JARA Jülich Aachen Research Alliance

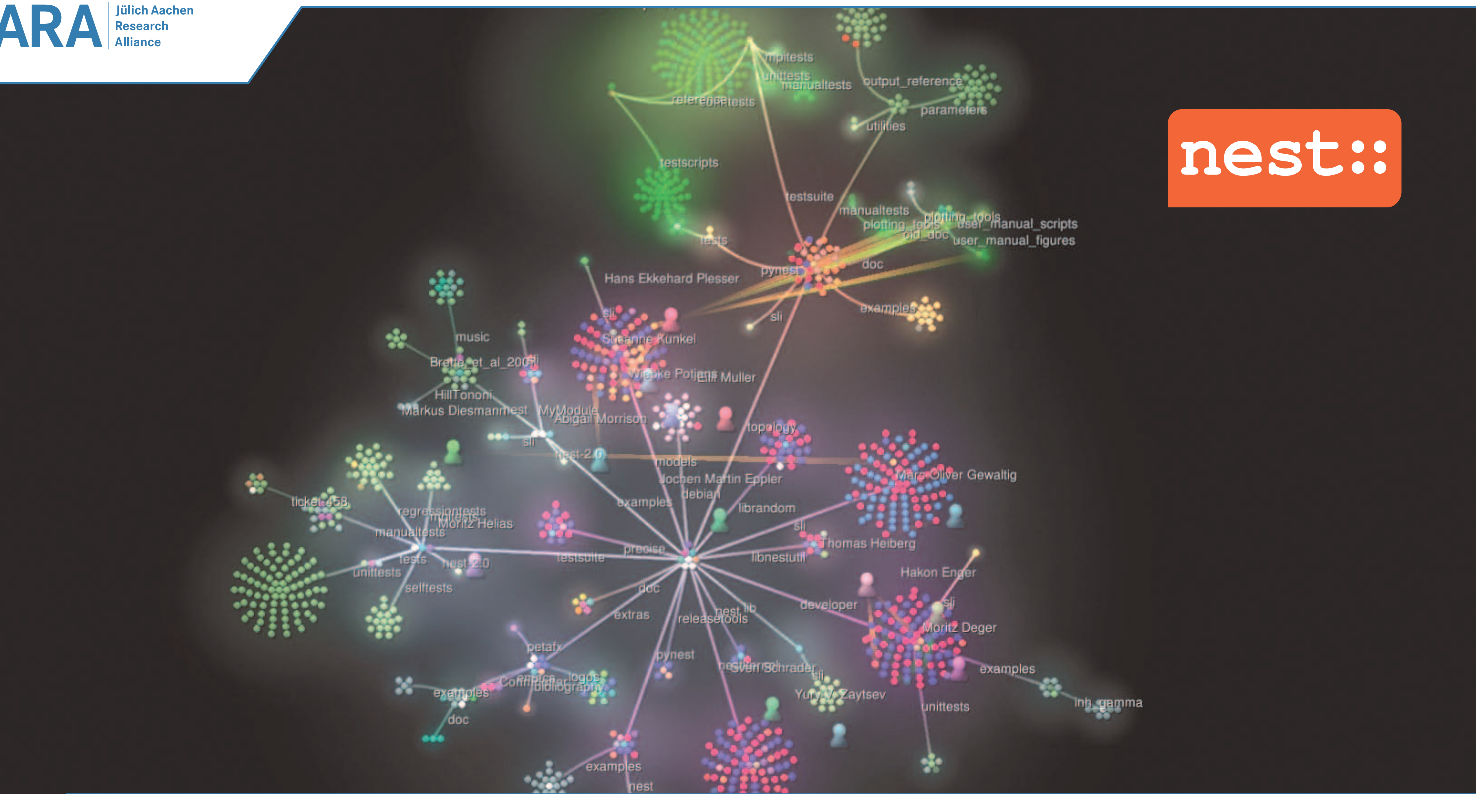



Image source: Yury V. Zaytsev

nest::

