Welcome to
NEST Conference
29-30 June 2020
First fully virtual conference
## NEST Conference 2020 — Program

### Monday, 29 June (all times CEST) – Morning

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:45</td>
<td>Registration</td>
</tr>
<tr>
<td>09:05</td>
<td>Welcome &amp; Introduction</td>
</tr>
<tr>
<td></td>
<td>Susanne Kunkel</td>
</tr>
</tbody>
</table>

**Session 1**

<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
<th>Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:25</td>
<td>NEST user-level documentation: Why, how and what’s next?</td>
<td>Sara Konradi</td>
</tr>
<tr>
<td>09:45</td>
<td>New Fenix resources at the Jülich Supercomputing Centre</td>
<td>Alexander Patronis</td>
</tr>
<tr>
<td>09:55</td>
<td>Minibreak</td>
<td></td>
</tr>
<tr>
<td>10:00</td>
<td>Toward a possible integration of NeuronGPU in NEST</td>
<td>Bruno Golosio</td>
</tr>
<tr>
<td>10:45</td>
<td>Coffee breakout</td>
<td></td>
</tr>
</tbody>
</table>

**Workshop 1**

<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
<th>Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:15</td>
<td>NEST multiscale co-simulation</td>
<td>Wouter Klijn</td>
</tr>
<tr>
<td>12:15</td>
<td>Lunch break</td>
<td></td>
</tr>
</tbody>
</table>

### - Afternoon

**Session 2**

<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
<th>Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>13:15</td>
<td>Thalamo-cortical spiking model of incremental learning combining</td>
<td>Chiara De Luca</td>
</tr>
<tr>
<td></td>
<td>perception, context and NREM-sleep-mediated noise-resilience</td>
<td></td>
</tr>
<tr>
<td>14:00</td>
<td>Minibreak</td>
<td></td>
</tr>
<tr>
<td>14:05</td>
<td>An Integrated Model of Visual Perception and Reinforcement</td>
<td>Petia Koprinkova-Hristova</td>
</tr>
<tr>
<td>14:25</td>
<td>Partial Information Decomposition Contextual Neurons in NEST</td>
<td>Sepehr Mahmoudian</td>
</tr>
<tr>
<td>14:45</td>
<td>Coffee breakout</td>
<td></td>
</tr>
</tbody>
</table>

**Session 3**

<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
<th>Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:15</td>
<td>deNEST: a declarative frontend for specifying networks and running</td>
<td>Tom Bugnon</td>
</tr>
<tr>
<td></td>
<td>simulations in NEST</td>
<td></td>
</tr>
<tr>
<td>15:35</td>
<td>Primate perisylvian cortex in NEST: from reflex to small world</td>
<td>Harry Howard</td>
</tr>
<tr>
<td>15:55</td>
<td>Minibreak</td>
<td></td>
</tr>
<tr>
<td>16:00</td>
<td>Modeling robust and efficient coding in the mouse primary visual</td>
<td>Stefan Mihalas</td>
</tr>
<tr>
<td></td>
<td>cortex using computational perturbations</td>
<td></td>
</tr>
<tr>
<td>16:45</td>
<td>Wrap-up</td>
<td>Hans Ekkehard Plesser</td>
</tr>
<tr>
<td>17:00</td>
<td>Mingle</td>
<td></td>
</tr>
</tbody>
</table>
Tuesday, 30 June (all times CEST) – Morning

<table>
<thead>
<tr>
<th>Session 4</th>
<th>Chair: Dennis Terhorst</th>
</tr>
</thead>
</table>
| 09:00     | A spiking neural network builder for systematic data-to-model workflow  
Carlos E Gutierrez |
| 09:20     | Insite: A Generalized Pipeline for In-transit Visualization and Analysis  
Simon Oehrl |
| 09:40     | Minibreak |
| 09:45     | NEST Component for Modeling Spiking Neuron-Astrocyte Networks  
Jugoslava Aćimović |
| 10:30     | Coffee breakout |

<table>
<thead>
<tr>
<th>Session 5</th>
<th>Chair: Tanguy Fardet</th>
</tr>
</thead>
</table>
| 11:00     | Atlas-mapped reconstruction and simulation of the cerebellar Lingula in NEST  
Alice Geminiani |
| 11:20     | Cerebro-cerebellar loops for sensorimotor adaptation  
Alberto Antonietti |
| 11:40     | Minibreak |
| 11:45     | Modeling plasticity inducing stimulation protocols in recurrent neuronal networks with homeostatic structural plasticity  
Swathi Anil |
| 12:30     | Lunch break |

- Afternoon

<table>
<thead>
<tr>
<th>Session 6</th>
<th>Chair: Jochen Eppler</th>
</tr>
</thead>
</table>
| 13:30     | Dendrites in NEST  
Willem Wybo |
| 13:50     | Just-in-time compilation for NESTML models in NEST Simulator  
Charl Linssen |
| 14:10     | NEST Desktop: A web-based GUI for the NEST Simulator  
Sebastian Spreizer |
| 14:30     | Minibreak |
| 14:40     | Evolving interpretable plasticity rules using NEST  
Jakob Jordan |
| 15:25     | Coffee breakout |

<table>
<thead>
<tr>
<th>Workshop 2</th>
<th>Chair: Stine Vennemo</th>
</tr>
</thead>
</table>
| 15:45      | deNEST: a declarative frontend for specifying networks and running simulations in NEST  
Tom Bugnon & William Mayner |
| 16:45      | Closing Remarks  
Abigail Morrison |
Sorted chronologically (by session/day).
NEST user-level documentation: Why, how and what’s next?

Sara Konradi¹, Steffen Graber¹, Dennis Terhorst¹

¹ Institute of Neuroscience and Medicine (INM-6) and Institute for Advanced Simulation (IAS-6), Jülich Research Centre, Jülich, Germany

Email: s.konradi@fz-juelich.de

Since the previous NEST Conference in 2019, we have made progress in improving the user-level documentation of NEST [1] with respect to completeness and accessibility towards the goal of operating NEST as part of the European research infrastructure EBRAINS from 2023. The concept of the user-level documentation has been worked out in a two-year process with the community. The talk summarizes the motivation behind high-quality documentation [2] and what we have achieved so far. We present how we use documentation-specific technology such as Sphinx [3] and Read the Docs [4] to produce user-level documentation that is easy to write, maintain, find, use and understand. An outcome of our analysis is that, for a community code, the user-level documentation also needs to be a community effort to ensure scalability and sustainability. Therefore, the talk discusses concepts of user-correctable documentation and its workflow. Finally, we outline the goals of the project for the next three years.

