

The NeuroML ecosystem for standardised multi-scale modelling in neuroscience

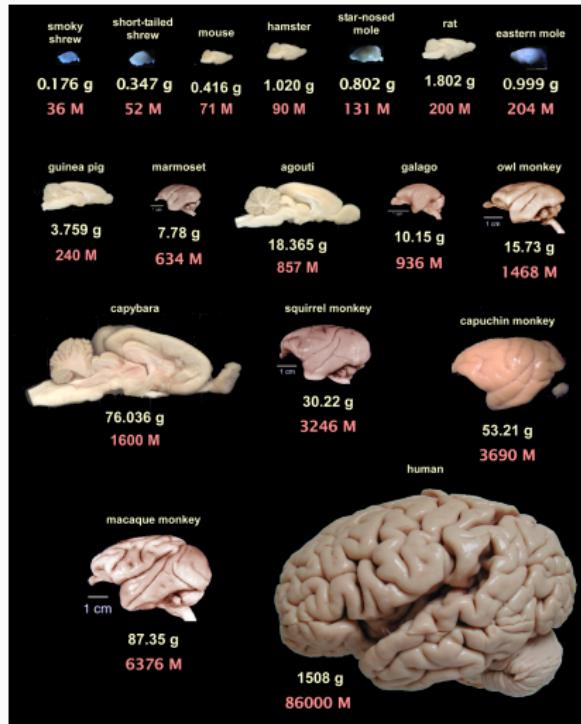
Ankur Sinha

Silver Lab

Department of Neuroscience, Physiology, & Pharmacology
University College London

2024-02-26

An understanding of the brain

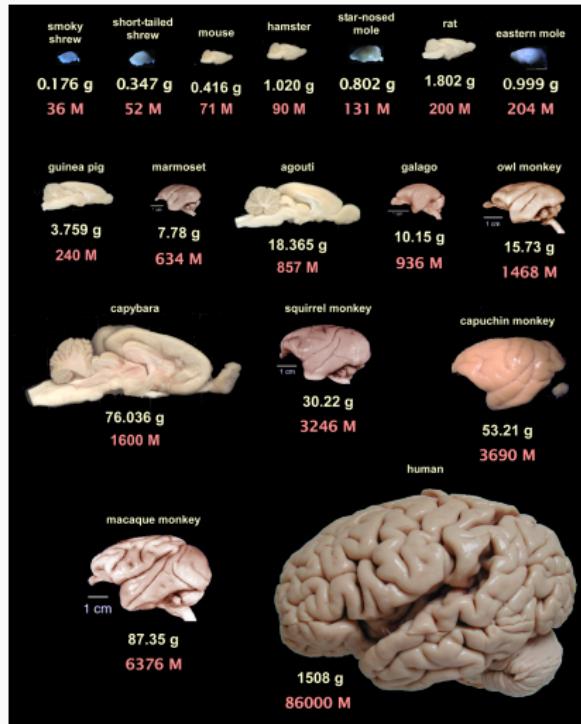


- ~86B neurons
- ~100T synapses
- also ~85B glia

¹ Herculano-Houzel, S. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience* 3, 31 (2009)

² von Bartheld, C. S. et al. The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting. *Journal of Comparative Neurology* 524, 3865–3895. ISSN: 1096-9861 (June 2016)

An understanding of the brain



- specialised circuits
- different neuronal types
- synaptic connections
- complex sub-cellular processes

¹ Herculano-Houzel, S. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience* 3, 31 (2009)

² von Bartheld, C. S. et al. The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting. *Journal of Comparative Neurology* 524, 3865–3895. ISSN: 1096-9861 (June 2016)

Models complement experimental neuroscience

Models are fully **observable, controllable**.

Models complement experimental neuroscience

Models are fully **observable, controllable**.

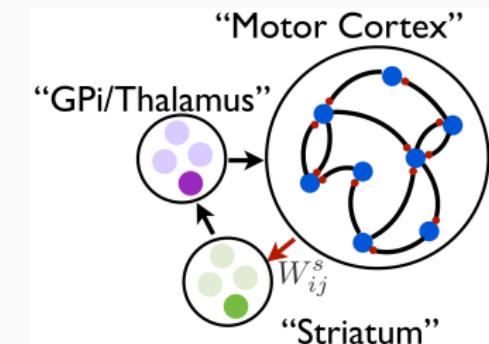
Models complement experimental neuroscience

Models are fully **observable, controllable**.

- Combine individual experimental results into **unified theories**
- Explore **generalisability** of experimental results over wider range of conditions
- **Generate** new experimentally testable, physically plausible hypotheses: **dictate experiment design**

Models: different scales

Models: different scales



¹ Murray, J. M. Local online learning in recurrent networks with random feedback. *eLife* 8 (eds Latham, P. et al.) e43299. ISSN: 2050-084X (2019)

¹ Schirmer, M. et al. Learning how network structure shapes decision-making for bio-inspired computing. *Nature Communications* 14. ISSN: 2041-1723 (May 2023)

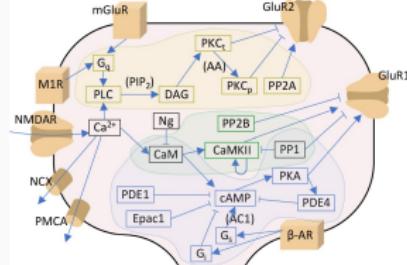
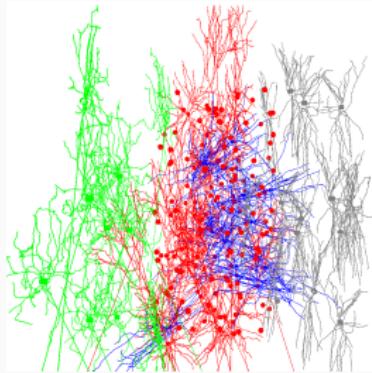
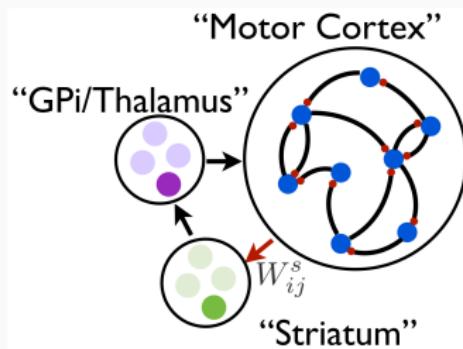
¹ Yao, H. K. et al. Reduced inhibition in depression impairs stimulus processing in human cortical microcircuits. *Cell Reports* 38. ISSN: 2211-1247.

<https://doi.org/10.1016/j.celrep.2021.110232> (Jan. 2022)

¹ Mäki-Marttunen, T. et al. A unified computational model for cortical post-synaptic plasticity. *eLife* 9 (eds Shouval, H. Z. et al.) e55714. ISSN: 2050-084X.

<https://doi.org/10.7554/eLife.55714> (July 2020)

Models: different scales



¹ Murray, J. M. Local online learning in recurrent networks with random feedback. *eLife* 8 (eds Latham, P. et al.) e43299. ISSN: 2050-084X (2019)

¹ Schirmer, M. et al. Learning how network structure shapes decision-making for bio-inspired computing. *Nature Communications* 14. ISSN: 2041-1723 (May 2023)

¹ Yao, H. K. et al. Reduced inhibition in depression impairs stimulus processing in human cortical microcircuits. *Cell Reports* 38. ISSN: 2211-1247.