# NEST: The Neural Simulation Tool

AN INITIATIVE OF

RWTH AACHEN UNIVERSITY

JÜLICH FORSCHUNGSZENTRUM

PARTNERS

PROJECTS

COMMUNITY BUILDING

QUALITY AND PERFORMANCE

APPLICATIONS

USAGE EXAMPLE

## A SIMULATOR TO EXPLORE AND UNDERSTAND BIOLOGICAL NEURONAL NETWORKS.

Science has driven the development of the NEST simulator for the past 20 years. Originally created to simulate the propagation of synfire chains using single-processor workstations, we have pushed NEST's capabilities continuously to address new scientific questions and computer architectures. Prominent examples include studies on spike-timing dependent plasticity in large simulations of cortical networks, the verification of mean-field models, models of Alzheimer's and Parkinson's disease and tinnitus. Recent developments include a significant reduction in memory requirements, as demonstrated by a record-breaking simulation of 1.73 billion neurons connected by 10.4 trillion synapses on the Japanese K supercomputer, paving the way for brain-scale simulations.

Running on everything from laptops to the world's largest supercomputers, NEST is configured and controlled by high-level Python scripts, while harnessing the power of C++ under the hood. An extensive testsuite and systematic quality assurance ensure the reliability of NEST.

nest::

# PARTNERS

The development of NEST is driven by the demands of neuroscience and carried out in a collaborative fashion at different institutes all over the world.

New functionality is always added by researchers who need it, ensuring the most usable implementation of novel features.

To guarantee the best possible performance of the code and a maximum accuracy of the scientific results obtained, neuroscientists are supported by experts in computational sciences, software engineering, and math.

## NORWEGIAN UNIVERSITY OF LIFE SCIENCES (NMBU)

The Computational Neuroscience Group at the Department of Mathematical Sciences and Technology (IMT) at NMBU has made key contributions to NEST's hybrid parallelization and is lead developer for the Topology Module for spatially structured networks. Combining NEST with the simulator NEURON and the toolbox LFPy, it develops multi-scale models predicting LFP and MUA signals from network simulations.

## NATIONAL BERNSTEIN NETWORK COMPUTATIONAL NEUROSCIENCE

Researchers from the Bernstein Center Freiburg (BCF) and the "Bernstein Facility Simulation and Database Technology" (BFSD) at Forschungszentrum Jülich work on neuron models and infrastructure for developing and deploying NEST.

## ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE (EPFL)

The Neurorobotics Section of the EPFL Blue Brain Project uses and develops NEST for neurorobotics simulations with closed action-perception loops. In addition, NEST is used to develop and investigate data-driven whole-brain models of rodents at the level of point neurons. Future work will focus on extending the capabilities of NEST towards the real-time control of simulated and physical robots.

## RUHR-UNIVERSITÄT BOCHUM (RUB)

NEST originated as a student project at the Institute for Neural Computation over 20 years ago. Today, the group Functional Neural Circuits at RUB is investigating top-down models of network function and adapting NEST accordingly.

## OKINAWA INSTITUTE OF SCIENCE AND TECHNOLOGY (OIST)

OIST is developing large-scale models of the cortex, the thalamus, and the basal ganglia using NEST in the context of a national strategic program on the K supercomputer.