Figure 1. Upgrade from previously static HTML to new documentation on Read the Docs.

Acknowledgements

Funding was received from the European Union’s Horizon 2020 Framework Programme for Research and Innovation under grant agreements 720270 (HBP SGA1), 785907 (HBP SGA2); the Helmholtz Association Initiative and Networking Fund SO-902 (ACA).

References

1. NEST Simulator (www.nest-simulator.org)
3. Sphinx (www.sphinx-doc.org)
4. Read the Docs (www.readthedocs.org)

Copyright 2020 S. Konradi, S. Graber & D. Terhorst under Creative Commons Attribution License (CC BY-NC 4.0).
New Fenix resources at the Jülich Supercomputing Centre

Alexander Patronis¹, Dirk Pleiter¹

¹ Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich GmbH
Email: a.patronis@fz-juelich.de, d.pleiter@fz-juelich.de

Fenix is a distributed e-infrastructure providing HPC, cloud and storage resources, with the Human Brain Project being the prime and lead consumer. Access for other science and engineering communities is provided through PRACE. Initially, only the Swiss supercomputing centre CSCS was providing resources, with German, French and Spanish supercomputing centres (JSC, CEA and BSC, respectively) contributing more recently; the Italian centre CINECA is set to provide additional resources. In this contribution, we will provide a brief overview of the Fenix infrastructure and the resources which have recently become available at the Jülich Supercomputing Centre (JSC).

The latest services to be offered at the JSC, in support of the Fenix infrastructure, are provided through JUSUF (Jülich Support for Fenix), a cluster comprising 205 nodes, each equipped with two of the latest-generation AMD EPYC (Rome) 64-core processors (offering 944.6 TFLOPs peak performance), 256 GB of main memory, and a local 1 TB NVMe device. The compute nodes are interconnected with a 100 Gb/s HDR InfiniBand fat-tree network. The majority of nodes are available for scalable or interactive computing within a HPC environment, and are suitable for the simulation of large NEST models, while a subset of nodes are reserved to serve virtual machines in a cloud environment.
Toward a possible integration of NeuronGPU in NEST

Bruno Golosio\textsuperscript{1,2}, Chiara De Luca\textsuperscript{3}, Elena Pastorelli\textsuperscript{3,4}, Francesco Simula\textsuperscript{4}, Gianmarco Tiddia\textsuperscript{1}, Pier Stanislao Paolucci\textsuperscript{4}

\textsuperscript{1} Dipartimento di Fisica, Università di Cagliari, Italy
\textsuperscript{2} Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Cagliari, Italy
\textsuperscript{3} Ph.D. Program in Behavioural Neuroscience, “Sapienza” University of Rome
\textsuperscript{4} Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Roma, Italy

Email: golosio@unica.it

NeuronGPU is a library for MPI-GPU simulation of large scale networks of spiking neurons, written in the CUDA/C++ programming languages. Currently, this library includes different adaptive-exponential-integrate-and-fire (AdEx) multisynapse models, with current or conductance based synapses, together with user definable models, and several devices, as Poisson signal generators, spike generators, multimeters, spike recorders. Several connection rules are available. In addition to the standard synapses, the library includes STDP synapses. The numerical solution of the differential equations that describe the neuronal dynamics is based on a GPU implementation of the 5th order Runge-Kutta method with adaptive step-size control. On a single Nvidia GeForce RTX 2080 Ti GPU board it is possible to simulate the activity of 1 million multisynapse conductance-based AdEx neurons with 1000 synapse per neurons, for a total of 1 billion synapse, in about 71 seconds for each second of neural activity, with about 97 seconds of build time.

Recently, having in mind the possibility of integrating this library into the well-known NEST simulator [1], NeuronGPU has been equipped with a Python interface, analogous to the one of the NEST simulator. The next step of this integration, which is currently in progress, is the implementation of an interface, either based on the MUSIC interface [2], or on more specialized functions, to manage a quick exchange of information between NEST and NeuronGPU in both directions. With this interface, the NEST capabilities will be further extended to include the possibility of simulation on GPU supports. Furthermore, this interface would allow the unique possibility to simulate hybrid networks, with a very large number of neurons and connections based on simpler models simulated on the GPU side, and a smaller number of neurons and connections based on more complex and detailed models simulated on the CPU side and used both for consistency checks and to shed light on the relationship between the dynamics of the network and that of the neuron internal variables.

Acknowledgements

This work has been supported by the European Union Horizon 2020 Research and Innovation program under the FET Flagship Human Brain Project (grant agreement SGA3 n. 945539 and grant agreement SGA2 n. 785907) and by the INFN APE Parallel/Distributed Computing laboratory. The implementation of the AdEx model of this library closely follows the aeif\textsubscript{cond}\textsubscript{beta}\textsubscript{multisynapse} model present in the NEST simulator and implemented by one of the authors, B. Golosio, in collaboration with Prof. Hans Ekkehard Plesser and Dr. Tanguy Fardet.

References


Copyright 2020 B. Golosio, C. De Luca, E. Pastorelli, F. Simula, G. Tiddia & P.S. Paolucci under Creative Commons Attribution License (CC BY-NC 4.0).
Breakout session: NEST multiscale co-simulation

Wouter Klijn

1 SimLab Neuroscience, Jülich Supercomputing Centre (JSC), Institute for Advanced Simulation, Forschungszentrum Jülich GmbH, JARA, 52425 Jülich, Germany
Email: w.klijn@fz-juelich.de

The brain is an inherently multiscale system and the interplay between these scales remains an open question: Efficient simulation at each scale is critical, but not sufficient for understanding the whole system. SGA2 saw the completion of four interrelated proof of concepts: NEST-TVB coupling, NEST-Arbor coupling, in transit analysis using Elephant and in-situ visualization of NEST and Arbor, each provides fundamental insight into the requirements of co-simulation.

