

The NeuroML ecosystem for standardised multi-scale modelling in neuroscience

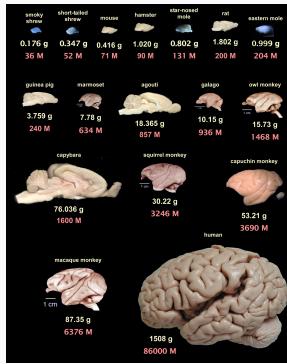
Ankur Sinha
Silver Lab
Department of Neuroscience, Physiology, & Pharmacology
University College London

2024-02-26

Notes

1 / 33

An understanding of the brain



- ~86B neurons
- ~100T synapses
- also ~85B glia
- specialised circuits
- different neuronal types
- synaptic connections
- complex sub-cellular processes

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

²von 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)

2 / 33

Notes

Models complement experimental neuroscience

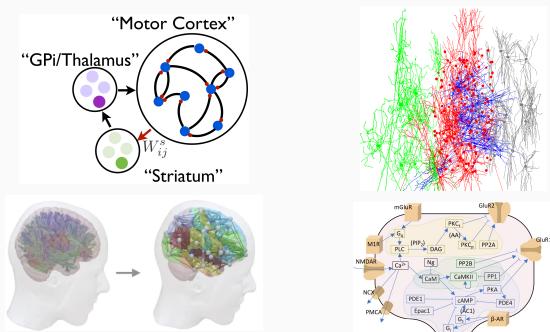
Models are fully observable, controllable.

- 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

3 / 33

Notes

Models: different scales



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

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

²Yao, H. K. et al. Reduced inhibition in depression impairs stimulus processing in human cortical microcircuits. *Cell Reports* 38, issn: 2211-2124 (Jan. 2022)

³Mäki-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)

Notes

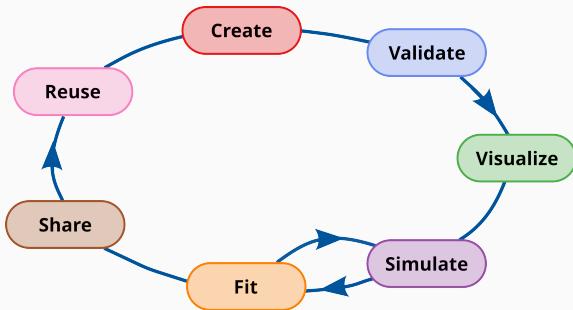
4 / 33

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

Notes

5 / 33

The model life cycle



Notes

6 / 33

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

Notes

7 / 33

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

Notes

8 / 33

Standards enable FAIR neuroscience



inCF International Neuroinformatics Coordinating Facility



COMBINE



SML



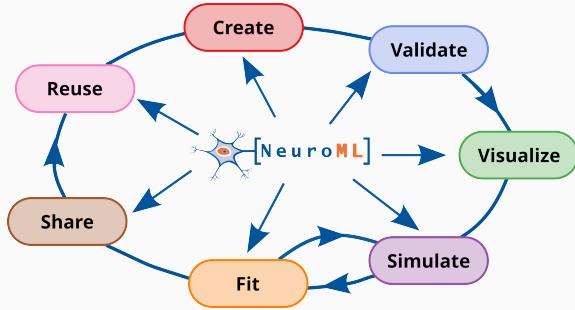
¹ Abrams, M. B. et al. A Standards Organization for Open and FAIR Neuroscience: the International Neuroinformatics Coordinating Facility. *Neuroinformatics* 20, 25–36. ISSN: 1559-0289. <https://doi.org/10.1007/s12081-020-09509-1> (2022). <https://incf.org/>

² Computational Modeling in Biology Network (COMBINE) <https://co.mbine.org/>

9 / 33

Notes

NeuroML ecosystem supports all stages of the model cycle



10 / 33

Notes

NeuroML ecosystem

- standard/specification
- software ecosystem

11 / 33

Notes

NeuroML standard

Model specification (**schema: XSD**)