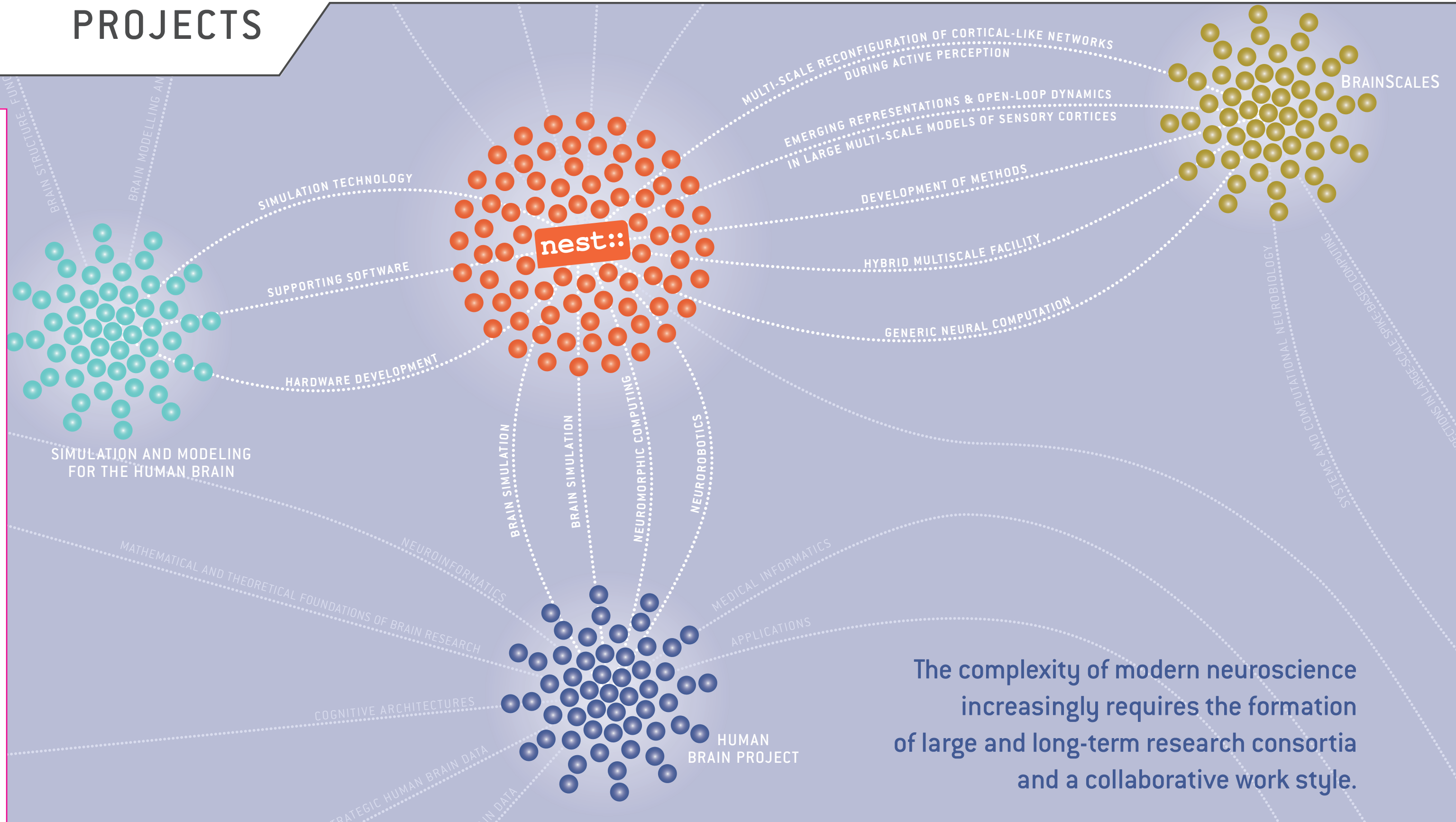
## ADVANCED INSTITUTE FOR COMPUTATIONAL SCIENCE (AICS)

The AICS in Kobe is collaborating with Forschungszentrum Jülich on the development of NEST for the K supercomputer and future exascale systems.

## FORSCHUNGSZENTRUM JÜLICH GMBH / RWTH AACHEN UNIVERSITY / JÜLICH AACHEN RESEARCH ALLIANCE

Neuroscientists at the Institute of Neuroscience and Medicine (INM) at Forschungszentrum Jülich use NEST for large-scale brain simulations on the world's biggest supercomputers. Together with neuroinformatics specialists at the INM and experts at the Jülich Supercomputing Centre (JSC) they develop high-performance algorithms and memory efficient data structures for NEST.