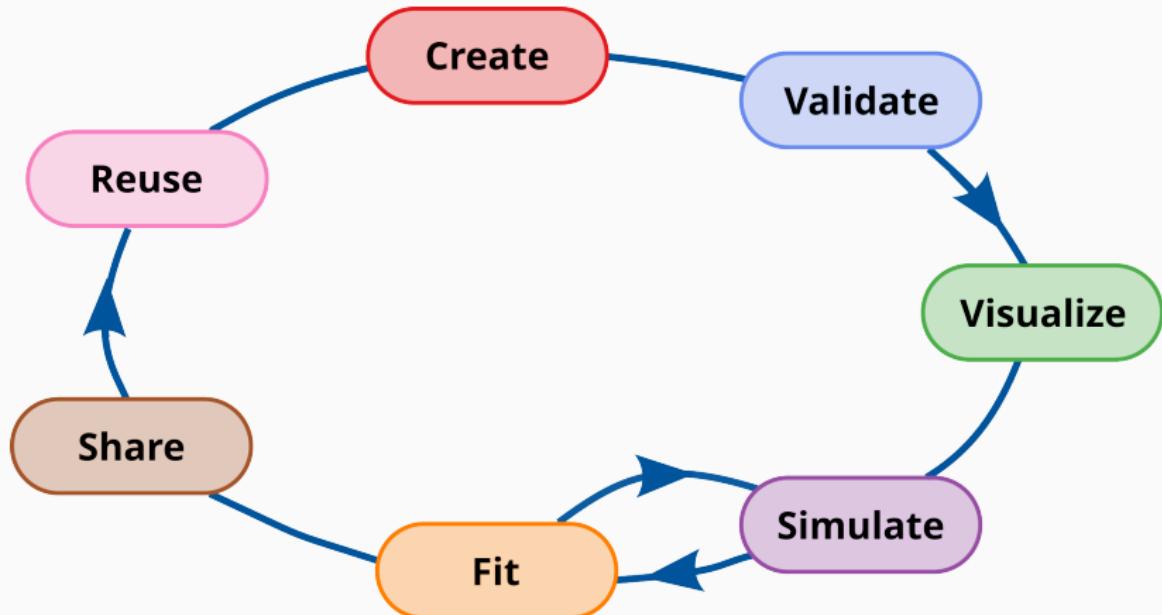
<https://doi.org/10.1016/j.celrep.2021.110232> (Jan. 2022)

¹ Mäki-Marttunen, T. et al. A unified computational model for cortical post-synaptic plasticity. *eLife* 9 (eds Shouval, H. Z. et al.) e55714. ISSN: 2050-084X.

<https://doi.org/10.7554/eLife.55714> (July 2020)

A *mechanistic* understanding of the brain
requires biophysically detailed modelling

The model life cycle



Computational modelling software ecosystem is fragmented

- many specialist tools:
 - NEURON, NEST, Brian, GENESIS, MOOSE, STEPS, ANNarchy, TVB, LFPy, NeuroLib, EDEN, Arbor, NetPyNE...

Computational modelling software ecosystem is fragmented

- many specialist tools:
 - NEURON, NEST, Brian, GENESIS, MOOSE, STEPS, ANNarchy, TVB, LFPy, NeuroLib, EDEN, Arbor, NetPyNE...
- **but:**
 - different APIs, syntax:
 - increased difficulty for users

Computational modelling software ecosystem is fragmented

- many specialist tools:
 - NEURON, NEST, Brian, GENESIS, MOOSE, STEPS, ANNarchy, TVB, LFPy, NeuroLib, EDEN, Arbor, NetPyNE...
- **but:**
 - different APIs, syntax:
 - increased difficulty for users
 - not well defined model descriptions:
 - models cannot be easily validated

Computational modelling software ecosystem is fragmented

- many specialist tools:
 - NEURON, NEST, Brian, GENESIS, MOOSE, STEPS, ANNarchy, TVB, LFPy, NeuroLib, EDEN, Arbor, NetPyNE...
- **but:**
 - different APIs, syntax:
 - increased difficulty for users
 - not well defined model descriptions:
 - models cannot be easily validated
 - custom machine readable internal representations:
 - models cannot be easily inspected/analysed

Computational modelling software ecosystem is fragmented

- many specialist tools:
 - NEURON, NEST, Brian, GENESIS, MOOSE, STEPS, ANNarchy, TVB, LFPy, NeuroLib, EDEN, Arbor, NetPyNE...
- **but:**
 - different APIs, syntax:
 - increased difficulty for users
 - not well defined model descriptions:
 - models cannot be easily validated
 - custom machine readable internal representations:
 - models cannot be easily inspected/analysed
 - ad-hoc utilities:
 - cannot be used with all tools

Makes computational neuroscience models
less
FAIR
(Findable, Accessible, Interoperable, Reusable)

Standards enable FAIR neuroscience



International Neuroinformatics
Coordinating Facility



COMBINE

¹ Abrams, M. B. et al. A Standards Organization for Open and FAIR Neuroscience: the International Neuroinformatics Coordinating Facility. *Neuroinformatics* 20, 25–36. ISSN: 1559-0089. <https://doi.org/10.1007/s12021-020-09509-0> (2022); <https://incf.org/>

¹ Computational Modeling in Biology NEtwork (COMBINE): <https://combine.org/>

Standards enable FAIR neuroscience



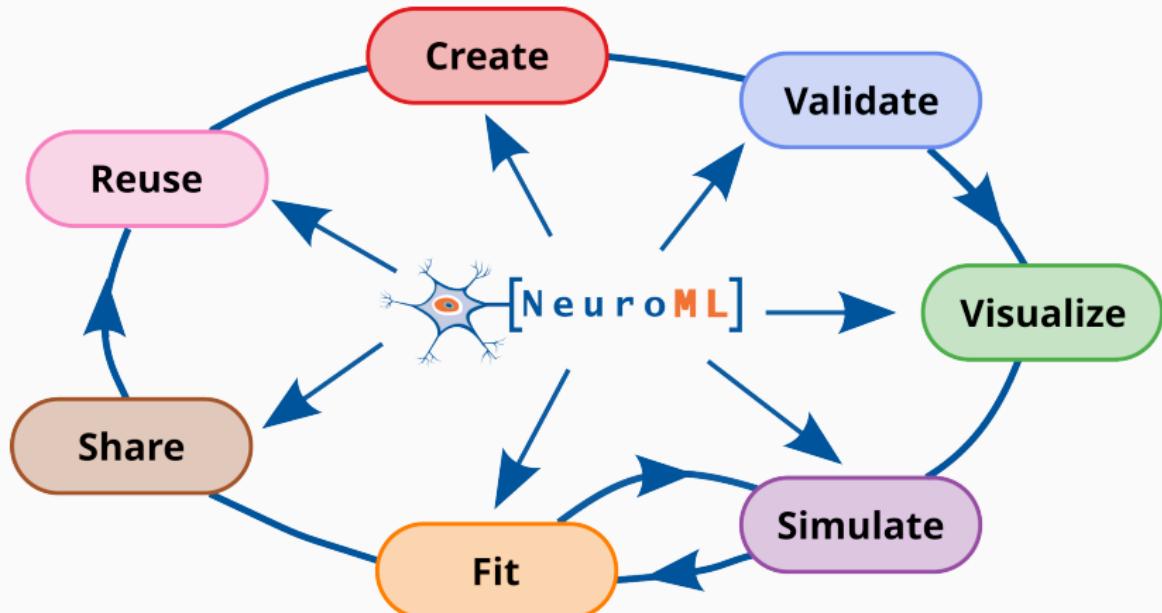
COMBINE



¹ Abrams, M. B. et al. A Standards Organization for Open and FAIR Neuroscience: the International Neuroinformatics Coordinating Facility. *Neuroinformatics* 20, 25–36. ISSN: 1559-0089. <https://doi.org/10.1007/s12021-020-09509-0> (2022); <https://incf.org/>

¹ Computational Modeling in Biology NEtwork (COMBINE): <https://combine.org/>

NeuroML ecosystem supports all stages of the model cycle



NeuroML ecosystem

- standard/specification
- software ecosystem

Model specification (**schema**: XSD)

- elements
- attributes
- hierarchical relationships

Model specification (**schema**: XSD)

- elements
- attributes
- hierarchical relationships

Dynamics (**LEMS component type definitions**)

- dynamical behaviour

NeuroML standard: schema: XSD

Way of specifying the structure of an XML document.

