

# **The NeuroML ecosystem for standardised multi-scale modelling in neuroscience**

---

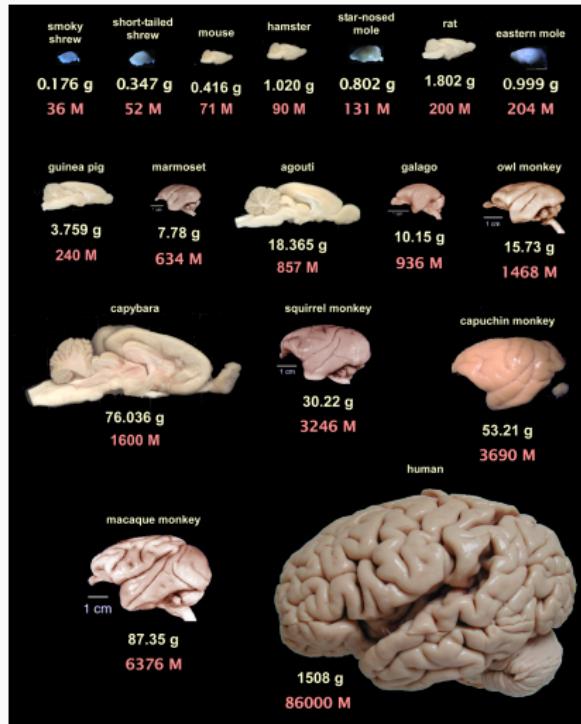
Ankur Sinha

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

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

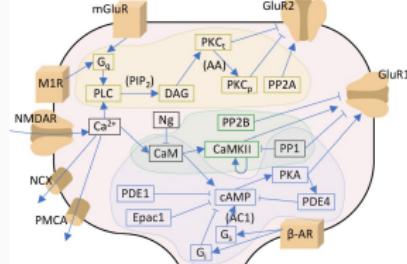
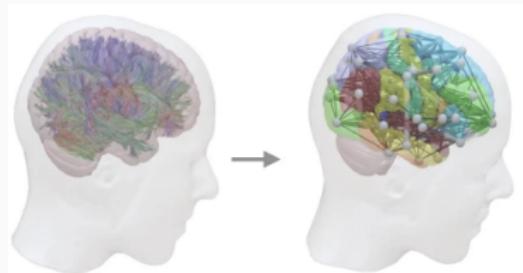
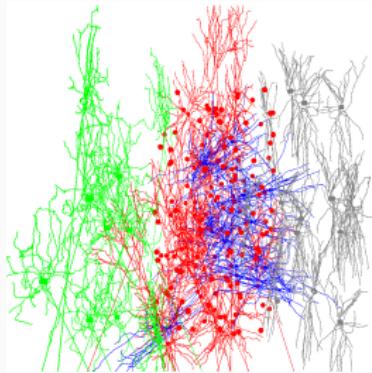
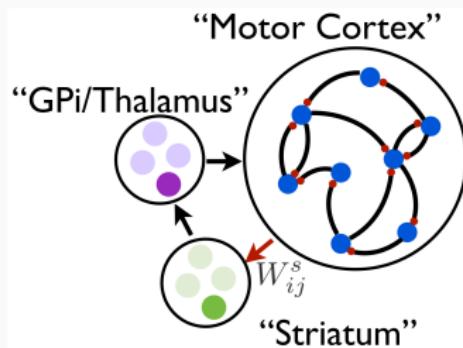
<sup>2</sup>von Bartheld, C. S. et al. The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting. *Journal of Comparative Neurology* 524, 3865–3895. ISSN: 1096-9861 (June 2016)

## Models complement experimental neuroscience

Models are fully **observable, controllable**.

