

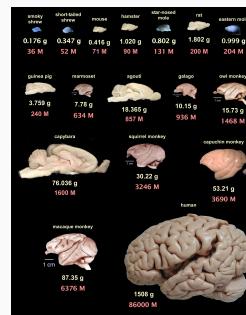
# The NeuroML ecosystem for standardised multi-scale modelling in neuroscience

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

2024-02-26

1/34

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



<sup>4</sup>Herculano-Houzel, S. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience*, 3, 31 (2009).

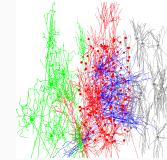
<sup>1</sup>von Bartheld, C. S. et al. The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting. *Journal of Comparative Neurology* 524, 3865–3895. ISSN: 1096-9861 (June 2016)

2 / 34

## Models complement experimental neuroscience

- 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 / 34



<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)

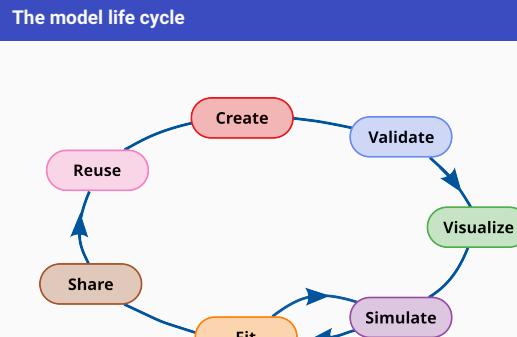
<sup>1</sup> Yao, H. K. et al. Reduced inhibition in depression impairs stimulus processing in human cortical microcircuits. *bioRxiv*, 2017.

<https://doi.org/10.1016/j.celrep.2021.1102>

4/34

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

5/34



6 / 34

**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:
      - cannot be easily inspected/analysed
    - ad-hoc utilities:
      - cannot be used with all tools

7/34

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

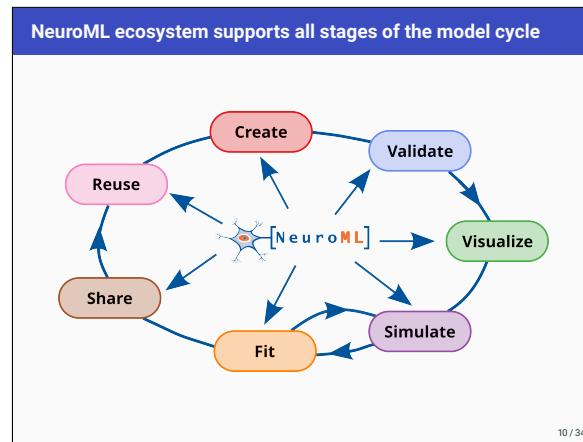
8 / 34

**Standards enable FAIR neuroscience**

<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. doi: 10.1007/s12192-020-00308-0 (2022). <https://doi.org/10.1007/s12192-020-00308-0>

<sup>2</sup> Computational Modeling in Biology Network (COMBINE) <http://www.combine.org/>

9 / 34



**NeuroML ecosystem**

- standard/specification
- software ecosystem

11 / 34

**NeuroML standard**

Model specification (schema: XSD)

- elements
- attributes
- hierarchical relationships

Dynamics (LEMS component type definitions)

- dynamical behaviour

12 / 34

**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**

One can validate a model description against the schema before simulation

<sup>1</sup> <https://www.w3.org/TR/xmlschema-1/>

13 / 34

**NeuroML standard: schema: XSD**

```
<xsd:simpleType name="MnL2Quantity_voltage"> <!-- For params with dimension voltage -->
<xsd:restriction base="xsd:string">
<xsd:pattern value="^((D\.)?([0-9]+)?([A-E]\-[0-9]+)?([a-z]+(V|mA))$"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="Izhikevich2007Cell1">
<xsd:annotation>
<xsd:documentation>Cell based on ...</xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
<xsd:extension base="BaseCellMembranePotCap">
<xsd:attribute name="v0" type="MnL2Quantity_voltage" use="required"/>
<xsd:attribute name="g" type="MnL2Quantity_conductance_perVoltage" use="required"/>
<xsd:attribute name="v" type="MnL2Quantity_voltage" use="required"/>
<xsd:attribute name="tau_m" type="MnL2Quantity_time" use="required"/>
<xsd:attribute name="a" type="MnL2Quantity_perturb" use="required"/>
<xsd:attribute name="b" type="MnL2Quantity_compliance" use="required"/>
<xsd:attribute name="c" type="MnL2Quantity_voltage" use="required"/>
<xsd:attribute name="d" type="MnL2Quantity_current" use="required"/>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

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

14 / 34

**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> O'Connor, D. 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 (2016)

15 / 34

**NeuroML standard: dynamics (LEMS)**