Under the umbrella of the Jülich Aachen Research Alliance (JARA), several cooperations between Forschungszentrum Jülich and RWTH Aachen University significantly contribute to the development of NEST. The collaboration with the High Performance Computing (HPC) Group at RWTH optimizes NEST for HPC systems. A collaboration between the Virtual Reality Group of RWTH and INM develops methods for interactive visualization of brain-scale simulation data. The Simulation Laboratory Neuroscience (SLNS) provides infrastructure for quality assurance and continuous integration for NEST.




**The complexity of modern neuroscience increasingly requires the formation of large and long-term research consortia and a collaborative work style.**

The NEST Initiative develops strategic plans with those consortia to enable collaboration on network models, ensures the availability of reliable and reproducible simulation technology, and safeguards the sustainability of associated investments.

## SIMULATION AND MODELING FOR THE HUMAN BRAIN

Since 2013 the Helmholtz Association has funded the portfolio theme "Supercomputing and Modeling for the Human Brain" (SMHB), which at the end of the initial 5 years funding will merge into the then fully operational new Jülich research program "Decoding the Human Brain" and the European Human Brain Project.

Helmholtz Portfolio Theme SMHB Supercomputing and Modeling for the Human Brain

NEST is the main simulation platform used and developed in the SMHB.

A strategic measure of the SMHB is the foundation of the Simulation Laboratory Neuroscience integrated into the national Bernstein Network Computational Neuroscience as the "Bernstein Facility for Simulation and Database Technology" in Germany.

## HUMAN BRAIN PROJECT

The EU Flagship Human Brain Project (HBP) started in October 2013.

Today, there is an explosion of data on the brain with 100,000 articles published in 2012 alone. However, there is no method available to integrate these data. The HBP plans to build up the technical infrastructure for modeling and simulating the whole brain. In strategy and scope this is like building the Very Large Telescope (VLT) of the ESO in Chile which enables us to reach out into previously unexplored territory. This requires the joint effort of many laboratories across Europe and hundreds of researchers.

For neuroscience where researcher are used to working in small groups the project is expected to induce a cultural change. NEST is the simulation engine developed and used in the HBP at the level of resolution of neurons and synapses.

Human Brain Project

## BRAINSCALES

BrainScaleS is a four-year multi-center and multidisciplinary EU-funded project investigating the brain's multi-scale nature to advance neuromorphic hardware and brain-inspired computing.

Research covers fields from electrophysiology and computational neuroscience to chip design and theoretical computer science. BrainScaleS and its precursor FACETS have introduced large-scale collaborative science into neuroscience, now feeding the Human Brain Project, and established Europe's leadership in neuromorphic computing.

BrainScaleS

In BrainScaleS, NEST is used for research on full-scale cortical network models as well as on abstract models of network dynamics and serves as the reference platform for the hardware systems HMF and SpiNNaker under development in Heidelberg and Manchester, respectively.

nest:: initiative

**The NEST Initiative was founded in 2001 to foster scientific exchange of computational neuroscience methods and to provide a framework for joint NEST development, support and documentation to benefit the scientific community.**

The NEST Initiative promotes NEST, e.g., through the NEST website and mailing lists, and distributes NEST code under open source licenses.

NEST's development is coordinated through a monthly developer video conference.

The NEST Initiative is incorporated as a non-profit organization in Ecublens, Switzerland, and is managed by a board of directors. Anyone who wants to support our work on NEST may join the Initiative as a community member. Active developers may become active members with full voting rights.

## PUBLICATION OF METHODS AND TECHNOLOGIES

The NEST Initiative promotes the development of a culture of collaborative neuroscience through scientific talks, publications, and the activities of its members. The transfer of computer science knowledge to neuroscience is a crucial aim.

In recent years, members of the NEST core development team have published many articles about novel methods and technologies for neural simulation. Prominent examples are the hybrid parallel and distributed execution of simulations, the user interface layer for NEST, the improvement of the performance and reduction of the memory usage, and suggestions for the description of neuronal network models in publications.

Currently, over 290 papers cite NEST as the simulator used in modeling studies.

## LICENSING

We have provided public releases of NEST's source code to the computational neuroscience community free of charge since 2004 under the proprietary NEST License.