- 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



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

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

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

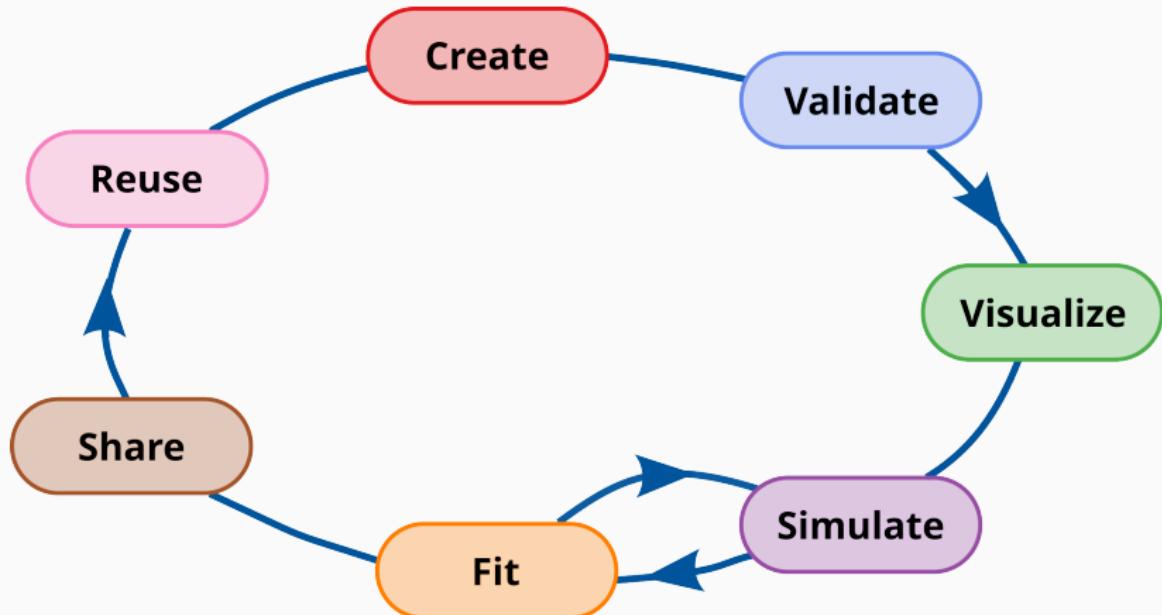
<https://doi.org/10.1016/j.celrep.2021.110232> (Jan. 2022)

<sup>1</sup> 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...
- **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