During SGA3, EBRAINS will deliver a modular framework for connecting models and simulation engines working at different or the same scales. It will include in situ analysis and communication infrastructure and science workflows that will provide the neuroscience community with novel tools for researching the multiscale nature of the brain. Co-simulation in EBRAINS is designed for large-scale deployment on HPC, including the necessary transducing modules between simulators and integration with EBRAINS tools for parameter space exploration, analysis and visualization.

NEST is a core component in the multiscale co-simulation efforts. The proposed breakout session will focus in the formalization of the interfaces of NEST with the co-simulation framework. Especially the Fat-endpoint approach taken for HPC deployment where a set of fundamental processing steps will be monitored in (near) real-time fashion, providing insight on simulation running on HPC resources (see figure 1). The target audience for this breakout session is NEST developers & co-simulation developers.

Figure 1: The Fat-endpoint approach sees applications on HPC systems deployed with an application companion. This application helps collect runtime insight into fundamental processing steps e.g.: steering, application loop, and inter application I/O.

Acknowledgements

Funded by the Helmholtz Association through the Helmholtz Portfolio Theme “Supercomputing and modeling for the Human Brain”. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 720270 (HBP SGA1) and No. 785907 (HBP SGA2).

Copyright 2020 W. Klijn under Creative Commons Attribution License (CC BY-NC 4.0).
Thalamo-cortical spiking model of incremental learning combining perception, context and NREM-sleep-mediated noise-resilience

Chiara De Luca\textsuperscript{3,4}, Bruno Golosio\textsuperscript{1,2}, Cristiano Capone\textsuperscript{4}, Elena Pastorelli\textsuperscript{3,4}, Giovanni Stegel\textsuperscript{5}, Gianmarco Tiddia\textsuperscript{1}, Giulia De Bonis\textsuperscript{4}, Pier Stanislao Paolucci\textsuperscript{4}

\textsuperscript{1} Dipartimento di Fisica, Università di Cagliari, Italy
\textsuperscript{2} Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Cagliari, Italy
\textsuperscript{3} Ph.D. Program in Behavioural Neuroscience, “Sapienza” University of Rome
\textsuperscript{4} Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Roma, Italy
\textsuperscript{5} Dipartimento di Chimica e Farmacia, Università di Sassari, Italy

Email: Chiara.DeLuca@roma1.infn.it

The brain exhibits capabilities of fast incremental learning from few noisy examples, as well as the ability to associate similar memories in autonomously-created categories, and to combine contextual hints with sensory perceptions. Together with sleep, these mechanisms are thought to be key components of many high-level cognitive functions. In particular, increasing experimental evidence is mounting for the role played by the combination of bottom-up (perceptual) and top-down/lateral (contextual) signals, and for the beneficial effects of sleep that appear to have impact on several aspects, such as pattern recognition, classification and decision making.

In this work [1], we exploited the combination of context and perception in a new thalamo-cortical model (ThaCo) based on a soft winner-take-all circuit of excitatory and inhibitory spiking neurons, starting from the description proposed by [2]. First, the new model is capable of undergoing multiple wake-sleep cycles during incremental learning, it adapts its pre-sleep, deep-sleep and post-sleep firing rates in a manner that is similar to the experimental measures of [3], and it demonstrates the cognitive role played by such adaptions. Second, it exploits the combination of context and perception during incremental learning, following the experimental cortical architectural principles and the evidence about the role of the combination of apical and basal inputs proposed by [4]; the two principles cooperate to support the incremental creation of a soft winner-take-all mechanism. Third, the model is capable of fast incremental learning from few examples and it is resilient when proposed with noisy perceptions and noisy contextual signals. Last, hundreds of memories during incremental learning are learnt by this model, also demonstrating comparable performances to artificial learning algorithms. Finally, the predictions of our model constitute a relevant contribution towards the reconciliation of recent experimental observations about both an average synaptic down-scaling effect and a differential modulation of firing rates.

Acknowledgements

This work has been supported by the European Union Horizon 2020 Research and Innovation program under the FET Flagship Human Brain Project (grant agreement SGA3 n. 945539 and grant agreement SGA2 n. 785907) and by the INFN APE Parallel/Distributed Computing laboratory.

References


Copyright 2020 C. De Luca, B. Golosio, C. Capone, E. Pastorelli, G. Stegel, G. Tiddia, G. De Bonis, P.S. Paolucci under Creative Commons Attribution License (CC BY-NC 4.0).
An Integrated Model of Visual Perception and Reinforcement Learning in NEST

Petia Koprinkova-Hristova¹, Nadejda Bocheva²

¹ Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria
² Institute of Neurobiology, Bulgarian Academy of Sciences, Sofia, Bulgaria

Email: pkoprinkova@yahoo.com

The paper summarizes our efforts to develop an integrated hierarchical spike timing neural network model of dynamic visual information processing and decision making with reinforcement in NEST. The inter- and intra-layer connectivity was based on the available information about involved brain structures starting from retinal ganglion cells through thalamic relay, visual cortex up to perceptual decisions area as well as basal ganglia that modulate the decision related to environmental feedback via reinforcement. The reported in [1] model structure was enriched by enhanced feedback connectivity as well as STDP synapses allowing for its behavior adaptation based on accumulated experimental data [2]. We investigated STDP synapses weights obtained after feeding the model output with an external signal corresponding to typical reaction times of the tested human subjects from three age groups (young, middle aged and elderly) during our experiments. Simulations with specific visual stimuli and external reinforcement signal demonstrated also that our model is able to change its decision accordingly.

Acknowledgements

The reported work was supported by the Bulgarian Science Fund under the project No DN02/3/2016 “Modelling of voluntary saccadic eye movements during decision making”.

References


Copyright 2020 P. Koprinkova-Hristova & N. Bocheva under Creative Commons Attribution License (CC BY-NC 4.0).
Partial Information Decomposition Contextual Neurons in NEST

Sepehr Mahmoudian1,2

1 Department of Data-driven Analysis of Biological Networks, Campus Institute for Dynamics of Biological Networks, Georg August University Göttingen, Göttingen, Germany
2 MEG Unit, Brain Imaging Center, Goethe University Frankfurt, Frankfurt, Germany

Email: sepehr.mahmoudian@gmail.com

The mammalian neocortex is the seat of our higher cognitive functions such as language. A core feature of the neocortex is the ability to create an internal representation of the world that is used to make predictions and interpret perception. To accomplish this, the neocortical machinery utilizes a hierarchical structure that is tightly coupled by pyramidal neurons.