- allows defining **types** and **extensions/restrictions** on types to create new types.
- allows generation of **APIs**

¹<https://www.w3.org/TR/xmlschema-1/>

NeuroML standard: schema: XSD

Way of specifying the structure of an XML document.

- allows defining **types** and **extensions/restrictions** on types to create new types.
- allows generation of **APIs**

*A model description can be validated against the schema
before simulation*

¹<https://www.w3.org/TR/xmlschema-1/>

NeuroML standard: schema: XSD

```
<xs:simpleType name="Nml2Quantity_voltage"> <!-- For params with dimension voltage -->
  <xs:restriction base="xs:string">
    <xs:pattern value="-?([0-9]*(\.[0-9]+)?)([eE]-?[0-9]+)?[\s]*(V|mV)"/>
  </xs:restriction>
</xs:simpleType>
```

¹ Izhikevich, E. M. *Dynamical systems in neuroscience*. (MIT Press, 2007)

NeuroML standard: schema: XSD

```
<xs:simpleType name="Nml2Quantity_voltage"> <!-- For params with dimension voltage -->
  <xs:restriction base="xs:string">
    <xs:pattern value="-?([0-9]*([.][0-9]+)?)([eE]-?[0-9]+)?[\s]*(V|mV)"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="Izhikevich2007Cell">
  <xs:annotation>
    <xs:documentation>Cell based on ...</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="BaseCellMembPotCap">
      <xs:attribute name="v0" type="Nml2Quantity_voltage" use="required"/>
      <xs:attribute name="k" type="Nml2Quantity_conductancePerVoltage" use="required"/>
      <xs:attribute name="vr" type="Nml2Quantity_voltage" use="required"/>
      <xs:attribute name="vt" type="Nml2Quantity_voltage" use="required"/>
      <xs:attribute name="vpeak" type="Nml2Quantity_voltage" use="required"/>
      <xs:attribute name="a" type="Nml2Quantity_pertime" use="required"/>
      <xs:attribute name="b" type="Nml2Quantity_conductance" use="required"/>
      <xs:attribute name="c" type="Nml2Quantity_voltage" use="required"/>
      <xs:attribute name="d" type="Nml2Quantity_current" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

¹ Izhikevich, E. M. *Dynamical systems in neuroscience*. (MIT Press, 2007)

NeuroML standard: LEMS component type definitions

Low Entropy Model Specification language

- domain independent
- allows creation of "Component Types" (**classes**) from which "Components" (**objects**) can be instantiated by providing the necessary parameters
- provides a **reference implementation/simulator**

¹ Cannon, R. C. et al. LEMS: a language for expressing complex biological models in concise and hierarchical form and its use in underpinning NeuroML 2. *Frontiers in Neuroinformatics* 8 (2014)

NeuroML standard: LEMS component type definitions

Low Entropy Model Specification language

- domain independent
- allows creation of "Component Types" (**classes**) from which "Components" (**objects**) can be instantiated by providing the necessary parameters
- provides a **reference implementation/simulator**
- machine readable: **translatable** into other formats

¹ Cannon, R. C. et al. LEMS: a language for expressing complex biological models in concise and hierarchical form and its use in underpinning NeuroML 2. *Frontiers in Neuroinformatics* 8 (2014)

NeuroML standard: dynamics (LEMS)

```
<ComponentType name="izhikevich2007Cell" extends="baseCellMembPotCap"
  description="Cell based ...>

  <Parameter name="v0" dimension="voltage" description="Initial membrane potential"/>

  <!--
  Defined in baseCellMembPotCap:
  <Parameter name="C" dimension="capacitance"/>
  -->
  <Parameter name="k" dimension="conductance_per_voltage"/>

  <Parameter name="vr" dimension="voltage" description="Resting membrane potential"/>
  <Parameter name="vt" dimension="voltage" description="Spike threshold"/>
  <Parameter name="vpeak" dimension="voltage" description="Peak action potential value"/>

  <Parameter name="a" dimension="per_time" description="Time scale of recovery variable u"/>
  <Parameter name="b" dimension="conductance" description="Sensitivity of recovery variable u to subthreshold
  ↪ fluctuations of membrane potential v"/>
  <Parameter name="c" dimension="voltage" description="After-spike reset value of v"/>
  <Parameter name="d" dimension="current" description="After-spike increase to u"/>

  <Attachments name="synapses" type="basePointCurrent"/>

  <Exposure name="u" dimension="current" description="Membrane recovery variable"/>

  <Dynamics><!-- snipped --></Dynamics>

</ComponentType>
```

NeuroML standard: XSD and LEMS

XSD:

```
<xs:attribute name="v0" type="Nml2Quantity_voltage" use="required"/>
<xs:attribute name="k" type="Nml2Quantity_conductancePerVoltage" use="required"/>
<xs:attribute name="vr" type="Nml2Quantity_voltage" use="required"/>
<xs:attribute name="vt" type="Nml2Quantity_voltage" use="required"/>
<xs:attribute name="vpeak" type="Nml2Quantity_voltage" use="required"/>
<xs:attribute name="a" type="Nml2Quantity_pertime" use="required"/>
<xs:attribute name="b" type="Nml2Quantity_conductance" use="required"/>
<xs:attribute name="c" type="Nml2Quantity_voltage" use="required"/>
<xs:attribute name="d" type="Nml2Quantity_current" use="required"/>
```

LEMS:

```
<Parameter name="v0" dimension="voltage" description="Initial membrane potential"/>
<Parameter name="k" dimension="conductance_per_voltage"/>
<Parameter name="vr" dimension="voltage" description="Resting membrane potential"/>
<Parameter name="vt" dimension="voltage" description="Spike threshold"/>
<Parameter name="vpeak" dimension="voltage" description="Peak action potential value"/>
<Parameter name="a" dimension="per_time" description="Time scale of recovery variable u"/>
<Parameter name="b" dimension="conductance" description="Sensitivity of recovery variable u to subthreshold
    fluctuations of membrane potential v"/>
<Parameter name="c" dimension="voltage" description="After-spike reset value of v"/>
<Parameter name="d" dimension="current" description="After-spike increase to u"/>
```

NeuroML standard: dynamics (LEMS)

```
<ComponentType name="izhikevich2007Cell" extends="baseCellMembPotCap"
  description="Cell based ..."
  <!-- snipped -->
  <Attachments name="synapses" type="basePointCurrent"/>

  <Exposure name="u" dimension="current" description="Membrane recovery variable"/>

  <Dynamics>
    <StateVariable name="v" dimension="voltage" exposure="v"/>
    <StateVariable name="u" dimension="current" exposure="u"/>

    <DerivedVariable name="iSyn" dimension="current" exposure="iSyn" select="synapses[*]/i" reduce="add" />

    <DerivedVariable name="iMemb" dimension="current" exposure="iMemb" value="k * (v-vr) * (v-vt) + iSyn - u"/>

    <TimeDerivative variable="v" value="iMemb / C"/>
    <TimeDerivative variable="u" value="a * (b * (v-vr) - u)"/>

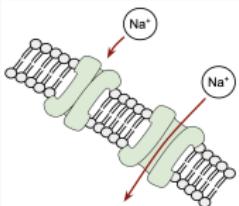
    <OnStart>
      <StateAssignment variable="v" value="v0"/>
      <StateAssignment variable="u" value="0"/>
    </OnStart>

    <OnCondition test="v .gt. vpeak">
      <StateAssignment variable="v" value="c"/>
      <StateAssignment variable="u" value="u + d"/>
      <EventOut port="spike"/>
    </OnCondition>