```
<ComponentType name="Izhikevich2007Cell1" extends="baseCellMembranePotCap"
  description="Cell based ...">
  <Parameter name="v0" dimension="voltage" description="Initial membrane potential"/>
  <!-- If used in baseCellMembranePotCap -->
  <Parameter name="C" dimension="capacitance"/>
  <Parameter name="g" dimension="conductance_per_voltage"/>
  <Parameter name="v" dimension="voltage" description="Reaching membrane potential"/>
  <Parameter name="tau_threshold" dimension="voltage" description="Spike threshold"/>
  <Parameter name="peak" 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"/>
  <Dynamic><!-- snipped --></Dynamic>
</ComponentType>
```

16 / 34

NeuroML standard: XSD and LEMS

XSD:

```
<xss:attribute name="v" type="Hal2Quantity_voltage" use="required"/>
<xss:attribute name="x" type="Hal2Quantity_conductancePerVoltage" use="required"/>
<xss:attribute name="y" type="Hal2Quantity_voltage" use="required"/>
<xss:attribute name="z" type="Hal2Quantity_voltage" use="required"/>
<xss:attribute name="a" type="Hal2Quantity_perUnit" use="required"/>
<xss:attribute name="b" type="Hal2Quantity_perUnit" use="required"/>
<xss:attribute name="c" type="Hal2Quantity_conductance" use="required"/>
<xss:attribute name="d" type="Hal2Quantity_current" use="required"/>
```

| EEMS

```

<Parameter name="v" dimension="voltage" description="Initial membrane potential"/>
<Parameter name="k" dimension="conductance_per_voltage" description="Resting membrane potential"/>
<Parameter name="vth" dimension="voltage" description="Spike threshold"/>
<Parameter name="tau" dimension="time" description="Decay time constant of membrane voltage"/>
<Parameter name="par_time" dimension="time" description="Time scale of recovery variable u"/>
<Parameter name="B" dimension="conductance" description="Sensitivity of recovery variable u to subthreshold spikes"/>
<Parameter name="d" dimension="current" description="After-spike reset value of v"/>
<Parameter name="u" dimension="current" description="After-spike reset value of u"/>

```

NeuroML standard: dynamics (LEMS)

```

<ComponentType name="iLimbwiw120207Cell1" extends="baseCellMeshPotCap"
  description="Cell based ...">
  <!-->
  <Attachments name="synapses" type="basePointCurrent"/>

  <Expose name="u" dimension="current" description="Membrane recovery variable"/>
  <Expose name="v" dimension="voltage" exposure="r"/>
  <StateVariable name="u" dimension="current" exposure="r"/>
  <StateVariable name="v" dimension="voltage" exposure="r"/>

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

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

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

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

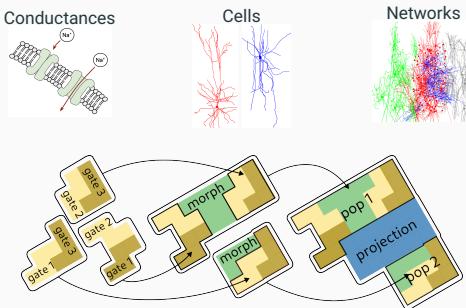
  <OnUpdate time="t" gt="vpeak">
    <StateAssignment variable="v" value="0" />
    <StateAssignment variable="u" value="u + d" />
    <Branch port="spike" />
  </OnUpdate>

