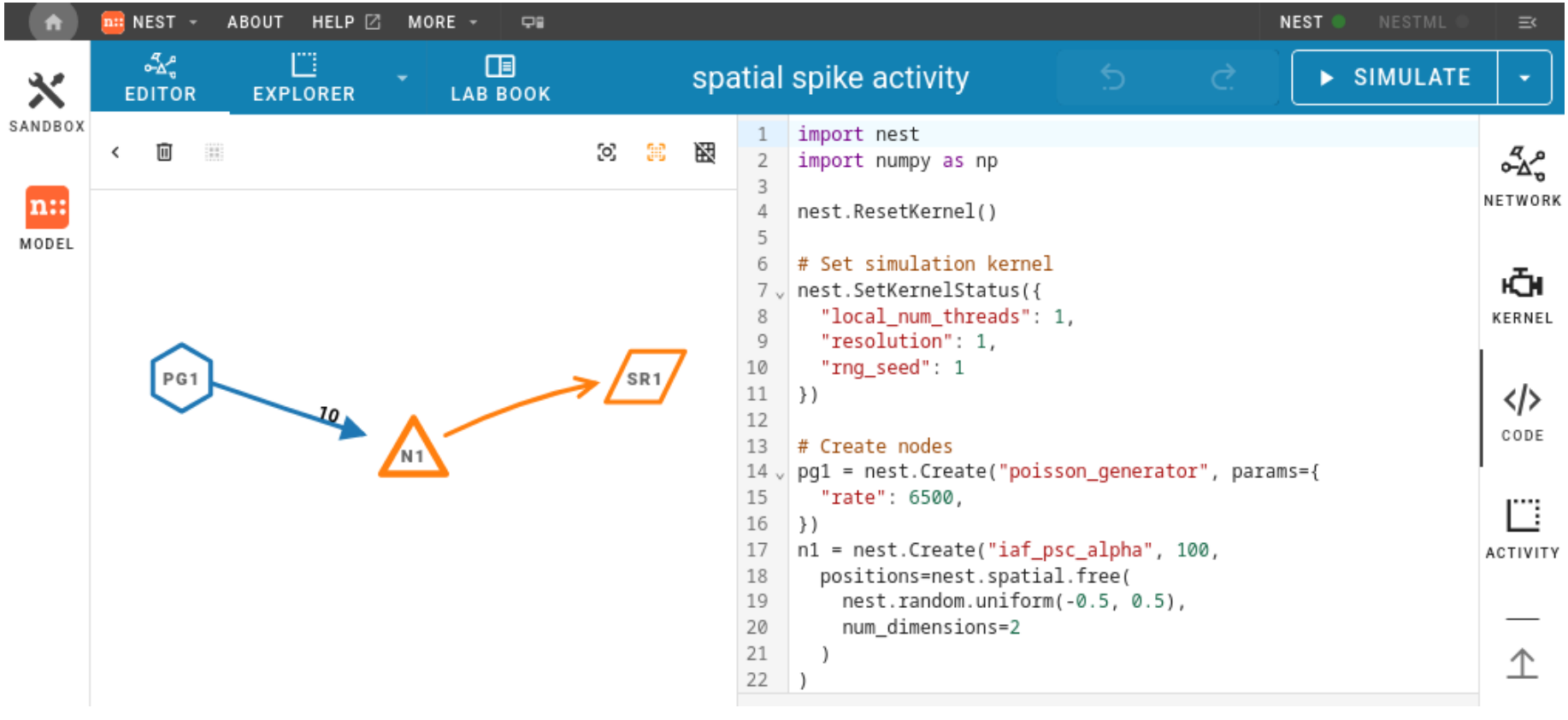
NEST Desktop - Construct network models



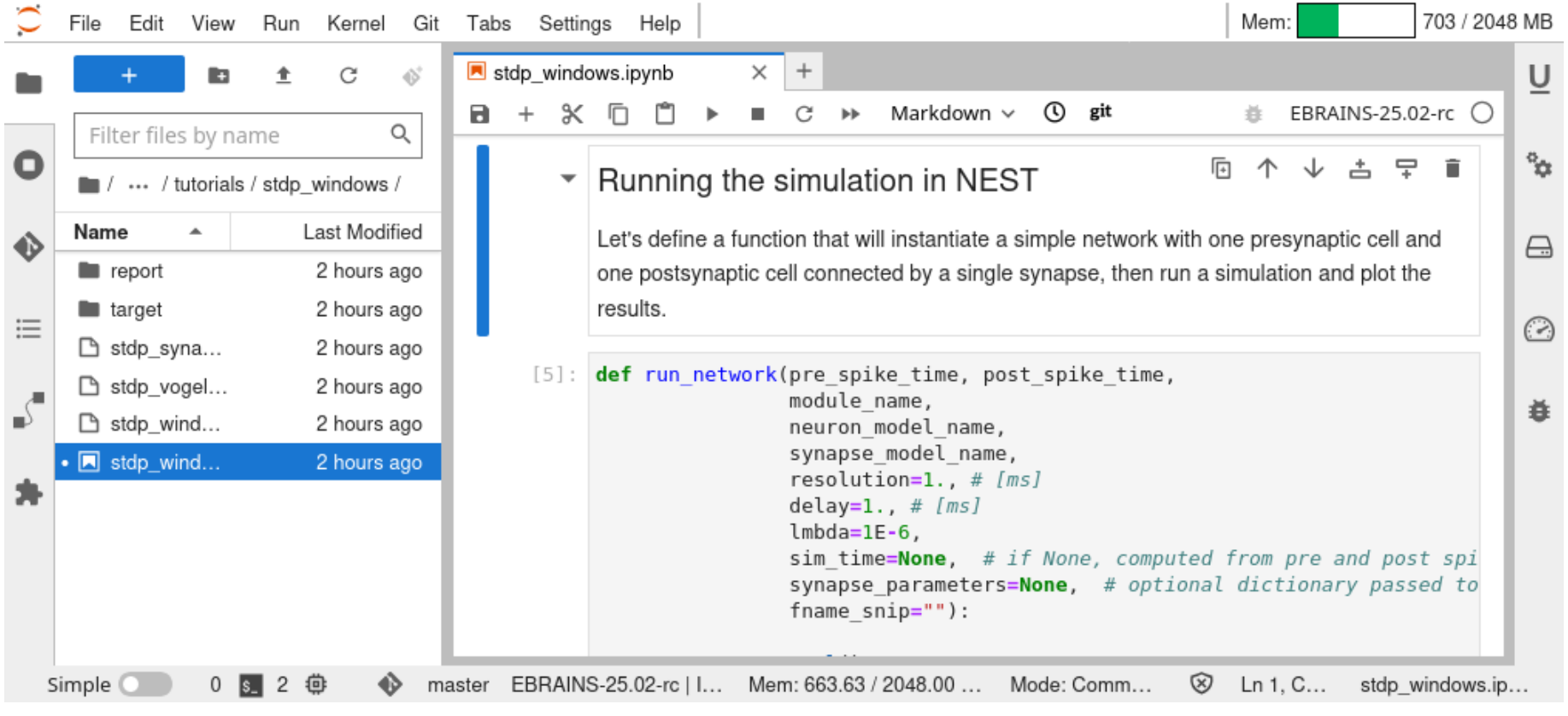
NEST Desktop - Simulated activity



NEST Desktop – Code export



EBRAINS Jupyter



https://lab.ebrains.eu – All tools installed and tested.

Hands on 👋

Getting started with NEST

- ➡️ NEST Desktop: https://nest-desktop.apps.ebrains.eu
  - create balanced networks; export to Python script
- ➡️ EBRAINS Jupyter: https://lab.ebrains.eu
  - run same simulation; add more analysis

e–prop in NEST: Bridging to machine learning

< Charl will present >

look for Korcsak-Gorzo and Espinoza Valverde

Hands on 👋

Lot's of cool stuff!