Since 2012, NEST is available under the GNU General Public License version 2 or later as a consequence of popular demand by community members. The main benefit of this change is that the new license ensures the free availability of NEST and allows the users to re-distribute modified versions of NEST to fellow researchers.

In addition to NEST licensing, we support licensing discussions for other scientific tools to assure good availability and allow the inspection of the tools by others. These are important cornerstones to reproducible science which foster the reuse of code.

## USER SUPPORT

One important aspect of the success of NEST is its user community. 300+ users are subscribed to the user mailing list, where questions about using and extending NEST are discussed alongside general modeling questions.

NEST developers are subscribed to this list to provide high-quality answers and support users with problems that require a deeper insight into the NEST architecture. In recent years, however, more and more questions get answered by users themselves, which shows the liveliness of our community.

In addition to this interactive help, we provide a documentation index of the NEST simulator, which explains general modeling questions, and a list of examples that help users to get started quickly with NEST.

## TEACHING AND TUTORIALS

NEST is one of the core simulators taught at the main computational neuroscience summer schools, i. e. the Advanced Course for Computational Neuroscience (ACCN), the Okinawa Institute of Technology's Computational Neuroscience Course (OCNC), and the Latin American School on Computational Neuroscience (LASCON).


In addition, we give tutorials on advanced topics around NEST at the annual Computational Neuroscience Meeting (CNS) to provide users with updates on current developments and direct contact with NEST developers. Recent tutorials featured topics such as the general modeling approach in NEST, creating structured large-scale network models, and developing neuron and synapses models for NEST.

NEST is a high performance neuronal network simulator, developed with a focus on reproducibility and correctness.

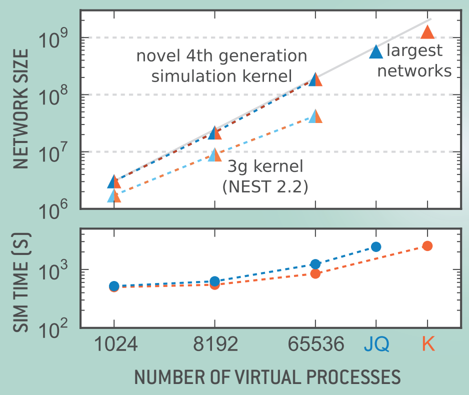
## PERFORMANCE AND MEMORY

To ensure the best possible use of available computing resources, we are constantly improving the performance and memory footprint of NEST. A considerable part of the recent software development has been directed towards the re-design of NEST in order to meet the memory challenges of brain-scale simulations on supercomputers, while keeping it slick and simple for laptop use (Helias et al. 2012, doi:10.3389/fninf.2012.00026).

The resulting 3rd generation simulation kernel (3g) as released with NEST 2.2 and the novel 4g kernel scale well in terms of memory usage and runtime on contemporary supercomputers such as the JUQUEEN BlueGene/Q machine (JQ, blue markers) and the K computer (red markers).

Profiling and modeling of NEST's resource consumption and run-time behavior are essential tools for this endeavour.

*Performance and memory consumption*




## TESTSUITE

We employ an extensive testsuite to ensure the correct compilation and installation of NEST on the user's computer.

Testing is a standard activity in computer science to make sure that all components of an application work as expected and that bogus user input leads to the correct error condition.

The testsuite of NEST consists of a set of hierarchical small unit tests, organized from simple to complex:

1. Test NEST's ability to report errors
2. Test that neurons and synapses have correct defaults
3. Test that neurons and synapses accept new parameters
4. Compare simulation results with analytical results
5. Test the parallelization features of NEST
6. Test the high-level user interface functions

In addition, we use regression tests to ensure that bugs that were fixed don't come back in newer versions.