At the cellular level, a single pyramidal neuron has tremendous processing power. The apical dendrites receive connections from other areas that provide context to the information processed in the area where the neuron is located [1]. The same pyramidal neurons form the building blocks of different modules throughout the neocortex, which suggests that perhaps some shared algorithms are used across the neocortex. Information theory is an ideal framework for studying such generic computations performed by these neurons in terms of the information processed and passed on.

Pyramidal neurons should integrate context with their driving information, i.e. two inputs. Yet, Shannon information theory was developed with only one input and output in mind. Mathematical advances [2] in the nascent field of partial information decomposition (PID) enable decomposing the output into those components unique to each input, that shared between the two inputs, and that which depends on both inputs but it is in neither, i.e. synergy. It is possible to derive learning rules [3][4] so as to train neurons to perform the required goal function based on these four terms.

I present the ongoing work on using the first generation of PID neurons for studying the hierarchical structure of the neocortex and their addition to NEST in form of new binary neurons.

Acknowledgements

This research was supported by the funding initiative Niedersächsisches Vorab of the Volkswagen Foundation and the Ministry of Science and Culture of Lower Saxony.

References


Copyright 2020 S. Mahmoudian under Creative Commons Attribution License (CC BY-NC 4.0).
We introduce *deNEST* ([https://github.com/tombugnon/denest](https://github.com/tombugnon/denest)), a Python package for specifying networks and running simulations in NEST using declarative parameter files. In contrast to procedural scripts, the declarative interface allows the user to fully specify the details of a simulation in a format that is concise, readable, and easily version-controlled, which facilitates sharing and reproducibility. Moreover, hiding the complexity of network setup behind a declarative interface makes simulations easier to reason about and facilitates running the same network in multiple conditions. Finally, the package provides a convenient API for interactive use with NEST.

The package was designed with large-scale multi-area networks in mind [1] but can be extended to other use-cases. We demonstrate the usage of our package with some worked examples and discuss limitations and future directions.

**References**

Primate perisylvian cortex in NEST: from reflex to small world

Harry Howard

1 Tulane University, New Orleans, United States of America

Email: howard@tulane.edu

[1] develops a six-area model of perisylvian cortex and instantiates it with two types of connectivity patterns, one representing monkey cortex and the other representing human cortex. The authors then propose that the processing advantages of the more elaborate human connectivity pattern is what underlies the evolution of language.

The neurocomputational model of [1] is based on a line of research that extends back to [2] and includes at least twelve other publications. The computer code for this work has never been published or distributed, to the best of our knowledge, so our first task is to replicate it in NEST for diffusion to a larger audience.

With a NEST instantiation in hand, we can begin to examine this object more critically and ask more theoretical questions. The first observation that we make is that the monkey network answers to the definition of a pyramidal neural network as first adumbrated in the connectionist framework in [3], which has recently been reborn in the deep-learning framework through the work of [4].

Activation flowing from base to peak of a pyramidal network has a distinctly linear structure, which reminds us of reflex action, see for instance [5], and leads us to the conjecture that monkeys are limited to calls (and not song) due to the pyramidal, quasi-reflex connectivity of their perisylvian cortex.

As for the more complex connectivity of human perisylvian cortex, we conjecture that it has the effect of creating a small-world topology, see for instance [6].

References


Copyright 2020 H. Howard under Creative Commons Attribution License (CC BY-NC 4.0).
Modeling robust and efficient coding in the mouse primary visual cortex using computational perturbations

Binghuang Cai¹, Yazan N. Billeh¹, Selmaan N. Chettih², Christopher D. Harvey², Christof Koch¹,
Anton Arkhipov¹, Stefan Mihalas¹

¹ Allen Institute for Brain Science, Seattle, WA, USA
² Department of Neurobiology, Harvard Medical School, Boston, MA, USA

Email: stefanm@alleninstitute.org

Investigating how visual inputs are encoded in visual cortex is important for elucidating the roles of cell populations in circuit computations. We here use a recently developed, large-scale model of mouse primary visual cortex (V1) [1] and perturb both single neurons as well as functional- and cell-type defined population of neurons to mimic equivalent optogenetic perturbations.

First, perturbations were performed to study the functional roles of layer 2/3 excitatory neurons in inter-laminar interactions. We observed activity changes consistent with the canonical cortical model [2]. Second, single neuron perturbations in layer 2/3 revealed a center-surround inhibition-dominated effect, consistent with recent experiments [3]. Finally, perturbations of multiple excitatory layer 2/3 neurons during visual stimuli of varying contrasts indicated that the V1 model has both efficient and robust coding features.

The circuit transitions from predominantly broad like-to-like inhibition at high contrasts to predominantly specific like-to-like excitation at low contrasts. These in silico results demonstrate how the circuit can shift from redundancy reduction to robust codes as a function of stimulus contrast.

Acknowledgements
We wish to thank the Allen Institute for Brain Science founder, Paul G. Allen, for his vision encouragement and support.

References
A spiking neural network builder for systematic data-to-model workflow

Carlos E Gutierrez¹, Henrik Skibbe², Kenji Doya¹

¹ Neural Computation Unit, Okinawa Institute of Science and Technology Graduate University, Okinawa, Japan
² Brain Image Analysis Unit, Riken Center for Brain Science, Wako, Japan

Email: carlos.gutierrez@oist.jp

Brain modeling requires an efficient incorporation of experimental data into mathematical features, conversion to executables codes, comparison with real system performance, and exploration of unknown parameters. There is no unique approach to build models of the brain, and frequently models are seen as black boxes, with questioned transparency, reproducibility and validation. Tools for systematic incorporation of anatomical and physiological data can improve the impact of computational models in neuroscience.

We present a framework to support data-driven development of spiking neural network (SNN) models based on a common Entity-Relationship (ER) data description. The framework is designed to capture knowns “as much as possible” and to optimize variables, as well as objective functions.