- elements
- attributes
- hierarchical relationships

Dynamics (**LEMS component type definitions**)

- dynamical behaviour

Notes

12 / 33

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/>

Notes

NeuroML standard: schema: XSD

```
<xsd:simpleType name="Mml2Quantity_voltage"> <!-- For params with dimension voltage -->
<xsd:restriction base="xsd:string">
  <xsd:pattern value="^-?[0-9]+([.][0-9]+)?([eE]-?[0-9]+)?([a-zA-Z]*V|N)" />
</xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="izhikevich2007Cell">
  <xsd:documentation>Cell based on ...</xsd:documentation>
  <xsd:annotation>
    <xsd:documentation>
      <xsd:base>BaseCellMembPotCap</xsd:base>
      <xsd:attribute name="v0" type="Mml2Quantity_voltage" use="required"/>
      <xsd:attribute name="k" type="Mml2Quantity_conductancePerVoltage" use="required"/>
      <xsd:attribute name="vr" type="Mml2Quantity_voltage" use="required"/>
      <xsd:attribute name="tau_m" type="Mml2Quantity_time" use="required"/>
      <xsd:attribute name="vpeak" type="Mml2Quantity_voltage" use="required"/>
      <xsd:attribute name="a" type="Mml2Quantity_perTime" use="required"/>
      <xsd:attribute name="b" type="Mml2Quantity_conductance" use="required"/>
      <xsd:attribute name="c" type="Mml2Quantity_voltage" use="required"/>
      <xsd:attribute name="d" type="Mml2Quantity_current" use="required"/>
    </xsd:base>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="BaseCellMembPotCap">
      <xsd:attribute name="v0" type="Mml2Quantity_voltage" use="required"/>
      <xsd:attribute name="k" type="Mml2Quantity_conductancePerVoltage" use="required"/>
      <xsd:attribute name="vr" type="Mml2Quantity_voltage" use="required"/>
      <xsd:attribute name="tau_m" type="Mml2Quantity_time" use="required"/>
      <xsd:attribute name="vpeak" type="Mml2Quantity_voltage" use="required"/>
      <xsd:attribute name="a" type="Mml2Quantity_perTime" use="required"/>
      <xsd:attribute name="b" type="Mml2Quantity_conductance" use="required"/>
      <xsd:attribute name="c" type="Mml2Quantity_voltage" use="required"/>
      <xsd:attribute name="d" type="Mml2Quantity_current" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

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

Notes

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

¹Cannon, 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)

Notes

NeuroML standard: dynamics (LEMS)

```
<ComponentType name="izhikevich2007Cell" extends="baseCellMembPotCap"
  description="Cell based ...">
  <xsd:Parameter name="v0" dimension="voltage" description="Initial membrane potential"/>
  <!--Defined in baseCellMembPotCap:-->
  <xsd:Parameter name="C" dimension="capacitance"/>
  <xsd:Parameter name="k" dimension="conductance_per_voltage"/>
  <xsd:Parameter name="vr" dimension="voltage" description="Resting membrane potential"/>
  <xsd:Parameter name="vt" dimension="voltage" description="Spike threshold"/>
  <xsd:Parameter name="vpeak" dimension="voltage" description="Peak action potential value"/>
  <xsd:Parameter name="a" dimension="per_time" description="Time scale of recovery variable u"/>
  <xsd:Parameter name="b" dimension="conductance" description="Sensitivity of recovery variable u to subthreshold
  fluctuations of membrane potential v"/>
  <xsd:Parameter name="c" dimension="voltage" description="After-spike reset value of v"/>
  <xsd:Parameter name="d" dimension="current" description="After-spike increase to u"/>
  <xsd:Attachments name="synapses" type="basePointCurrent"/>
  <xsd:Exposure name="u" dimension="current" description="Membrane recovery variable"/>
  <xsd:Dynamics><!-- snipped --></xsd:Dynamics>
</ComponentType>
```

Notes

NeuroML standard: XSD and LEMS

XSD:

```
<x:attribute name="v0" type="Nml2Quantity_voltage" use="required"/>
<x:attribute name="k" type="Nml2Quantity_conductancePerVoltage" use="required"/>
<x:attribute name="vr" type="Nml2Quantity_voltage" use="required"/>
<x:attribute name="vt" type="Nml2Quantity_voltage" use="required"/>
<x:attribute name="a" type="Nml2Quantity_perTime" use="required"/>
<x:attribute name="b" type="Nml2Quantity_conductance" use="required"/>
<x:attribute name="c" type="Nml2Quantity_voltage" use="required"/>
<x: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="Resetting membrane potential"/>
<Parameter name="vt" dimension="voltage" description="Spike threshold"/>
<Parameter name="a" dimension="voltage" description="Peak action potential value"/>
<Parameter name="b" dimension="per_time" description="Time scale of recovery variable u"/>
<Parameter name="c" dimension="per_time" description="Sensitivity of recovery variable u to subthreshold fluctuations of membrane potential v"/>
<Parameter name="d" dimension="current" description="After-spike increase to u"/>
```

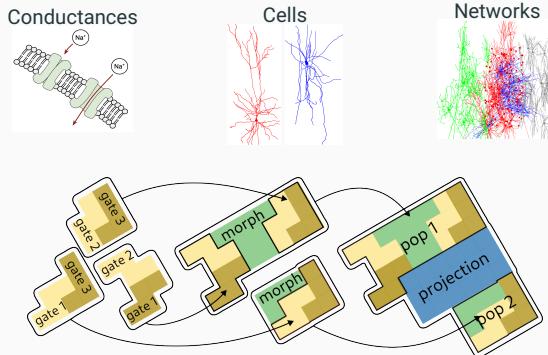
17 / 33

NeuroML standard: dynamics (LEMS)

```
<ComponentType name="izhikevich2007Cell" extends="baseCellMemPotCap"
  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) + 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>
<OnDefinition test="v . gt . vpeak">
  <StateAssignment variable="u" value="c"/>
  <StateAssignment variable="u" value="u + d"/>
  <EventOut port="spike"/>
</OnDefinition>
</Dynamics>
</ComponentType>
```

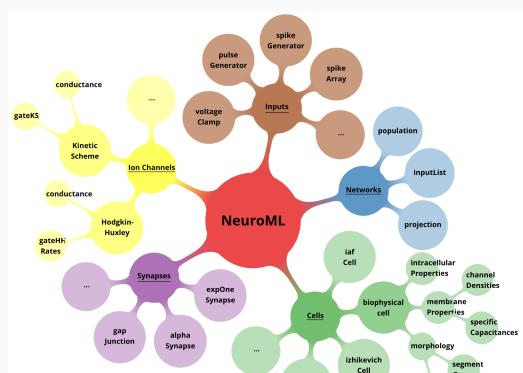
18 / 33

NeuroML is declarative, modular, structured, hierarchical



19 / 33

NeuroML provides users with a set of curated model elements



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

Notes

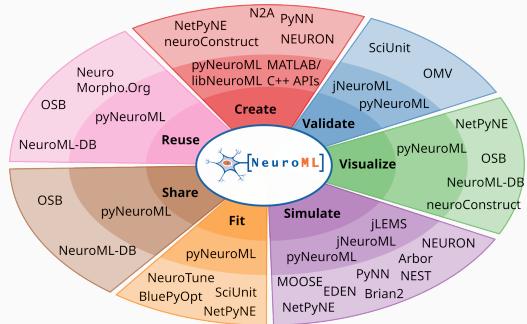
Notes

Notes

Notes

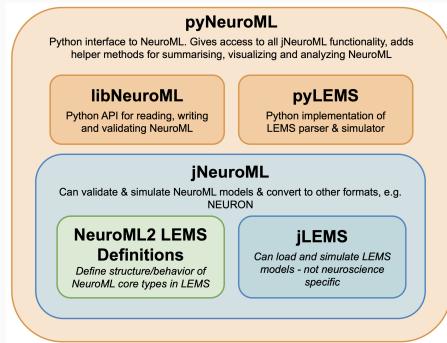
20 / 33

NeuroML software ecosystem



21 / 33

NeuroML software ecosystem: core tools



pip install pyneuroml

22 / 33

NeuroML software ecosystem: pyNeuroml

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

# inspection
element.info()
summary(doc)
ml2_to_png(doc)
ml2_to_svg(doc)
generate_umgraph(doc)

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

plot_chanel_densities(cell)
plot_time_series(file)

# simulation
run_lems_with_neuronl("sim.xml")
run_lems_with_neuronl_neuron("sim.xml")
run_on_neu("jneuroml_neuron", "sim.xml")

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

23 / 33

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)
```