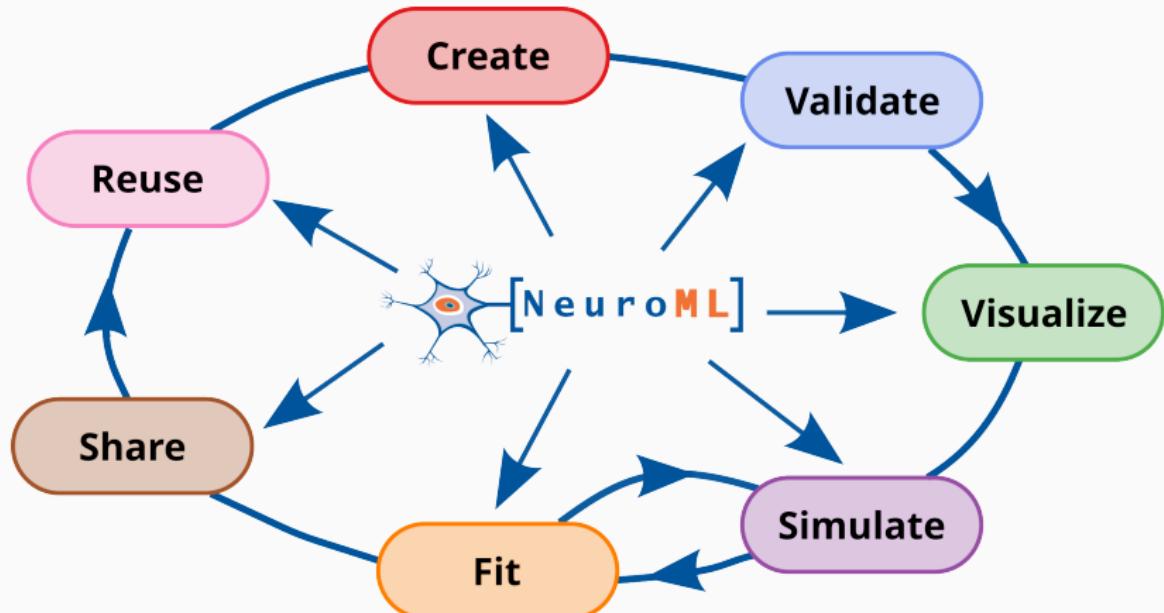
COMBINE



<sup>1</sup> 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/>

<sup>1</sup> 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

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**

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

<sup>1</sup><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>

<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>
```

<sup>1</sup> 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**
- machine readable: **translatable** into other formats

<sup>1</sup> 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)"/>

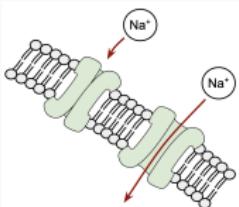
    <OnStart>
      <StateAssignment variable="v" value="v0"/>
      <StateAssignment variable="u" value="0"/>
    </OnStart>

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

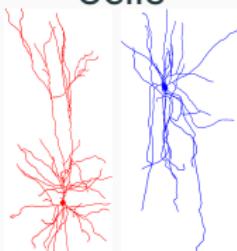
  </Dynamics>
</ComponentType>
```

# NeuroML is declarative, modular, structured, hierarchical

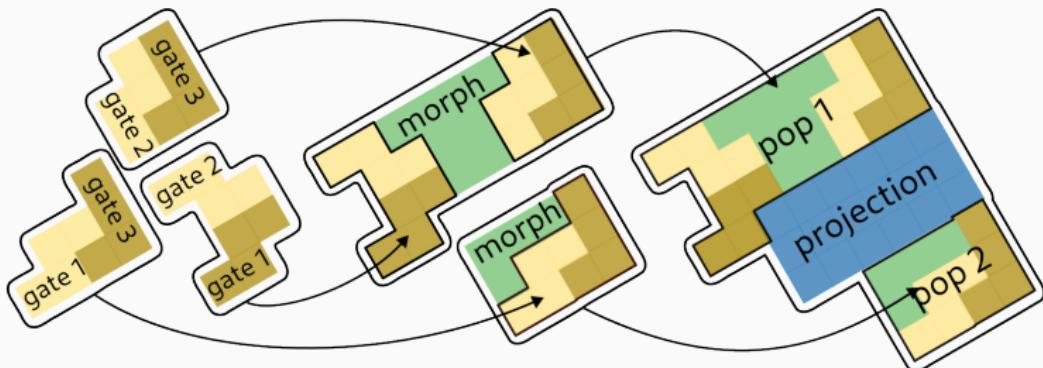
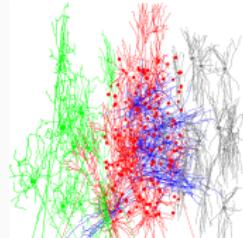
Conductances