We arrange all the data attributes including species, brain regions, neurons, projections, neuron models, and references as tables and relations within a database management system (DBMS) and GUI interfaces for navigation. This supports a systematic and collaborative SNN building and references traceability for every detail attached to it.

We tested this data-to-model framework for our basal ganglia model, linking data from 51 papers and existing neuron and synapse models for NEST 3 code generation. We also implemented output of SONATA model representation [1] and validated it by code generation and simulation by NEST 2.18 [2].

The organized data-driven modeling aims to shorten the gap between multiple data sources and its effective use in models of the brain, while assisting researchers over the workflow.

Our framework supports data integrity and consistency, desirables characteristics for assembling SNN’s into larger structures, and data comparisons across species. Furthermore, it is aimed to link Brain/MINDS [3] data sources for systematic SNN modeling of the marmoset brain.

Acknowledgements

This research is supported by the Collaboration Research for Development of Techniques in Brain Science Database Field (FY2019), and the Collaborative Technical Development in Data-driven Brain Science (FY 2020) grants, and internal OIST funding.

References


Copyright 2020 C.E. Gutierrez, H. Skibbe & K. Doya under Creative Commons Attribution License (CC BY-NC 4.0).
Neuronal network simulators are essential to computational neuroscience, enabling the study of the nervous system through in-silico experiments. Through utilization of high-performance computing resources, these simulators are able to simulate increasingly complex and large networks of neurons today. It also creates new challenges for the analysis and visualization of such simulations. In-situ and in-transport strategies are popular approaches in these scenarios [1, 2]. They enable live monitoring of running simulations and parameter adjustment in the case of erroneous configurations which can save valuable compute resources.

This talk will present the current status of our pipeline for in-transport analysis and visualization of neuronal network simulator data [3]. The pipeline is able to couple with NEST [4] along other simulators with data management (querying, filtering and merging) from multiple simulator instances. Finally, the data is passed to end-user applications for visualization and analysis. The goal is to be integrated into third party tools such as the multi-view visual analysis toolkit ViSimpl [5].

Acknowledgements
This project/research has received funding from the European Union’s Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2).

References
NEST Component for Modeling Spiking Neuron-Astrocyte Networks

Jugoslava Aćimović¹, Mikko Lehtimäki¹, Tiina Manninen¹, Jonas Stapmanns²,³, Han-Jia Jiang²,³, Sacha van Albada²,⁴, Markus Diesmann²,⁵,⁶, Marja-Leena Linne¹

¹ Faculty of Medicine and Health Technology, Computational Neuroscience, Tampere University, Tampere, Finland
² Jülich Research Centre, Institute of Neuroscience and Medicine (INM-6) and Institute for Advanced Simulation (IAS-6), Jülich, Germany
³ RWTH Aachen University, Aachen, Germany
⁴ Institute of Zoology, Faculty of Mathematics and Natural Sciences, University of Cologne, Cologne, Germany
⁵ Department of Psychiatry, Psychotherapy and Psychosomatics, School of Medicine, RWTH Aachen University, Aachen, Germany
⁶ Department of Physics, Faculty 1, RWTH Aachen University, Aachen, Germany

Email: jugoslava.acimovic@tuni.fi, mikko.lehtimaki@tuni.fi

Recent findings suggest a potential conceptual revolution about the roles of astrocytes in modulation of neuronal circuits. Astrocytic influence is necessary for synaptic maturation in developing neurons, and in the adult brain astrocytes are in close proximity to synapses to regulate neurotransmission. Moreover, astrocytes have been shown to take part in transitions between brain states and in brain rhythms during sleep. Astrocytic modulations of mammalian synapses, circuits, and behavior are, however, complex, and we have therefore evaluated computational neuron-astrocyte interaction models [1] to analyze in detail the role of astrocytes in synaptic and circuit functions. Our study confirmed that astrocytes possess a multitude of mechanisms to modulate neural systems, including the so-called slow inward current (SIC). SIC is an excitatory current in neurons caused by astrocytic glutamate release and activation of neuronal extrasynaptic NMDA receptors. In this study, we implemented a computational model of the SIC [2] as a new NEST component. The following steps were needed to harmonize the SIC model with the spiking network formalism: 1) we extended the standard synaptic model available in NEST to facilitate communication with an astrocyte model, 2) we added a component that implements intracellular production of ions and molecules in astrocytes, and 3) we added a new astrocyte-to-neuron interaction through SIC. The new NEST component was then incorporated into a model of mammalian cortical network [3, 4]. Our preliminary tests show that astrocytes modulate global network synchronization at the time scale of seconds.

In summary, we propose a new NEST component to support further studies of nonstandard synaptic astrocyte-mediated communication and its role in modulating global activity states in large spiking neuronal networks. This present model is a step towards developing more realistic circuits with complex connectivity and larger diversity of neuronal and astroglial cell types.

Acknowledgements

This research has received funding from the European Union’s Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2) and from Academy of Finland project no. 297893, 326494 and 326495.

References

4. Aćimović et al., in preparation

Copyright 2020 J. Aćimović, M. Lehtimäki, T. Manninen, J. Stapmanns, H-J. Jiang, S. van Albada, M. Diesmann & M-L. Linne under Creative Commons Attribution License (CC BY-NC 4.0).
Atlas-mapped reconstruction and simulation of the cerebellar Lingula in NEST

Alice Geminiani¹, Dimitri Rodarie², Robin De Schepper¹, Claudia Casellato¹, Egidio D’Angelo¹

¹ Department of Brain and Behavioral Sciences, University of Pavia, Pavia, Italy
² Blue Brain Project, École Polytechnique Fédérale de Lausanne, Geneva, Switzerland