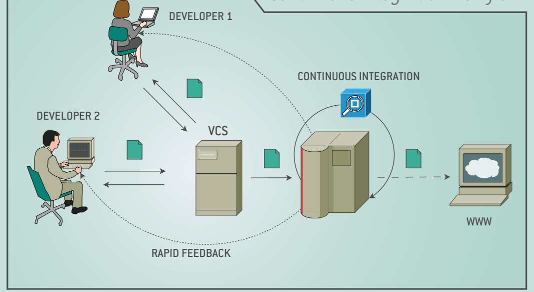
## CONTINUOUS INTEGRATION

To guarantee the quality of NEST development, we use continuous integration (CI), i.e. a practice where quality assurance (QA) is applied continuously as opposed to the traditional procedure, where QA is only applied during the integration phase after completing all development. CI decreases the risks associated with the integration by spreading required efforts over time, which helps to improve software quality and to reduce the time taken to deliver it.

A Jenkins-based CI infrastructure (Zaytsev and Morrison, 2012, doi:10.3389/fninf.2012.00031) helps with regular and automated testing of new commits and reports identified problems such as failed and broken tests in a timely fashion.

This technology also allows us to create regular public releases of NEST from the internal revision.

*Continuous integration workflow*



NEST is used for diverse applications in computational neuroscience and to investigate cortical function and dysfunction.

## REFERENCE FOR NEUROMORPHIC HARDWARE

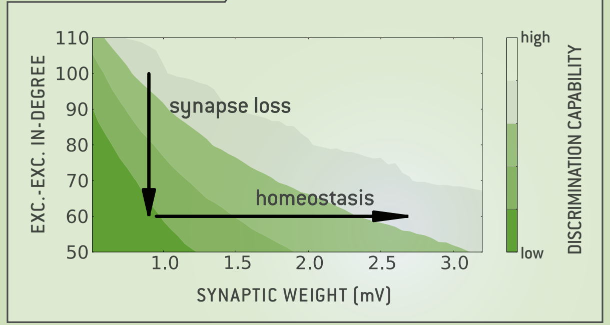
Brain-inspired hardware devices provide a novel substrate for simulating neural networks. These substrates can either use digital or analog circuits. Due to their highly parallelized architectures, these devices can surpass conventional computers in terms of speed and energy efficiency. However, difficulties arise both on the technical and conceptional level when leaving the realm of conventional computing architectures.

Harware-specific constraints can be implemented in NEST in order to investigate model dynamics at different scales under hardware-specific conditions. By pre-processing data or through online interactions (e.g. using the MUSIC library), NEST can enrich the repertoire of hardware simulations.

## NEURODEGENERATIVE DISEASES

Biologically realistic network models implementing brain function enable us to investigate the effects of neurodegenerative diseases on network dynamics and computational performance. This increases our knowledge about the relationship between disease-related changes at the cellular, dynamical and behavioural level.

*Classification performance*




With the help of NEST, Bachmann et al. (2013, doi:10.1186/1471-2202-14-S1-P282) have shown that homeostatic mechanisms can fully restore the sensitivity and classification performance of neural networks subject to a disruption of connectivity, a hallmark of Alzheimer's disease.

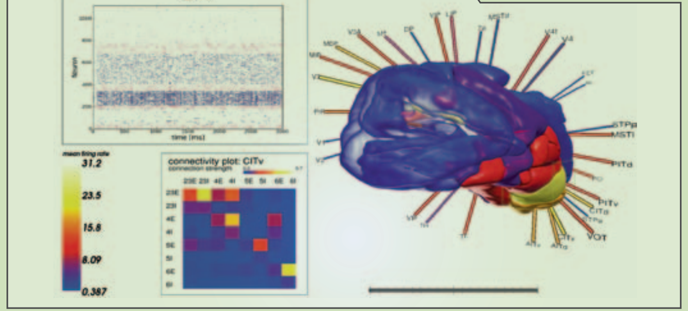
## CORTICAL MODELING

Cognitive and sensual processes in the mammalian cortex are supported by structured networks ranging from cellular circuits to sets of interacting areas.

NEST has been used to develop a full-scale model of a 1mm$^2$ patch of early sensory cortex (Potjans and Diesmann, 2012, doi:10.1093/cercor/bhs358). This model accounted for various aspects of spontaneous cortical activity.