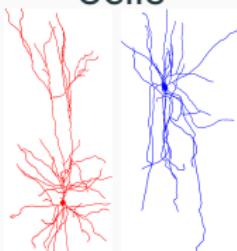
  </Dynamics>
</ComponentType>
```

NeuroML is declarative, modular, structured, hierarchical

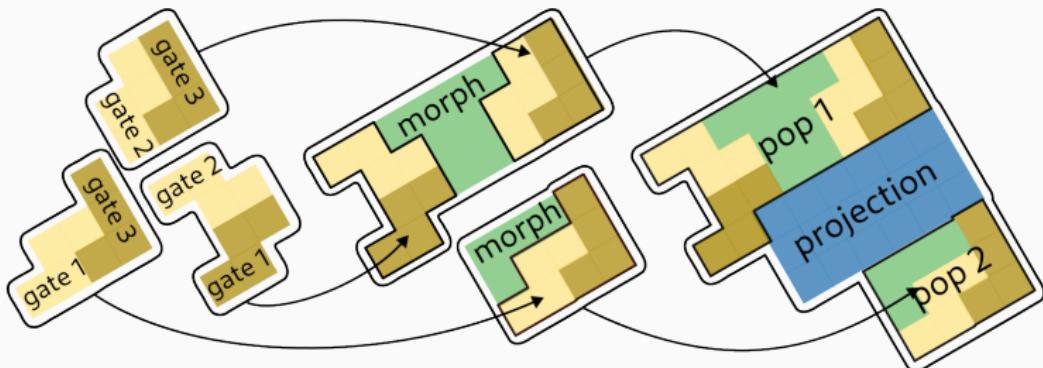
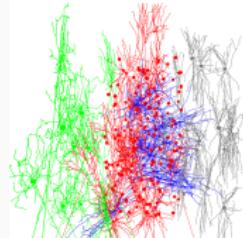
Conductances



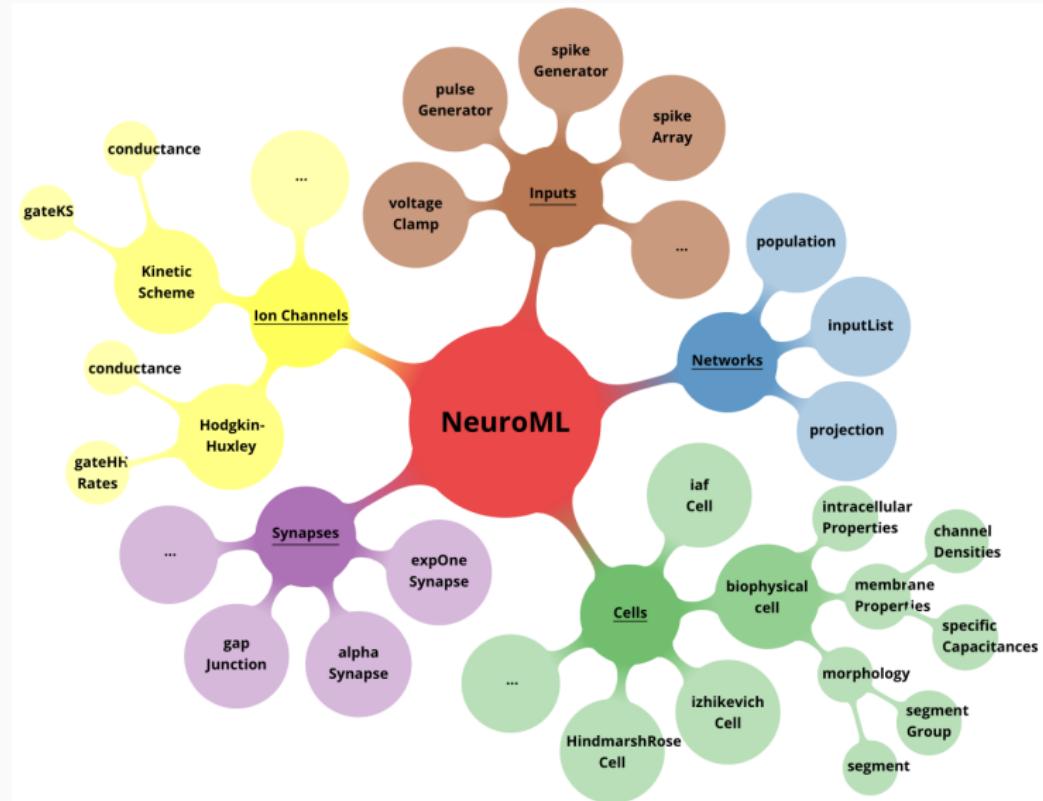
Cells



Networks

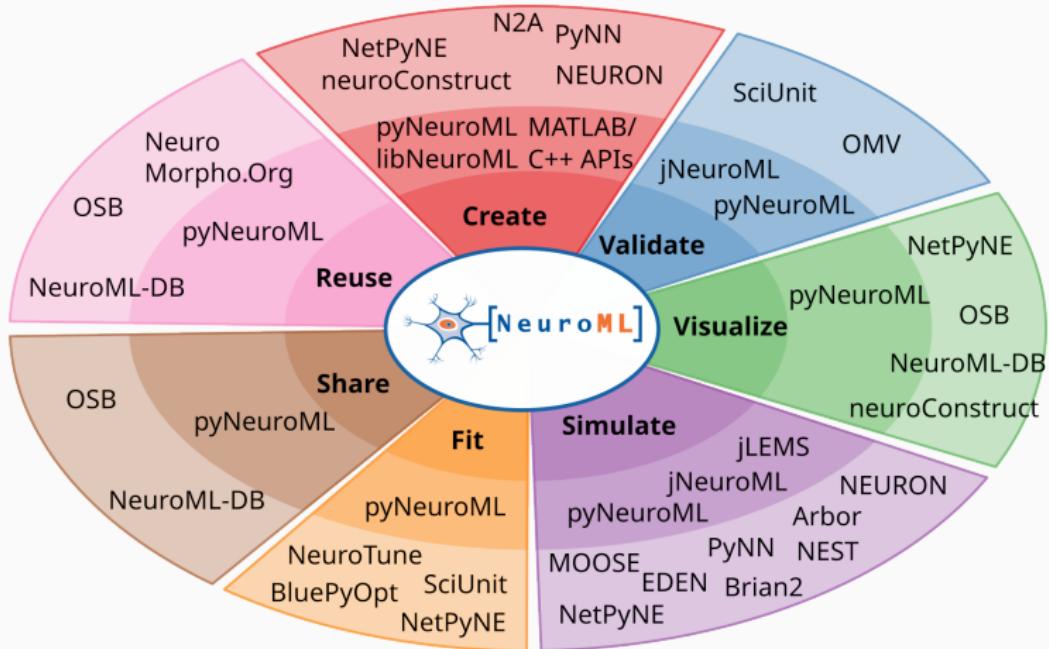


NeuroML provides users with a set of curated model elements



¹ Full standard is at: <https://docs.neuroml.org/Userdocs/Specification.html>

NeuroML software ecosystem



NeuroML software ecosystem: core tools

pyNeuroML

Python interface to NeuroML. Gives access to all jNeuroML functionality, adds helper methods for summarising, visualizing and analyzing NeuroML

libNeuroML

Python API for reading, writing and validating NeuroML

pyLEMS

Python implementation of LEMS parser & simulator

jNeuroML

Can validate & simulate NeuroML models & convert to other formats, e.g. NEURON

NeuroML2 LEMS Definitions

Define structure/behavior of NeuroML core types in LEMS

jLEMS

Can load and simulate LEMS models - not neuroscience specific

```
pip install pyneuroml
```

NeuroML software ecosystem: pyNeuroML

```
# validation
validate_neuroml2("file.nml")
doc.validate(recursive=True)

# inspection
element.info()
summary(doc)
nml2_to_png(doc)
nml2_to_svg(doc)
generate_nmlgraph(doc)

# visualisation/analysis
plot_2D(cell)
plot_interactive_3d(cell)
plot_interactive_3d(network)

plot_channel_densities(cell)
plot_time_series(file)

# simulation
run_lems_with_jneuroml("sim.xml")
run_lems_with_jneuroml_neuron("sim.xml")
run_on_nsg("jneuroml_neuron", "sim.xml")