Email: alice.geminiani@unipv.it

To investigate temporal and spatial properties of brain functioning, spiking network models of brain circuits should embed realistic temporal dynamics of neurons, synapses and microcircuit, with region-dependent spatial features. Indeed, in the case of the cerebellum, region-specific structural properties and local network dynamics impact on mapping, integration and processing of input signals [1]. We here describe the reconstruction of the mouse Lingula (region of the cerebellar Vermis) and the simulation in NEST, analyzing network dynamics and activity propagation in the 3D space. The network is reconstructed based on density and orientation data from the Allen Brain Atlas as in [2], using strategies of the cerebellar scaffold [3], with new features for folded volumes. The main cerebellar cortical neurons are placed using a particle placement strategy, except for Purkinje Cells, positioned as parallel arrays with constraints on anisotropic distances among cells. Neurons are then connected using volume intersection algorithms applied on rotated morphologies and bended fibers according to the folded volume. The resulting circuit includes about $10^5$ neurons and $10^7$ connections. The network is then simulated in NEST [4], with neurons modelled as Extended-Generalized Leaky Integrate and Fire [5] and alpha conductance-based synapses. A 4-Hz input is provided to the Granular layer, and a 100-Hz burst to two spots of adjacent glomeruli. The activity patterns throughout the circuit (e.g. oscillations, center-surround patterns) are analyzed in the 3D folded volume. The simulations provide insights into spatial features of signal propagation in the cerebellum and give an example of how large-scale network simulations in NEST can be used to investigate temporal and spatial network dynamics in specific region geometry. High-Performance Computing resources are used and will be fundamental in future simulations of a full mouse cerebellum. Eventually, the reconstructed atlas-mapped spiking cerebellar regions could be embedded in whole-brain frameworks built on Atlases, where interconnected brain areas can be simulated even using different descriptions (e.g. NEST-The Virtual Brain co-simulations).

Acknowledgements

This work has received funding from EU’s H2020 Framework Programme for Research and Innovation under Grant Agreement N. 785907 (HBP SGA2), SGA3 and Voucher Programme VM-CerebNEST. HPC resources are provided by ICEI/FENIX Infrastructure (project Simulations of cerebellar circuits: from detailed models to large-scale spiking networks with plasticity, id. ich027).

References


Copyright 2020 A. Geminiani, D. Rodarie, R. De Schepper, C. Casellato & E. D’Angelo under Creative Commons Attribution License (CC BY-NC 4.0).
Cerebro-cerebellar loops for sensorimotor adaptation

Alberto Antonietti¹, Edoardo Negri¹, Robin De Schepper¹, Alessandra Pedrocchi², Egidio D’Angelo¹,³, Claudia Casellato¹

¹ University of Pavia, Pavia, Italy
² Politecnico di Milano, Milano, Italy
³ IRCCS Mondino Foundation, Pavia, Italy

Email: alberto.antonietti@unipv.it

Sensorimotor signals are integrated and processed by the cerebellar circuit to predict accurate control of actions. In order to investigate how single neuron dynamics and geometrical modular connectivity affect cerebellar processing, we have exploited an olivocerebellar Spiking Neural Network, built with advanced structural features in terms of neuronal populations and connectome, simulated in NEST [1]. Distributed long-term plasticity has been introduced at multiple connection sites. Synaptic strengths are modulated through bidirectional ad hoc learning rules, generating long-term potentiation and depression [2].

The architecture is modular, including the cerebellum working both as a forward and inverse model. The cerebellar circuits have been inserted into loops and receive input signals coding both the system status information and sensory or motor error signals driving the plasticity. Encoding and decoding strategies have been designed to interface the cerebellar spiking networks with other control blocks representing specific neural regions. In particular, sensory feedback and intentional planning signals are converted into spike patterns and fed to the neural circuits. On the other hand, the spiking network outputs are sent as actuation commands to a peripheral system (eventually a virtual neurorobot). We emulated a pointing task perturbed by prismatic glasses [3]. In an arm-reaching experiment, humans wear prism lenses shifting their visual field by a certain lateral deviation (e.g., 25°). Subjects could correct their reaching movements based on touching error, i.e., the horizontal displacement between the touched and target positions, only after the target point had been reached. In order to mimic the control system blocks involved in this sensorimotor adaptation paradigm, we included abstract cerebral cortex modules and multiple instances of the cerebellar model into the control system. While the cerebral cortex was not provided with plasticity mechanisms, the two cerebellar modules included plastic connections. Neural plasticity in the cerebellar forward and inverse internal models allowed to achieve complementary processing able to predict a sensory discrepancy and to generate compensatory motor commands.

Acknowledgements

This research was supported by the European Union’s Horizon 2020 Framework Program for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2). The simulations were performed with ICEI/Fenix computational resources at CSCS (project ID: ich027).

References


Copyright 2020 A. Antonietti, E. Negri, R. de Schepper, A. Pedrocchi, E. D’Angelo & C. Casellato under Creative Commons Attribution License (CC BY-NC 4.0).
Modeling plasticity inducing stimulation protocols in recurrent neuronal networks with homeostatic structural plasticity

Swathi Anil¹, Han Lu²,³, Júlia V. Gallinaro², Stefan Rotter², Stefan Rotter²,°, Andreas Vlachos¹²,4,°

¹ Department of Neuroanatomy, Institute of Anatomy and Cell Biology, Faculty of Medicine, University of Freiburg, Freiburg, Germany
² Bernstein Center Freiburg & Faculty of Biology, University of Freiburg, Freiburg, Germany
³ Institute of Cellular and Integrative Neuroscience, University of Strasbourg, Strasbourg, France
⁴ Center for Basics in Neuromodulation (NeuroModulBasics), Faculty of Medicine, University of Freiburg, Freiburg, Germany
° joint senior authors

Email: swathi.anil@anat.uni-freiburg.de

A defining feature of neural tissue is the ability to respond to specific stimuli with functional and structural adaptations. Such changes at synaptic contact sites, i.e., synaptic plasticity, are considered fundamental for complex brain functions. In past decades activity-dependent plasticity mechanisms have been extensively studied, with Hebbian and homeostatic plasticity being cornerstone concepts. However, it remains unclear how these two mechanisms, which are based on positive or negative feedback, respectively, can co-exist in the same networks, neurons, and even synapses [1]. Interestingly the effects of classic Hebbian plasticity paradigm, e.g., local tetanic electrical stimulation, have not yet been systematically evaluated for their effects on homeostatic synaptic plasticity induction. In the present study, we adopted a computational approach to explore the effects of classic LTP/LTD-protocols in an inhibition-dominated random network of leaky integrate-and-fire point neurons, which undergoes structural remodeling based on firing rate homeostasis [2]. Each neuron in this network has a set point of intracellular calcium concentration, reflecting its firing rate. Any change induced by external input is counteracted by changes in structural connectivity to bring firing rates back to this set point. We examined the results of classic plasticity protocols, using trains of very short DC pulses (e.g., 100 pulses at 100 Hz; 900 pulses at 1 Hz), and recorded the resulting changes in firing rate and network connectivity. Results obtained in a small network (500 neurons) suggest that reliable network remodeling can be triggered, very similar to what was previously observed in a model of transcranial DC stimulation (tDCS) protocols [3]. In particular, we found differential effects of structural plasticity among the stimulated and non-stimulated neurons, and between these groups. We are currently expanding these simulations to larger network models and explore plasticity protocols used in clinical practice for (non-)invasive brain stimulation.