- ➡️ NEST examples: https://nest-simulator.org/documentation
  - ➡️ *E-prop plasticity examples: "generating sine waves"*

⇒Try it on  EBRAINS

NESTML

nest::ml

Search docs

- NESTML language documentation
- Installing NESTML
- Running NESTML
- Models library
- Tutorials
- Extending NESTML
- Getting help
- Citing NESTML

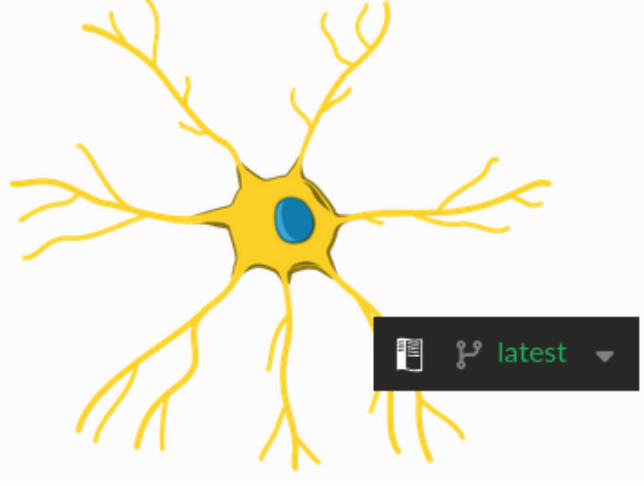
/ The NESTML modeling language

The NESTML modeling language

NESTML is a domain-specific language for neuron and synapse models. These dynamical models can be used in simulations of brain activity on several platforms, in particular the NEST Simulator.

NESTML combines:

- an easy to understand, yet powerful syntax;
- a flexible processing toolchain,



https://nestml.readthedocs.io

NESTML models library

nest::ml

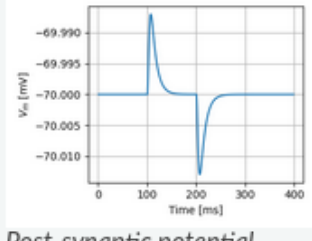
Search docs

- NESTML language documentation
- Installing NESTML
- Running NESTML
- Models library
  - Neuron models
  - Synapse models
- Tutorials
- Extending NESTML

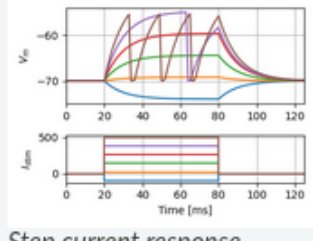
iaf_psc_alpha_neuron

iaf_psc_alpha - Leaky integrate-and-fire neuron model

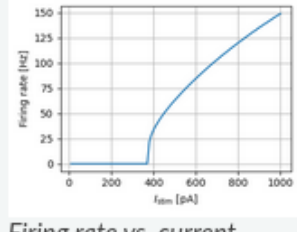
Source file: iaf_psc_alpha_neuron.nestml



Post-synaptic potential



Step current response



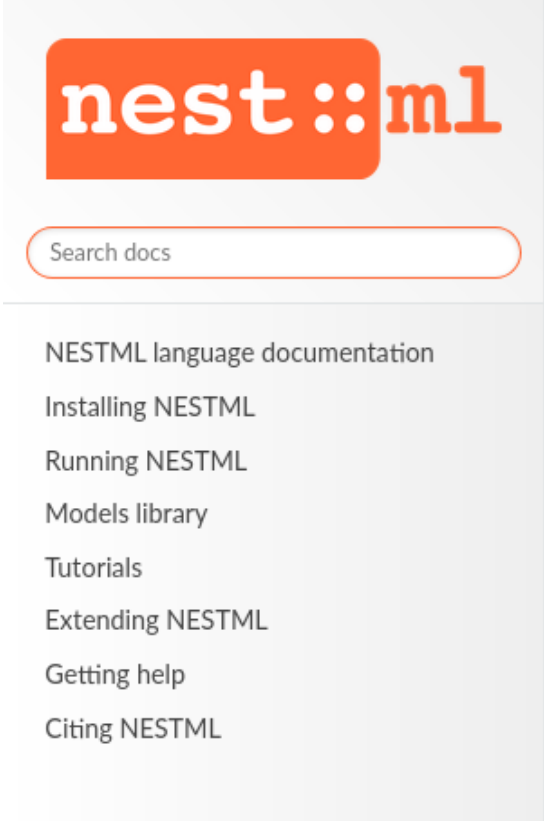
Firing rate vs. current

iaf_psc_delta_neuron

latest

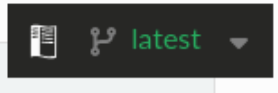
iaf_psc_delta - Current-based leaky integrate-and-fire neuron model with delta-kernel post-synaptic currents