# sharing
create_combine_archive("sim.xml")
```

```
$ pynml "file.nml" -validate
$ pynml-summary "file.nml"
$ pynml -png "file.nml"
$ pynml -svg "file.nml"
$ pynml "file.nml" -graph
$ pynml "file.nml" -matrix 1
$ pynml-plotmorph "cell.nml"
$ pynml-plotmorph -i "cell.nml"
$ pynml-plotmorph -i "network.nml"
$ pynml-channelanalysis "channel.nml"
$ pynml-plotchan "channel.nml"
$ pynml-plotspikes "sim.xml"
$ pynml-plottimeseries "sim.xml"
$ pynml-plottimeseries "*.dat"

$ pynml "siml.xml"
$ pynml "siml.xml" -neuron -run
$ pynml-archive "file.xml"
```

NeuroML: creating/simulating models

Python script to create a new network, and validate it:

```
from neuroml import * # NeuroML API libNeuroML

newdoc = NeuroMLDocument(id="new_doc")
newcell = IafTauCell(id="cell_0", leak_reversal="-60mV", thresh="0mV", tau="5ms", reset="-70mV")
newdoc.add(newcell)

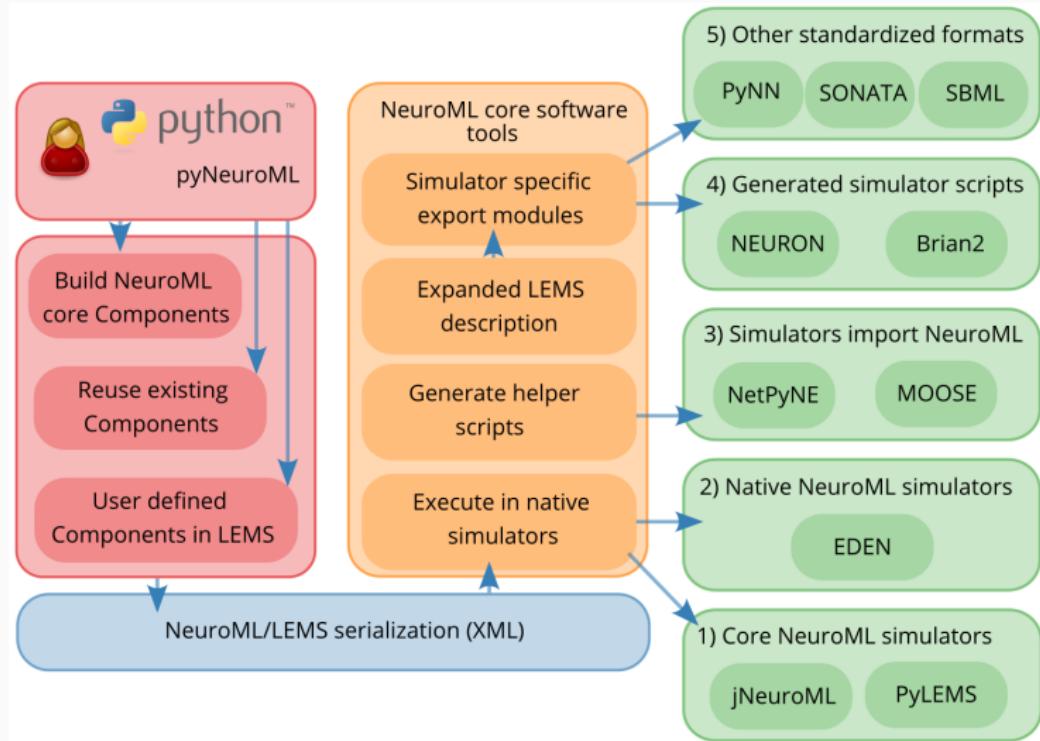
network = newdoc.add(Network, id="new_net", validate=False)
population = network.add(Population, id="new_pop", size=10, component=newcell.id)

# Helper method to ensure all parameters
# present and appropriate
newdoc.validate(recursive=True)
```

Resultant NeuroML XML serialization:

```
<neuroml id="new_doc">
  <iafTauCell id="cell_0" leakReversal="-60mV" thresh="0mV" reset="-70mV" tau="5ms"/>
  <network id="new_net">
    <population id="new_pop" component="cell_0" size="10"/>
  </network>
</neuroml>
```

NeuroML: creating/simulating models



NeuroML: validating models

Model description

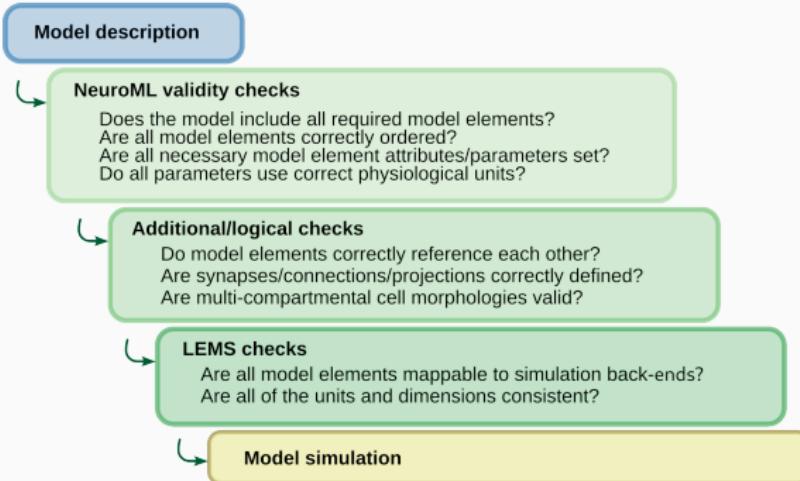
NeuroML validity checks

- Does the model include all required model elements?
- Are all model elements correctly ordered?
- Are all necessary model element attributes/parameters set?
- Do all parameters use correct physiological units?

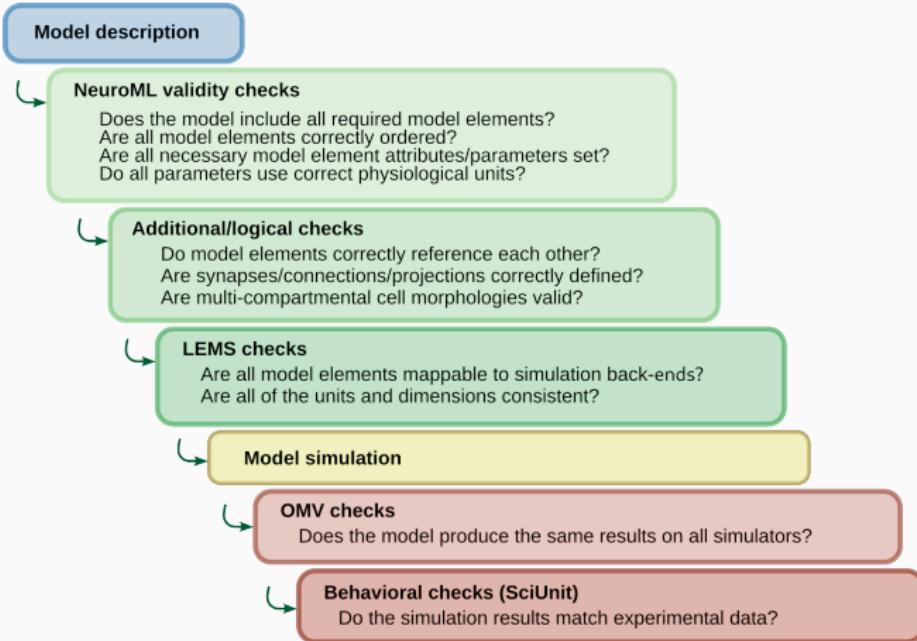
Additional/logical checks

- Do model elements correctly reference each other?
- Are synapses/connections/projections correctly defined?
- Are multi-compartmental cell morphologies valid?

NeuroML: validating models



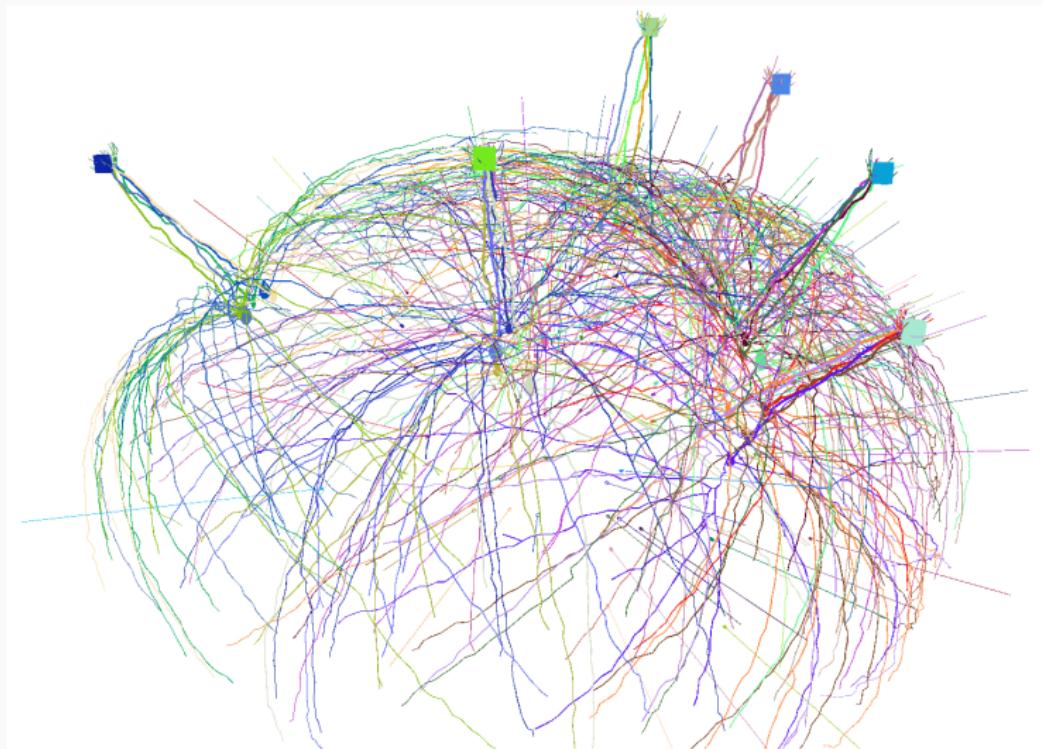
NeuroML: validating models



¹<https://github.com/OpenSourceBrain/osb-model-validation>

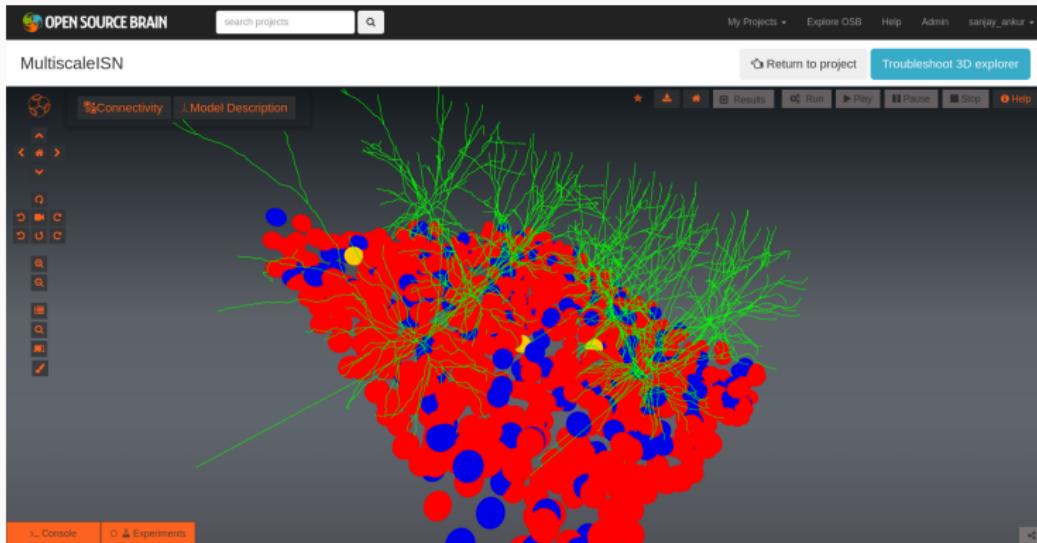
¹<https://sciunit.io>

NeuroML: visualising/analysing models



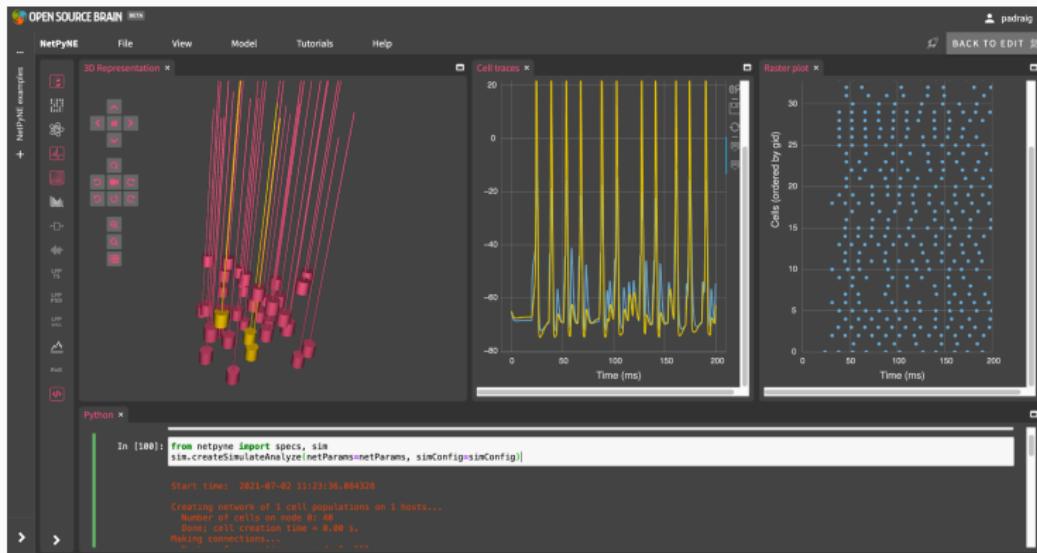
¹ 3D interactive visualisation of Migliore et al. [10] using pynml-plotmorph

NeuroML: visualising/analysing models



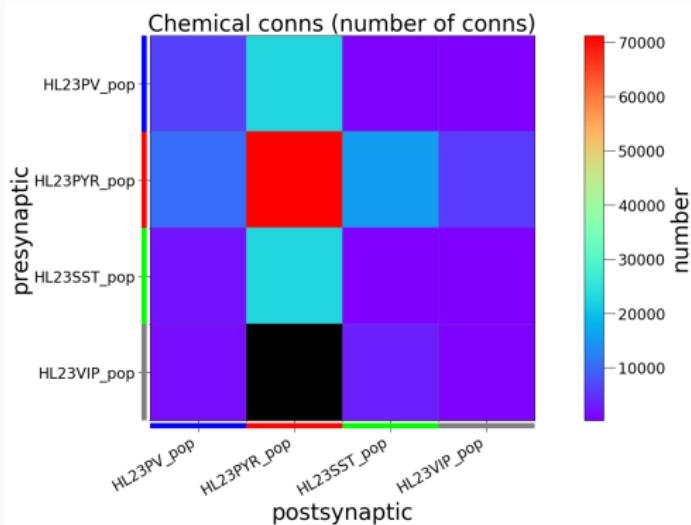
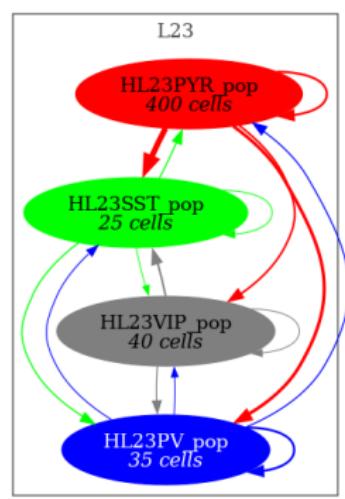
¹ 3D interactive visualisation of Sadeh et al. [11] on Open Source Brain: <https://vi.opensourcebrain.org>

NeuroML: visualising/analysing models



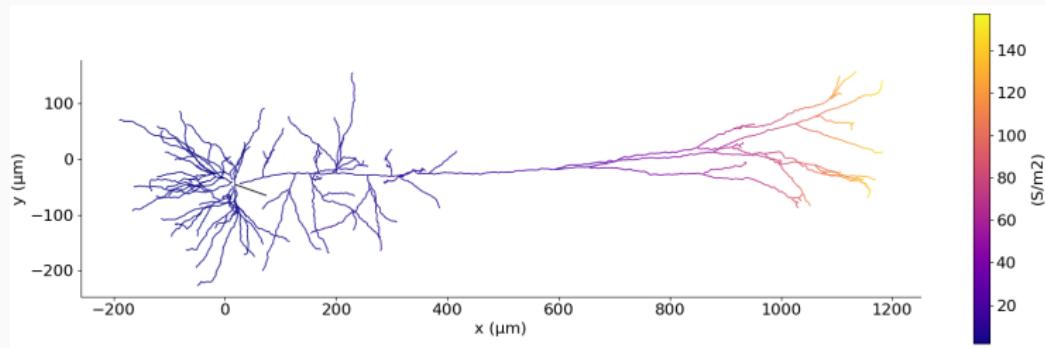
¹ 3D interactive visualisation using NetPyNE-UI on Open Source Brain v2: <https://opensourcebrain.org>

NeuroML: visualising/analysing models



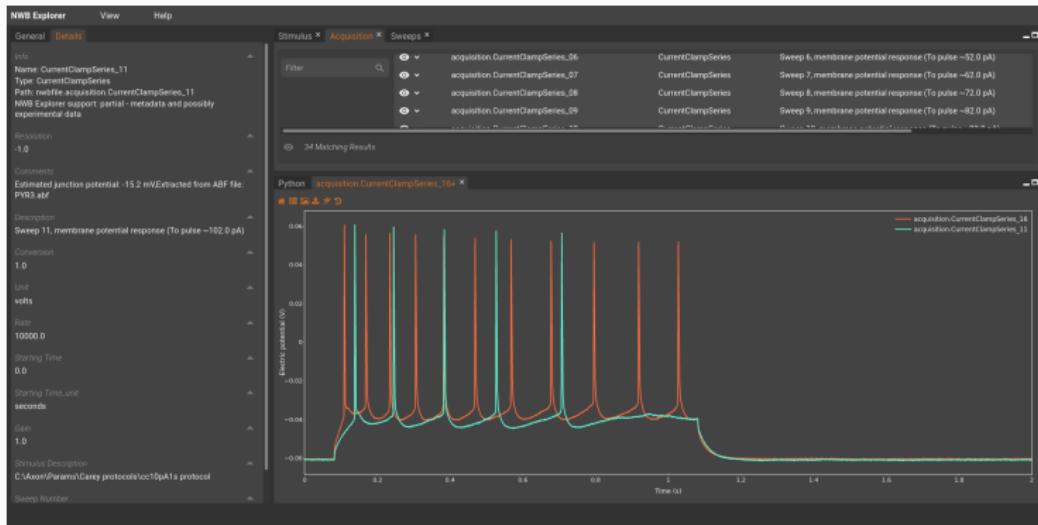
¹ Connectivity metrics for NeuroML conversion of Yao et al. [5]

NeuroML: visualising/analysing models



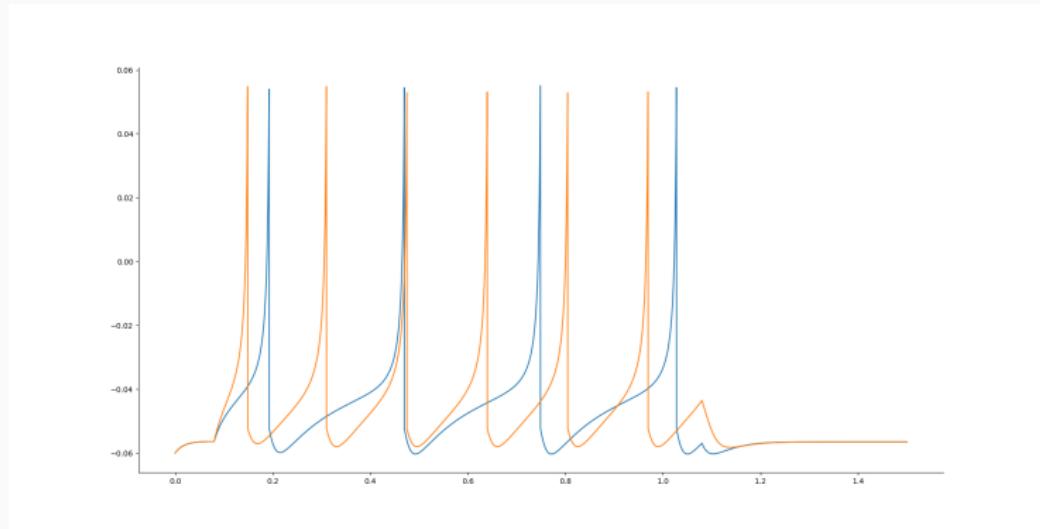
¹ Automated visualisation of ionic conductance density on a multi-compartmental cell

NeuroML: fitting models



¹ Visualising Lanore et al. [12] in NWB Explorer on Open Source Brain

NeuroML: fitting models



¹ Izhikevich cells fitted using NeuroML fitting pipeline: NeuroTune, using Inspyred

NeuroML: sharing and re-using models

The screenshot shows the GitHub interface for the 'OpenSourceBrain' organization. The top navigation bar includes links for Overview, Repositories (2.0k), Projects, Packages, Teams (17), People (72), and a search bar. Below the navigation is a sidebar titled 'Repositories' with categories: All (selected), Public, Private, Sources, Forks, Archived, Mirrors, and Templates. A search bar labeled 'sort: name' is present. The main content area displays a grid of repository cards. Each card contains the repository name, a brief description, the programming language (indicated by a colored dot), the number of stars, and the number of forks. The repositories listed include:

- WilsonCowan: Wilson-Cowan model (Python, 10 stars, 2 forks)
- WeilerEtAl08-LaminarCortex: Models based on data in: Weiler N, Wood L, Yu J, Solla SA, Shepherd GM (2008) Top-down laminar organization ... (Python, 0 stars, 0 forks)
- WangBuzsaki1996: Model from: Wang XJ, Buzsaki G (1996) Gamma oscillation by synaptic inhibition in a hippocampal interneuronal network m... (AMPL, 3 stars, 4 forks)