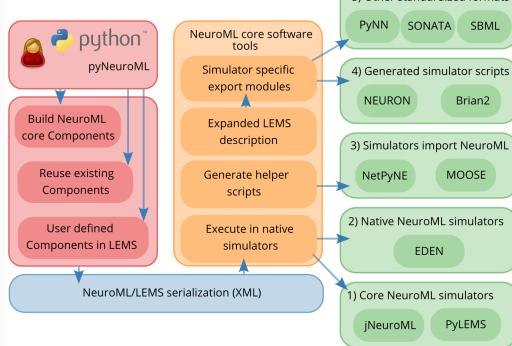
Resultant NeuroML XML serialization:

```
<neuroml id="new_doc">
  <ifTauCell 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>
```

Notes

24 / 33

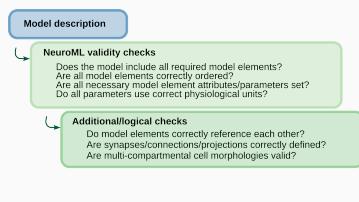
NeuroML: creating/simulating models



25 / 33

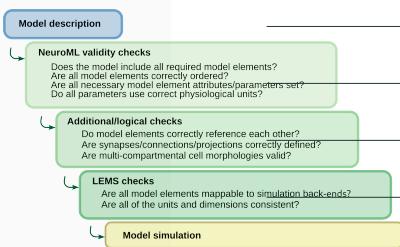
Notes

NeuroML: validating models

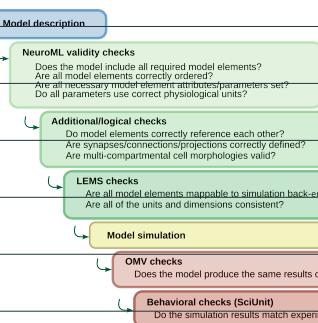


¹<https://github.com/OpenSourceBrain/osb-model-validation>

²<https://sciumit.io>



Notes



Notes

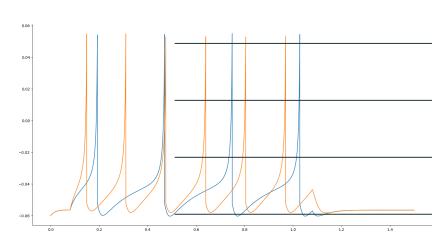
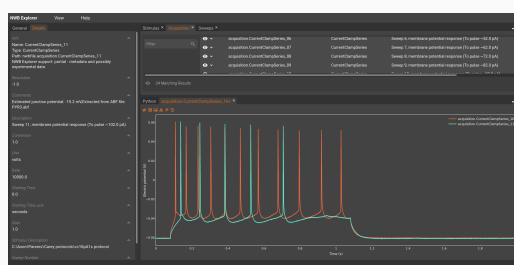
NeuroML: visualising/analysing models



26 / 33

Notes

NeuroML: fitting models



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

²Izhikevich cells fitted using NeuroML fitting pipeline: NeuroTune, using Inspyred