Acknowledgements
Supported by Bundesministerium für Bildung und Forschung (BMBF, 01GQ1804A).

References

Copyright 2020 S. Anil, J.V. Gallinaro, S. Rotter & A. Vlachos under Creative Commons Attribution License (CC BY-NC 4.0).
Dendrites in NEST

Willem A.M. Wybo¹, Jakob Jordan¹, Mihai Petrovici¹, Walter Senn¹

¹ Department of Physiology, University of Bern, Bern, Switzerland
Email: willem.a.m.wybo@gmail.com

The dendritic trees of neurons endow them with a computational repertoire that is far larger than that of their point-like counterparts [1]. Yet, how this repertoire can be exploited in networks of neurons remains largely unexplored. While NEST provides the infrastructure to efficiently simulate large-scale networks [2], it does not natively support neurons with dendritic trees.

Here, we present a new modeling framework for NEST that enables users to define and simulate custom compartmental models. These models use the standard NEST interface to receive and send spikes, but implement coupled compartments that can be targeted by AMPA, NMDA or GABA synapses. The dendritic layout is defined by the user at runtime, and can thus be adapted to the neuron type or dendritic computation at hand.

A new method has recently shown that models with few compartments can be fitted to reproduce the dynamics of their detailed counterparts [3]. NEAT (the NEural Analysis Toolbox, https://neatdend.readthedocs.io/en/latest/) implements this method and provides a high-level Python interface to reduce dendritic models. NEAT thus extracts the parameters of the reduced compartmental models for efficient simulation in NEST. In the future, exporting these models to NEST will be automatized.

In recent years, there has been a growing interest in the computational role of dendritic trees. For instance, theories about credit assignment in the brain hypothesize that dendritic trees may integrate error signals required for learning [4]. Adding models with dendrites thus renders NEST an attractive option for researchers interested in these theories and may further expand its user base.

Acknowledgements
This work was supported by the European Union’s Horizon 2020 Framework Programme for Research and Innovation (Human Brain Project, Specific Grant Agreements No. 785907, SGA1-2,3) and by the SNF Sinergia grant CRSII5_180316 (with F. Helmchen). We thank our lab colleagues for helpful discussions and critical comments.

References

Copyright 2020 W.A.M. Wybo, J. Jordan, M. Petrovici & W. Senn under Creative Commons Attribution License (CC BY-NC 4.0).
Just-in-time compilation for NESTML models in NEST Simulator

Charl Linssen\textsuperscript{1,2}, Jochen M. Eppler\textsuperscript{1}, Abigail Morrison\textsuperscript{1,3}

\textsuperscript{1} Simulation Lab Neuroscience, Jülich Supercomputer Centre, Institute for Advanced Simulation, Jülich-Aachen Research Alliance, Forschungszentrum Jülich GmbH
\textsuperscript{2} Institute for Neuroscience and Medicine INM-6 / INM-10, Jülich-Aachen Research Alliance, Institute for Advanced Simulation IAS-6, Forschungszentrum Jülich GmbH
\textsuperscript{3} Institute of Cognitive Neuroscience, Ruhr-University Bochum, Germany

Email: c.linssen@fz-juelich.de

NESTML is a modeling language for neurons and synapses, designed to decrease design time and design complexity for the computational modeler, without compromising on simulation performance by means of fully automated source-code generation. Although support for synapse models was added early on, the current workflow engenders a limitation in memory and runtime performance. In general, starting with a NESTML model, the toolchain is invoked to generate target platform code for that model, which is then (cross-)compiled and built. A neuron model and a synapse model are thus independently built. However, in practice, storing and updating state variables pertaining to the postsynaptic neuron in each synapse instance is redundant, because this state only needs to be maintained once for each neuron. The specification of the state and dynamics is dependent on the type of synaptic plasticity rule that is used, so would typically be a part of the NESTML synapse model. A workflow is therefore desired that can identify dynamics that is redundant for certain, given, combinations of neuron and synapse models, and adjust the generated code in a way to eliminate the redundancy. We refer to this workflow as the “co-generation” of neuron and synapse model code for the target platform. Although models can still be built independently as before, a new “co-generation” mode processes a neuron and synapse as a dyad.

We further note that the collection of (neuron, synapse) pairs is typically not known at the time that NEST Simulator is conventionally built, but only emerges after the user has specified network connectivity in their simulation script (for example, through the PyNEST interface). For this reason, we propose to implement just-in-time (“JIT”) compilation into NEST Simulator. Hooking into the PyNEST call nest.Simulate() allows us to retrieve a list of (neuron, synapse) pairs from the NEST kernel and generate a NEST user module, that can be dynamically loaded into a running NEST Simulator instance. Control is then passed back to PyNEST for further execution of the simulation. The “co-generation” workflow is thus completely automated and does not alter the way that a user would typically interact with NEST Simulator.

Acknowledgments

This project has received funding from the Helmholtz Association through the Helmholtz Portfolio Theme “Supercomputing and Modeling for the Human Brain” and the European Union’s Horizon 2020 research and innovation programme under grant agreement No 720270 (HBP SGA1), and No 785907 (HBP SGA2).