To study large-scale interactions, the model was extended to all areas of the macaque visual cortex and has been successfully run with up to 20 million neurons connected via $2{\cdot}10^{11}$ synapses. To display and analyze the simulation results, an interactive visualization tool called VisNEST is under development in cooperation with the Virtual Reality Group of RWTH Aachen University.

*Simulation visualization*

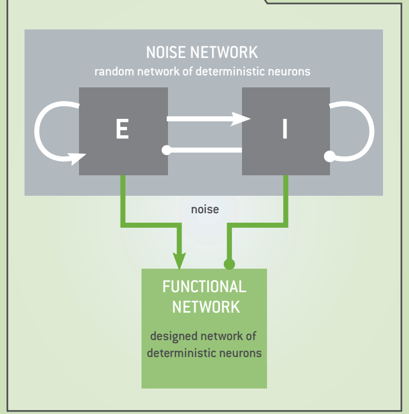



## BRAIN-INSPIRED COMPUTING

A central goal of neuroscience is to understand brain function and its realization in the neural hardware. Implementing functional algorithms using biologically realistic ingredients often constitutes a major challenge to computational neuroscientists. A number of powerful functional network models, for example, rely on the presence of uncorrelated background noise. So far, the biological origin of this noise remains unclear.

With the help of NEST and mathematical tools, Tetzlaff et al. (2012, doi:10.1371/journal.pcbi.1002596) have recently shown that inhibitory feedback, abundant in biological neural networks, can efficiently decorrelate neural network activity.

*Network sketch*




This effect allows deterministic neural networks to serve as a natural source of uncorrelated "noise" for functional neural architectures.

# USAGE EXAMPLE

*Simulation script*

```
import nest
import nest.raster_plot

J_ex = 0.1         # excitatory weight in mV
J_in = -0.5        # inhibitory weight in mV
p_rate = 20000.0   # external Poisson rate in Hz

neuron_params = {"C_m": 1.0, "tau_m": 20.0, "E_L": 0.0,
                 "V_reset": 0.0, "V_m": 0.0, "V_th": 20.0}

# Set parameters of neurons and devices
nest.SetDefaults("iaf_psc_delta", neuron_params)
nest.SetDefaults("poisson_generator", {"rate": p_rate})

# Create neurons and devices
nodes_ex = nest.Create("iaf_psc_delta", 10000)
nodes_in = nest.Create("iaf_psc_delta", 2500)
noise = nest.Create("poisson_generator")
espikes = nest.Create("spike_detector")

# Configure synapse models
nest.CopyModel("static_synapse", "excitatory", {"weight": J_ex})
nest.CopyModel("static_synapse", "inhibitory", {"weight": J_in})

# Connect the random net and instrument it with the devices
nest.DivergentConnect(noise, nodes_ex, model="excitatory")
nest.DivergentConnect(noise, nodes_in, model="excitatory")
nest.ConvergentConnect(nodes_ex[:50], espikes, model="excitatory")
nest.RandomConvergentConnect(nodes_ex, nodes_ex+nodes_in, 1000,
                             model="excitatory")
nest.RandomConvergentConnect(nodes_in, nodes_ex+nodes_in, 250,
                             model="inhibitory")

nest.Simulate(100)  # milliseconds

# Plot results
nest.raster_plot.from_device(espikes, hist=True)
nest.raster_plot.show()
```

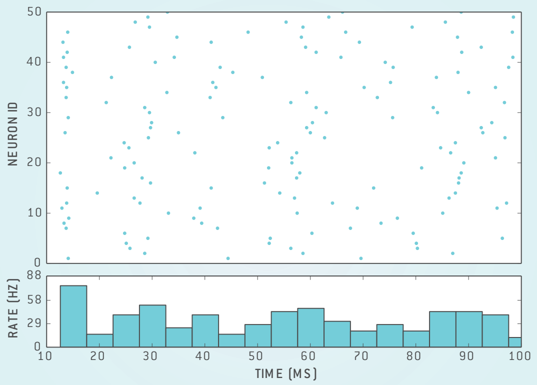
## NEST is a simulator for spiking neural network models such as models of visual or auditory information processing, models of learning and plasticity, or models of network activity dynamics.