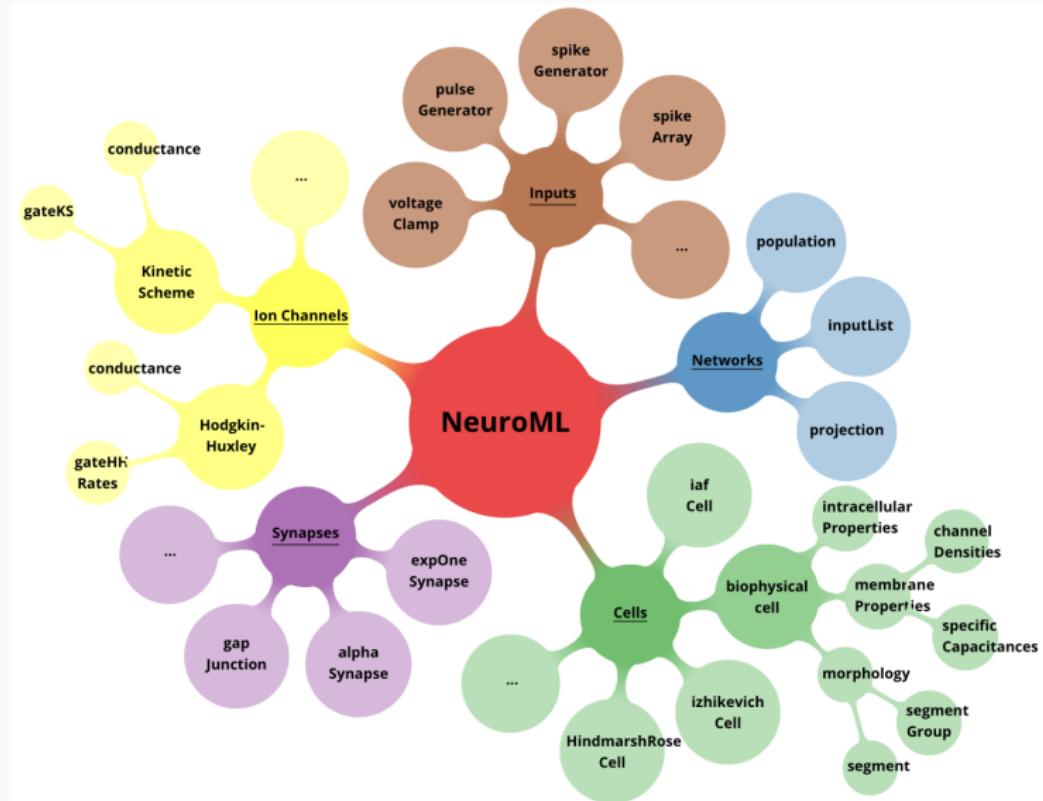
Cells



Networks

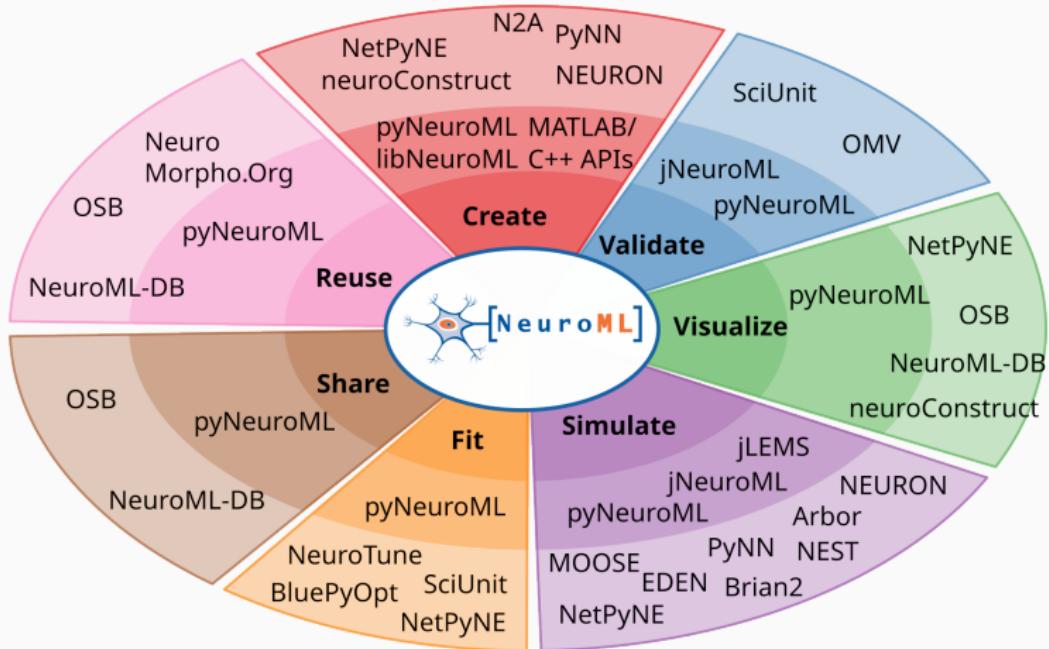


# NeuroML provides users with a set of curated model elements



<sup>1</sup> 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)