Model description and code generation



[2]:
```
nestml_ou_model = """
model ornstein_uhlenbeck_noise_neuron:

    parameters:
        mean_noise real = 500    # mean of the noise
        sigma_noise real = 50    # std. dev. of the noise
        tau_noise ms = 20 ms     # time constant of the noise

    internals:
        A_noise real = sigma_noise * ((1 - exp(-2 * resolution() / tau_noise))

    state:
        U real = mean_noise   # set the initial condition

    update:
        U = mean_noise \
          + (U - mean_noise) * exp(-resolution() / tau_noise) \
          + A_noise * random_normal(0, 1)
"""
```

Save to a temporary file and make the model available to instantiate in NEST:



[3]:
```
# generate and build code
module_name, neuron_model_name_adapt_curr = \
    NESTCodeGeneratorUtils.generate_code_for(nestml_ou_model)
```

Hands on 👋

NESTML hands-on

- ➡️ Creating a new neuron model: https://lab.ebrains.eu
  - clone https://github.com/nest/nestml

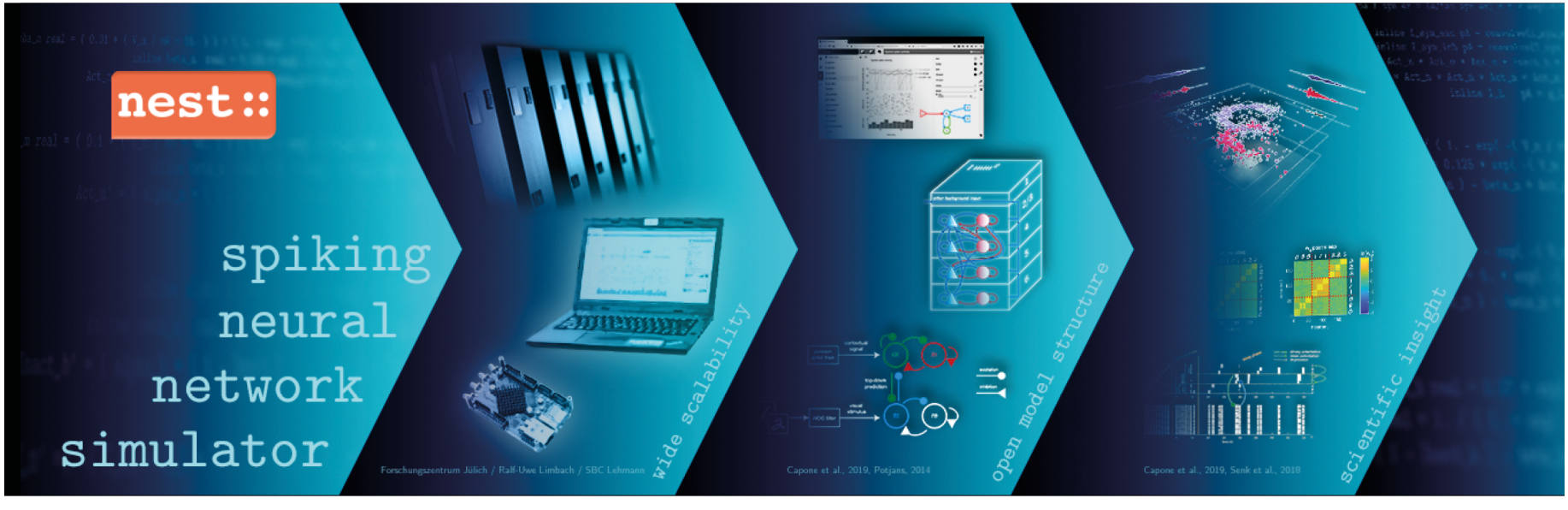
⇒Try it on  EBRAINS

Wrap-up

What we have seen

- ☑ Overview of NEST and its links to other tools and projects
- ☑ NEST Desktop and EBRAINS JupyterLab
- ☑ Eprop in NEST
- ☑ NESTML and first steps in creating neuron and synapse models

Where to go from here…



Where to go from here…

NEST Conference 2025

https://nest-simulator.org/conference

NEST-related publications

https://nest-simulator.org/publications

PyNN

https://neuralensemble.org/docs/PyNN/

More tools

https://www.ebrains.eu/tools

Feed-back

https://www.surveymonkey.com/r/NICE-NEST