</Dynamic>
</ComponentType>

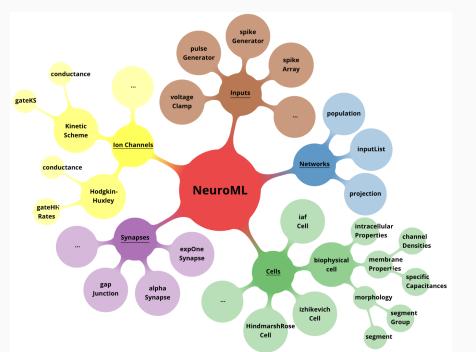
```

18 / 34

**NeuroML is declarative, modular, structured, hierarchical**

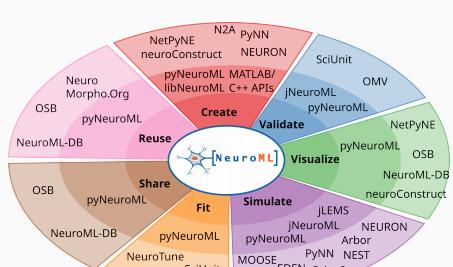


NeuroML provides users with a set of curated model elements

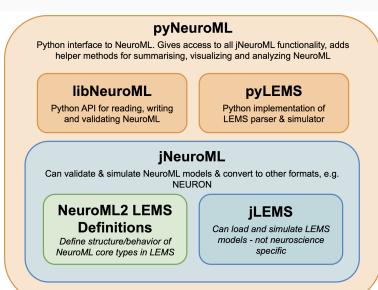


20 / 34

NeuroML software ecosystem



NeuroML software ecosystem: core tools



```
pip install pyneuroml
```

NeuroML: creating/simulating models

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

```

from neuroml import * # NeuroML API libNeuroML

neurod = NeuroMLDocument(id="new_doc")
neurocell = IafTauCell(id="cell_0", leak_reversal="-60mV", thresh="0mV", tau="5ms",
neurod.add(neurocell)

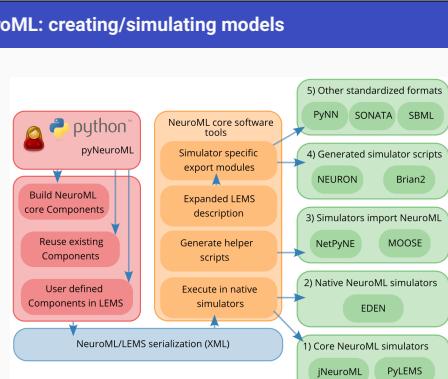
network = neurod.add(Network, id="new_net", validate=False)
population = network.addPopulation(id="new_pop", size=10, component=neurocell.id)

# Helper method to ensure all parameters
# present and appropriate

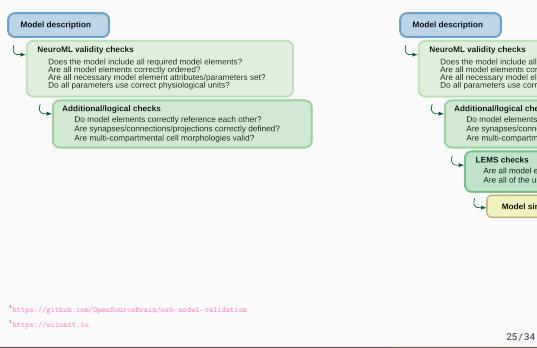
```

## 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: validating models



25 / 34

## NeuroML: visualising/analysing models



\*3D interactive visualisation of Migliore et al. [1] using pyneuroplotmorph  
\*\*3D interactive visualisation of Sadeh et al. [11] on Open Source Brain: <https://vt.opensourcebrain.org>

## NeuroML: fitting models

- Figure from docs
- Mention inspyred

27 / 34

## NeuroML: sharing and re-using models

- GitHub, OSBv1, OSBv2, NeuroML-DB

28 / 34

## NeuroML: Documentation

- Jupyterbook

29 / 34

## NeuroML: community

- Editorial board, scientific committee, communication channels

30 / 34

## NeuroML: closing the neuroscience research loop with OSB

- Open Source Brain Video

31 / 34

## NeuroML: projects

- GSoC, Outreachy, good computer science students

32 / 34

## 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](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>

## NeuroML: the APIs

- Python API