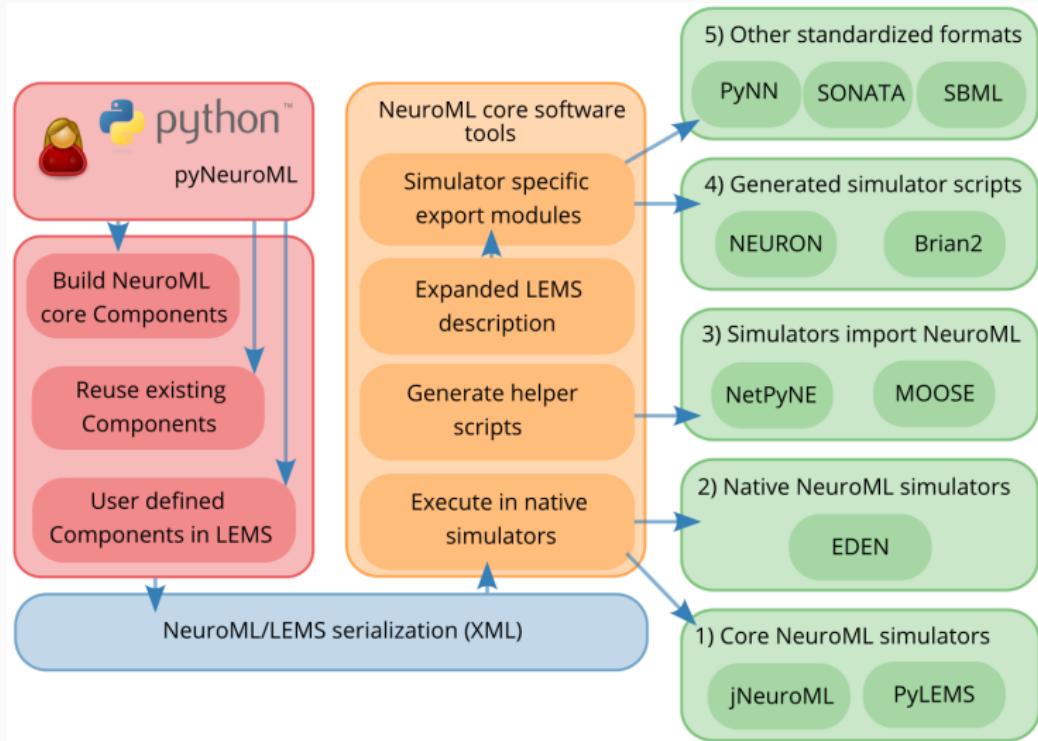
network = newdoc.add(Network, id="new_net", validate=False)
population = network.add(Population, id="new_pop", size=10, component=newcell.id)

# 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?

## 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?

### LEMS checks

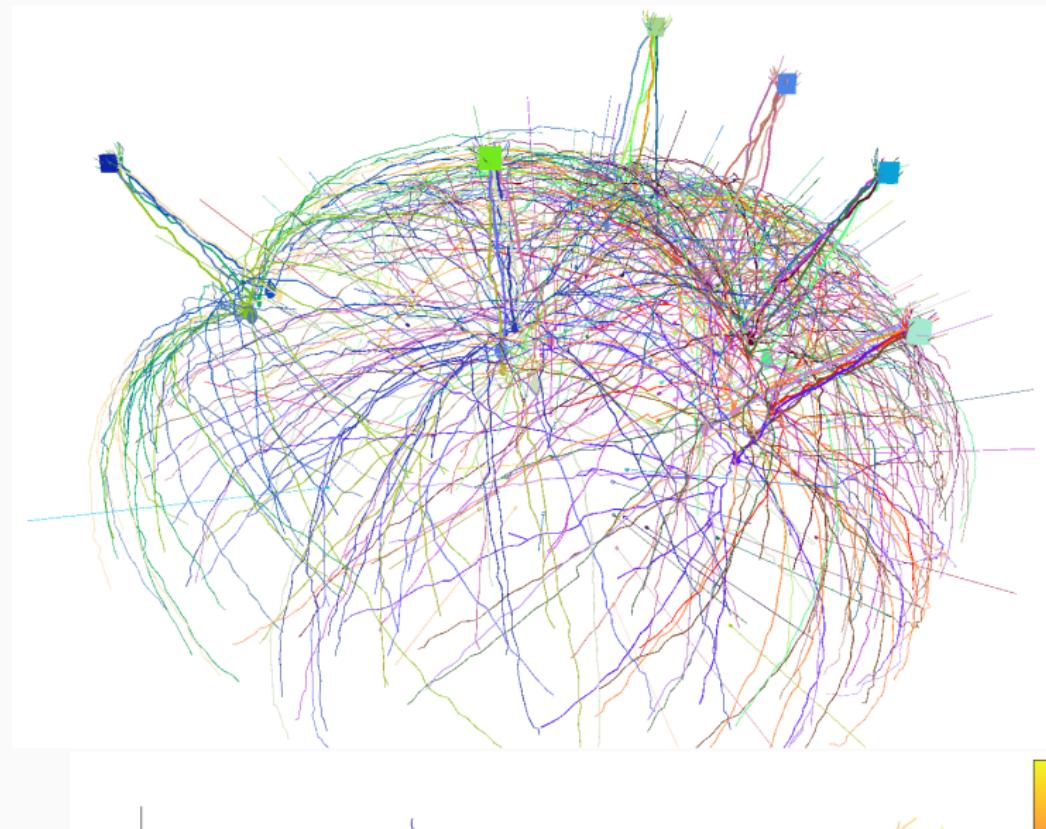
- Are all model elements correctly ordered?
- Are all of the units consistent?