Notes

NeuroML: sharing and re-using models

- ¹ Standardized models on GitHub: Open Source Brain: <https://github.com/OpenSourceBrain>
- ¹ Standardized models on Open Source Brain v1: <https://v1.opensourcebrain.org>
- ¹ Standardized models on Open Source Brain v2: <https://v2.opensourcebrain.org>
- ¹ Standardized models on NeuroML-DB: <https://neuroml-db.org>

The screenshot shows the Open Source Brain homepage on the left, featuring a large green 'Model the brain' button, a search bar, and sections for 'Learn about the Hodgkin-Huxley model', 'Simulate electrophysiologically detailed cell models', and 'Explore more OSB projects'. Below these are links for creating an account and logging in. On the right, a search results page for 'Modelling the brain, together' is displayed, showing results like 'WormNeuroAtlas', 'M1-channelspikes-0198', and 'Thalamic reticular network with spiking'. Each result includes a preview image, a 'Details' link, and a 'View' button.

NeuroML: community: events

<p>Conference Date 8-11 April, 2024</p> <p>Conference Location UGS, London, UK</p> <p>Important Dates</p> <ul style="list-style-type: none"> Deadline for abstracts submission deadline 1 March, 2024 Notification of acceptance for breakout and tutorials 18 March, 2024 Lighting talk and poster submission deadline 11 March, 2024 Notification of acceptance for lightning talks and posters 19 March, 2024 <p>Travel support applications NOW CLOSED</p> <p>Notification of travel support awards 21 February, 2024</p>	 <p>the COMBINE computational modeling in biology network</p> <p>The "Computational Modeling in Biology" Network (COMBINE) is an initiative to coordinate the development of the various community standards and tools in systems biology, proteomics, genomics, and related fields, and to organize a community meeting, with a focus on development of standards and tools for the analysis and integration of biological data. The aim of the meeting is to bring together a diverse group of scientists and practitioners from across the field, including many graduate students and early-career researchers. The meeting will provide a forum for the presentation of new research results and/or posters during HARMONY 2024, and the time is devoted to develop hands-on hacking and interaction between people focused on the practical development of software and standards.</p> <p>HARMONY 2024 will be held at University College London</p> <p>Local organizers are Sarah Keeling (skeeling@ucl.ac.uk) and Padraig Gleeson (pgleeson@ucl.ac.uk).</p> <p>Workshop Location</p> 
---	--

The screenshot shows the CNS*2024 homepage. At the top, there's a navigation bar with links for Home, Calendar, Blog, About CNS*, Meetings, Membership, Publications, Working groups, and Get help. Below the navigation is a banner for the "33rd Annual Computational Neuroscience Meeting" in Natal, Brazil, from July 20-24, 2024. The banner features a photo of a tropical beach and the text "CNS 2024". To the right of the banner is a sidebar titled "Quick Links" containing links to CNS*, CNS*2024, Camp Neuromodulation Ltd, IFN* Neuroblast, IAP* Training series, JCN* Journal, CNS* on Twitter, CNS* on LinkedIn, and CNS* on YouTube. The main content area has sections for "CNS*2024 Important Dates", "Registration", "Abstracts", "Travel", "Add to registration", "Submit an abstract", and "Registration fees". There are also sections for "Register for CNS*2024", "Checklist for registration fees", "Visa and travel information for Brazil", and "July 20-24, 2024". On the right side, there are sections for "Upcoming Events", "Meetings", and a "View Full Calendar" button. At the bottom, there are sections for "Announcements" and "News", with links to "CNS*2024 Abstract submission", "CNS*2024 Registration", and "CNS*2024 Registration is now open! Discounted pricing for early registrants".

30 / 33

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>

31 / 33

NeuroML: closing the neuroscience research loop with OSB

- Open Source Brain Video

Notes

Notes

Notes

Notes

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

<https://docs.neuroml.org>
<https://opensourcebrain.org>

33 / 33

Notes

Notes

Notes

Notes