A NEST simulation follows the logic of a laboratory electrophysiological experiment, but takes place inside a computer. The neural network is generated from neurons and their connections, which can either be static, or exhibit synaptic plasticity as observed for example during learning. NEST provides a large set of tested models for neurons and synapses as well as virtual devices that help stimulating the network and observing its response.

NEST is designed to optimally use the available computing hardware, from laptop computers to the world's largest supercomputers. NEST is extensible at many levels. As a module to the interpreted language Python, NEST can be combined with Python's many packages for scientific computing. Users with programming experience can also extend NEST's repertoire of neuron and synapse models.

The example script on the left simulates spiking activity in a sparse random network with recurrent excitation and inhibition. The figure to the right shows the spiking activity of 50 neurons as a raster plot. Time increases along the horizontal axis, neuron index increases along the vertical axis. Each dot corresponds to a spike of the respective neuron at a given time. The lower part of the figure shows a histogram with the mean firing-rate of the neurons.

*Simulation results*




*Key features of NEST*

**Neuron and synapse models:**

- INTEGRATE AND FIRE (IAF) NEURON MODELS with current- and conductance-based synapses
- ADAPTIVE-EXPONENTIAL IAF (Brette and Gerstner, 2005)
- MAT2 NEURON MODEL (Kobayashi et al., 2009)
- HODGKIN-HUXLEY TYPE NEURON MODELS with one compartment
- Simple MULTI-COMPARTMENT NEURON MODELS
- STATIC AND PLASTIC SYNAPSE MODELS (STDP, STP, dopamine)

**Accuracy and verification:**

- Each neuron model uses an APPROPRIATE SOLVER
- EXACT INTEGRATION is used for suitable neuron models
- GRID BASED NEURONAL INTERACTION and INTERACTION IN CONTINUOUS TIME
- EXTENSIVE TESTSUITE for continuous verification of the code base

**Interfaces:**

- PYTHON BASED USER INTERFACE (PyNEST)
- Built-in SIMULATION LANGUAGE INTERPRETER (SLI)
- TOPOLOGY MODULE for creating complex spatially structured networks
- Interface to the Multi Simulator Coordinator MUSIC
- Backend for the simulator-independent modeling tool PyNN

**Parallelization:**

- MULTI-THREADING to use multi-processor machines efficiently
- MPI-PARALLELISM to use computer clusters and supercomputers

JARA Jülich Aachen Research Alliance

The development of NEST is coordinated by the NEST Initiative, a non-for-profit organization open for interested scientists. NEST is available as free software under the terms of the GNU General Public License to ensure wide dissemination and allow researchers to adapt NEST to their needs.

More information about NEST as well as its source code is available on the homepage of the NEST Initiative.

**http://www.nest-initiative.org**


## ACKNOWLEDGEMENTS

Over the years, NEST's development has been supported by the Weizmann Institute, University Bochum, University & BCCN Freiburg, Honda Research Institute Europe, MPI for Fluid Dynamics, Norwegian University of Life Sciences, RIKEN Brain Science Institute, Helmholtz Association and Forschungszentrum Jülich, EPFL and BlueBrainProject, EU grants FACETS (FP6-15879), BrainScales (FP7-269921) and Human Brain Project (604102), Research Council of Norway grant eNeuro (178892/V30) and JARA.

Special thanks for contributing to this brochure go to Sacha van Albada, Michaela Bleuel, Paul Chorley, David Dahmen, Markus Diesmann, Jochen Martin Eppler, Marc-Oliver Gewaltig, Bernd Hentschel, Jakob J. Jordan, Susanne Kunkel, Abigail Morrison, Christian Nowke, Alexander Peyser, Hans Ekkehard Plesser, Martina Reske, Maximilian Schmidt, Jannis Schuecker, Tom Tetzlaff, and Yury V. Zaytsev.

In particular, we gratefully acknowledge the Jülich Aachen Research Alliance (JARA) for the realization of this brochure.


AN INITIATIVE OF

RWTH AACHEN UNIVERSITY

JÜLICH FORSCHUNGSZENTRUM