### Model simulation

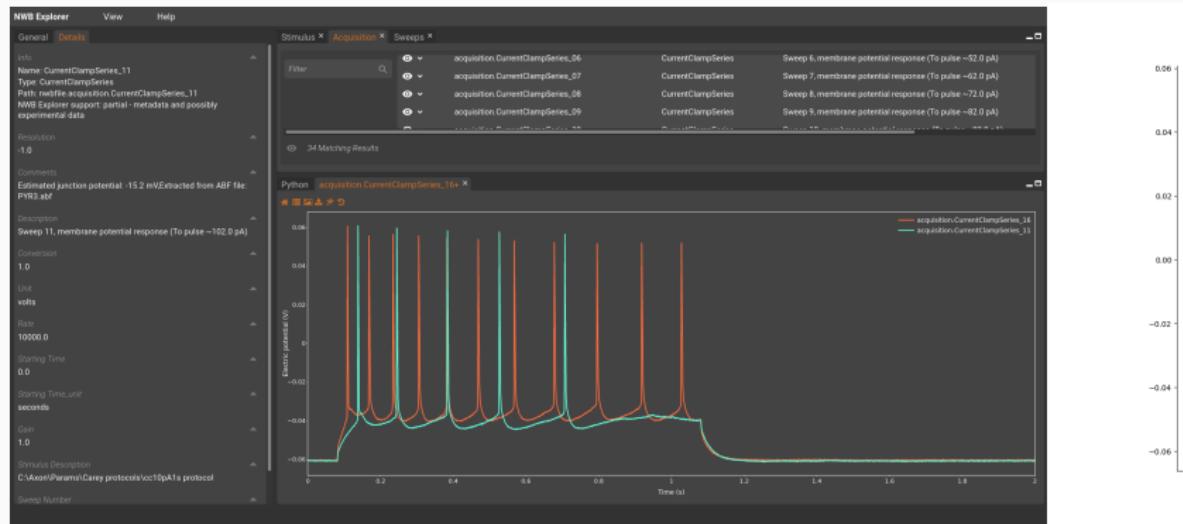
<sup>1</sup><https://github.com/OpenSourceBrain/osb-model-validation>

<sup>1</sup><https://sciumit.io>

# NeuroML: visualising/analysing models



# NeuroML: fitting models



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

<sup>1</sup> Izhikevich cells fitted using NeuroML fitting pipeline: NeuroTune, using Inspyred

# NeuroML: sharing and re-using models

The screenshot shows the 'Repositories' section of the Open Source Brain website. The left sidebar includes links for All, Public, Private, Sources, Forks, Archived, Mirrors, and Templates. The main area displays a table of 1k+ repositories, each with a title, description, language, number of stars, and a star icon. A search bar at the top right allows users to search for specific repositories.

Name	Language	Stars
WilsonCowan	Jupyter Notebook	2
WeilerEtAl08-LaminarCortex	Python	0
WangBuzsaki1996	AMPL	3
VogelsPockelerEtAl2011	Python	1
VierlingClaassenEtAl2010	XSCLT	0
VervaekeEtAl-GolgiCellNetwork	XSCLT	1
VERTEXShowcase	MATLAB	1
VI Network Models	Python	8
tutorials	HTML	1
TobinEtAl2017	XSCLT	0
TheVirtualBrainShowcase	Python	2
Thalamocortical	XSCLT	9

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

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

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

<sup>1</sup> 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](mailto:s.keating@ucl.ac.uk)) and Padraig Gleeson, ([p.gleeson@ucl.ac.uk](mailto:p.gleeson@ucl.ac.uk)).

**Workshop Location**



<sup>1</sup><https://combine.org/events/>

<sup>1</sup><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>