References


Copyright 2020 C. Linssen, J.M. Eppler & A. Morrison under Creative Commons Attribution License (CC BY-NC 4.0).
NEST Desktop: A web-based GUI for the NEST Simulator

Sebastian Spreizer\textsuperscript{1,2,3}, Johanna Senk\textsuperscript{2}, Stefan Rotter\textsuperscript{1}, Markus Diesmann\textsuperscript{2,4,5}, Benjamin Weyers\textsuperscript{1}

\textsuperscript{1} Department IV - Computer Science, Human-Computer Interaction, University of Trier, Trier, Germany
\textsuperscript{2} Institute of Neuroscience and Medicine (INM-6) and Institute for Advanced Simulation (IAS-6) and JARA-Institute Brain Structure-Function Relationships (INM-10), Jülich Research Centre, Jülich, Germany
\textsuperscript{3} Bernstein Center Freiburg & Faculty of Biology, University of Freiburg, Freiburg, Germany
\textsuperscript{4} Department of Psychiatry, Psychotherapy and Psychosomatics, School of Medicine, RWTH Aachen University, Aachen, Germany
\textsuperscript{5} Department of Physics, Faculty 1, RWTH Aachen University, Aachen, Germany

Email: spreizer@uni-trier.de

We have developed a web-based graphical user interface (GUI) for the NEST simulator: NEST Desktop [1]. This GUI enables the rapid construction, parametrization, and instrumentation of neuronal network models typically used in computational neuroscience. The primary objective was to create a tool of classroom strength that allows users to rapidly explore neuroscience concepts without the need to learn a simulator control language at the same time.

Earlier versions of NEST Desktop required a full NEST installation on the user’s machine which limited not only the uptake by a non-expert audience but also the network models studied to what can be simulated on a laptop or desktop computer. To ease the use of the app and increase the range of simulations possible with NEST Desktop, we have separated the GUI from the simulation kernel: the web browser renders the GUI while the simulation kernel runs on a centrally maintained server.

NEST Desktop is designed as an extendable multi-purpose GUI for the NEST simulator. In this contribution we discuss the potential of using an in-situ pipeline developed for neuronal network simulators to enable the app to receive larger data sets from an ongoing NEST simulation. This enhances the interactivity of NEST for large simulations on HPC facilities. In addition, we present concepts and prototypes of an interface to the Python toolbox Elephant enabling statistical analysis of simulation results and the integration of NESTML for the specification of custom models.

In order to give students and researchers installation-free access to the compute resources being built up by the European Union, we integrated NEST Desktop into the EBRAINS infrastructure [2] also facilitating long-term sustainability. The same code remains available as a stand-alone version of NEST Desktop [3] for applications in teaching and training and installations at other sites.

Acknowledgements

This project has received funding from the European Union’s Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreement No. 785907 (Human Brain Project SGA2) and 945539 (Human Brain Project SGA3) and the Helmholtz Association Initiative and Networking Fund under project number SO-092 (Advanced Computing Architectures, ACA).

References

1. https://nest-desktop.readthedocs.io
2. https://ebrains.eu/services/simulation/nest/

Copyright 2020 S. Spreizer, J. Senk, S. Rotter, M. Diesmann & B. Weyers under Creative Commons Attribution License (CC BY-NC 4.0).
Evolving interpretable plasticity rules using NEST

Jakob Jordan*¹, Maximilian Schmidt²,³, Walter Senn¹, and Mihai A. Petrovici¹⁴

¹ Department of Physiology, University of Bern, Bern, Switzerland
² Ascent Robotics, Tokyo, Japan
³ RIKEN Center for Brain Science, Tokyo, Japan
⁴ Kirchhoff-Institute for Physics, Heidelberg University, Heidelberg, Germany

Email: jakob.jordan@pyl.unibe.ch

* These authors contributed equally to this work.

Continuous adaptation allows survival in an ever-changing world. Adjustments in the synaptic coupling strength between neurons are essential for this capability, setting us apart from simpler, hard-wired organisms. Can we find general, phenomenological models that capture these adjustments? Insights into this question are essential both for understanding biological information processing and for developing cognitively performant artificial systems. We suggest an automated approach for finding biophysically plausible plasticity rules in spiking neural networks based on the definition of task families, associated performance measures and biophysical constraints [1]. Using genetic programming, we evolve compact symbolic expressions for synaptic plasticity rules in spiking networks simulated with NEST [2]. To this end, we develop new synapse modules that parse arbitrary symbolic expressions via the C++ library SymEngine [3]. These modules are highly flexible and allow customization of synaptic plasticity rules at runtime via PyNEST. We illustrate our implementation by evolving efficient plasticity rules in error-driven and correlation-driven learning scenarios.

Acknowledgements

We gratefully acknowledge funding from the European Union, under grant agreements 604102, 720270, 785907 (HBP) and the Manfred Stärk Foundation. We further express our gratitude towards the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for co-funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC). We would like to thank all participants from the HBP SP9 Fürberg meetings for stimulating interactions and Tomoki Fukai for initial discussions and support. We also thank Henrik Mettler for helpful comments on the manuscript. All network simulations carried out with NEST (www.nest-simulator.org).

References

3. https://github.com/symengine/symengine

Copyright 2020 J. Jordan, M. Schmidt, W. Senn & M.A. Petrovici under Creative Commons Attribution License (CC BY-NC 4.0).
We introduce deNEST ([https://github.com/tombugnon/denest](https://github.com/tombugnon/denest)), a Python package for specifying networks and running simulations in NEST using declarative parameter files. In contrast to procedural scripts, the declarative interface allows the user to fully specify the details of a simulation in a format that is concise, readable, and easily version-controlled, which facilitates sharing and reproducibility. Moreover, hiding the complexity of network setup behind a declarative interface makes simulations easier to reason about and facilitates running the same network in multiple conditions. Finally, the package provides a convenient API for interactive use with NEST. The package was designed with large-scale multi-area networks in mind [1] but can be extended to other use-cases. During the session, we will provide an extensive tutorial on how to use the package, and assist the participants in writing parameter files for their own use cases.

References
NEST Initiative Membership

The NEST Initiative is a non-profit member-based organization incorporated in Ecublens, VD, Switzerland, on 4 October 2012. Any natural (i.e., you as an individual) or legal (e.g., your company or university) person sharing and supporting the goals of the NEST Initiative can become a member of the Initiative.

Read more about membership types and benefits at:

https://nest-initiative.org/membership