- VogelsSpraklerEtAl2011: Network from Vogels et al. Science 2011 (Python, 1 star, 3 forks)
- VierlingClaassenEtAl2010: Network model from Engaging distinct oscillatory neocortical circuits, Vierling-Claassen et al 2010 (XSLT, 0 stars, 0 forks)
- VervaekeEtAl-GolgiCellNetwork: GitHub repository for an OSB project containing a cerebellar Golgi cell based on Vervaeke et al 2010 (XSLT, 1 star, 2 forks)
- VERTEXShowcase: An OSB showcase project to create examples of export from & import to the VERTEX simulator (MATLAB, 1 star, 0 forks)
- VINetworkModels: Network models of primary visual cortex (VI) based on spiking point neurons (Python, 8 stars, 4 forks)
- tutorials: OSB tutorials (HTML, 1 star, 1 fork)
- TobinEtAl2017: Cell models from "Wiring variations that enable and constrain neural computation in a sensory microcircuit" Tobin et al. 2017, e... (XSLT, 0 stars, 4 forks)
- TheVirtualBrainShowcase: (Python, 2 stars, 0 forks)
- Thalamocortical: GitHub repository for an OSB project containing a thalamocortical network model based on Trajkov et al. 2005 (XSLT, 9 stars, 12 forks)

¹ Standardized models on Github: Open Source Brain: <https://github.com/OpenSourceBrain>

NeuroML: sharing and re-using models

OPEN SOURCE BRAIN

Explore OSB Help Sign in Sign up

Modelling the brain, together

Open Source Brain is a resource for sharing and collaboratively developing computational models of neural systems.

Learn more about the OSB interface Learn about the Hodgkin Huxley model Simulate electrophysiologically detailed cell models Explore more OSB projects



Or create an account to add your own models and run simulations!

[Sign up](#) [Sign in](#) [?](#)

To add Open Source Brain version 2 now!

¹ Standardized models on Open Source Brain v1: <https://v1.opensourcebrain.org>

NeuroML: sharing and re-using models

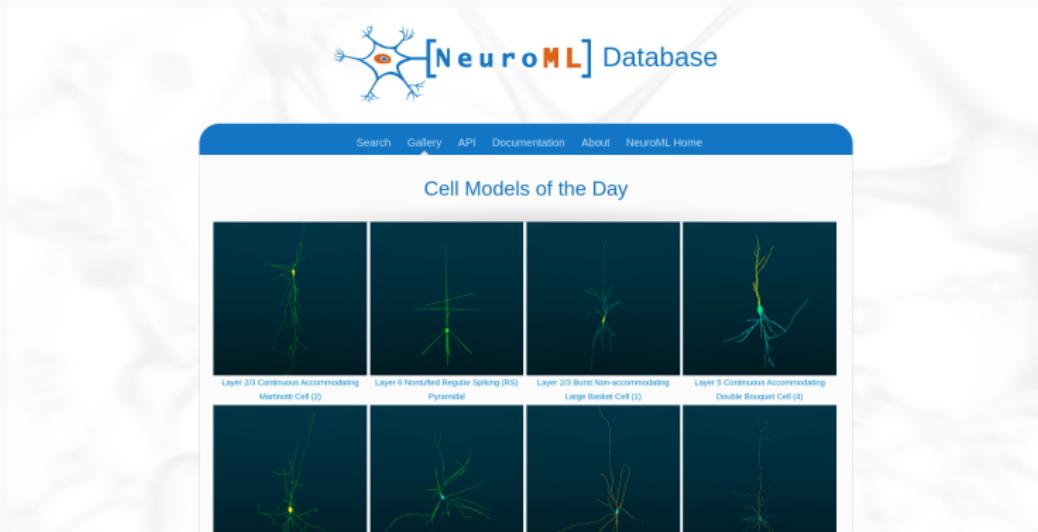
The screenshot shows the Open Source Brain v2 interface. On the left is a sidebar with 'DASHBOARD' and 'Workspaces' sections, and 'INFO & SUPPORT' with links to 'Main site', 'About', 'Guided tour', 'Documentation', and 'Chat'. A 'Create new' button is at the bottom of the sidebar. The main area is titled 'ALL REPOSITORIES 2024' and lists four repositories:

- WormNeuroAtlas** by Padraig, GitHub, modeling, OpenWorm, C. elegans, main, Open Details
- M1-channelopathies-OSB** by Salvadordura@gmail.com, GitHub, modeling, main, Open Details
- Thalamocortical network with epilepsy** by Osbadmin, GitHub, modeling, NetPyNE, Epilepsy, master, Open Details. A note says: 'Thalamocortical computer model. NetPyNE version of a model of a thalamocortical...' and a 'See more' link.
- Spiking neuronal networks performing motor control** by Osbadmin, GitHub, modeling, NetPyNE, master, Open Details. A note says: 'Artificial neural networks (ANNs) have been successfully trained to perform a w...' and a 'See more' link.
- Spiking neuronal network model of visual-motor cortex playing a virtual racket-ball game** by Osbadmin, GitHub, modeling, NetPyNE, master, Open Details. A note says: 'Recent models of spiking neuronal networks have been trained to perform behav...' and a 'See more' link.

At the bottom are navigation icons for back, forward, search, and other functions.

¹ Standardized models on Open Source Brain v2: <https://v2.opensourcebrain.org>

NeuroML: sharing and re-using models



¹ Standardized models on NeuroML-DB: <https://neuroml-db.org>

NeuroML: community: events

Home People Standards Events Documents Examples Contact  

Conference Date
8-11 April, 2024

Conference Location
UCL, London, UK

Important Dates

Breakouts and tutorials submission deadline
11 March, 2024

Notification of acceptance for breakout and tutorials
18 March, 2024

Lightning talk and poster submission deadline
11 March, 2024

Notification of acceptance for lightning talks and posters
18 March, 2024

Travel support applications **NOW CLOSED**

Notification of travel support awards
21 February, 2024

the  **combinE**
computational modeling in biology network

The "Computational Modeling in Biology" Network (COMBINE) is an initiative to coordinate the development of the various community standards and formats in systems biology, synthetic biology and related fields. HARMONY is a codefest-type meeting, with a focus on development of the standards, interoperability and infrastructure. There are generally not many general discussions or oral presentations during HARMONY; instead, the time is devoted to allowing hands-on hacking and interaction between people focused on practical development of software and standards.

HARMONY 2024 will be held at **University College London**

Local organizers are Sarah Keating (s.keating@ucl.ac.uk) and Padraig Gleeson, (p.gleeson@ucl.ac.uk).

Workshop Location



¹<https://co.mbine.org/events/>

NeuroML: community: events

Home Calendar Blog About OCNS Meetings Membership Publications Working groups Get help

33rd Annual Computational Neuroscience Meeting
CNS*2024, Natal, Brazil



**CNS
2024**

33rd Annual Computational Neuroscience Meeting
Natal, Rio Grande do Norte
July, 20-24 2024
Brazil

CNS*2024 Important Dates

CNS*2024 Program

- Attendee Info
 - Register
 - Add to registration
 - Submit an abstract
 - Registration Fees
 - General & Areas

July 20-24, 2024

Register for CNS*2024

CNS*2024 Important Dates

Visa and travel information for Brazil

Quick Links

- FAQ
- CNS*2024
- Comp-Neuro Mailing List
- INCF: NeuroStars
- INCF: Training space
- OCNS on YouTube
- OCNS on Twitter
- OCNS on Facebook
- OCNS on LinkedIn

Upcoming Events

No events

[View Full Calendar](#)

Announcements

News • **CNS*2024 Abstract Submission** is now open!
Submission deadline: March 11, 2024.

News • **CNS*2024 Registration** is now open! Discounted pricing for early registrants.

¹<https://www.cnsorg.org>

NeuroML: projects: GSoC

- Open source, cross simulator, large scale network models in NeuroML and PyNN
- Implementation of SWC to NeuroML converter in PyNeuroML
- Incorporate new features into an advanced, cross-platform 3D viewer for NeuroML cells and networks

<https://summerofcode.withgoogle.com/programs/2024/organizations/incf>

NeuroML: closing the neuroscience research loop with OSB

- Open Source Brain Video

NeuroML: resources

Sinha, A. et al. **The NeuroML ecosystem for standardized multi-scale modeling in neuroscience.** *bioRxiv*. eprint:

<https://www.biorxiv.org/content/early/2023/12/11/2023.12.07.570537.full.pdf>. <https://www.biorxiv.org/content/early/2023/12/11/2023.12.07.570537> (2023)(in review)

<https://docs.neuroml.org>

<https://opensourcebrain.org>