

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

---

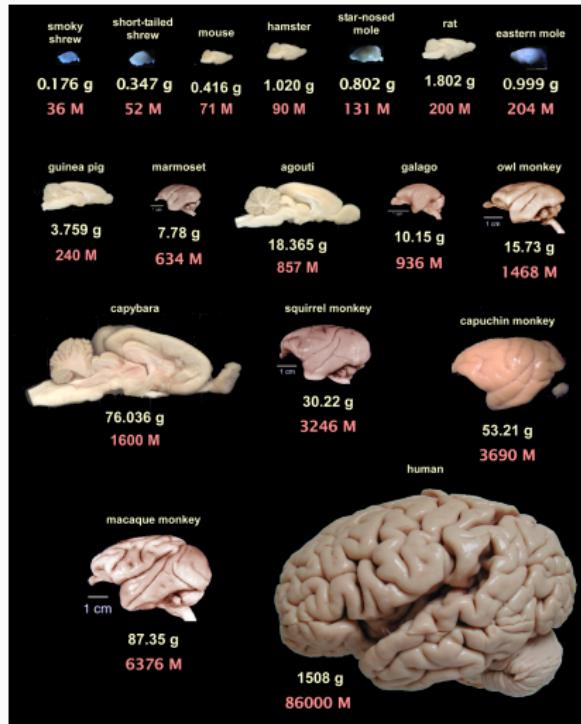
Ankur Sinha

Silver Lab

Department of Neuroscience, Physiology, & Pharmacology  
University College London

2024-02-26

# An understanding of the brain

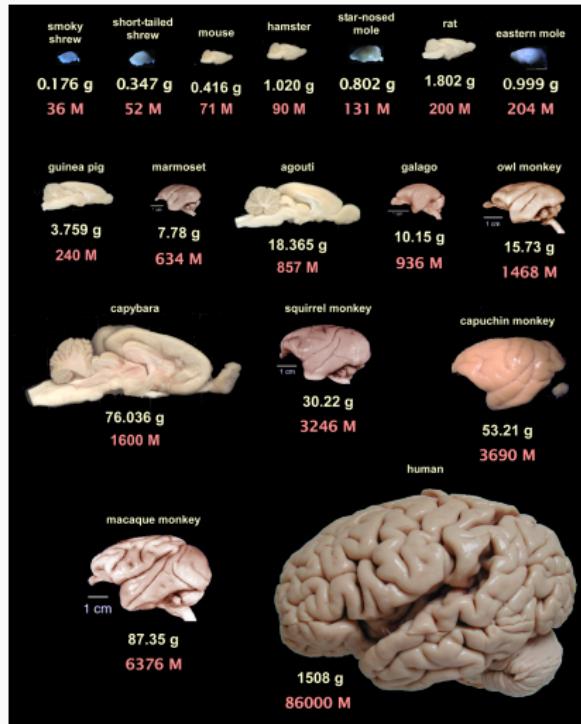


- ~86B neurons
- ~100T synapses
- also ~85B glia

<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)

# An understanding of the brain



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

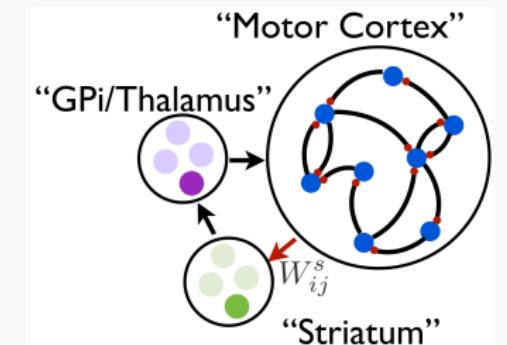
<sup>1</sup> Herculano-Houzel, S. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience* 3, 31 (2009)

<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)

## Experiments provide a window into the brain

Multiple scales of experiments/data sources go here

# Models test & unify experimental results; generate hypotheses



<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)

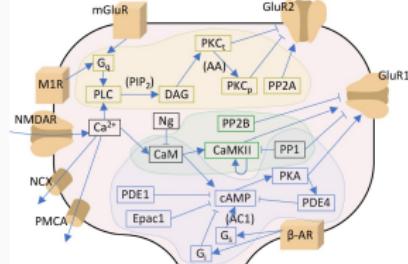
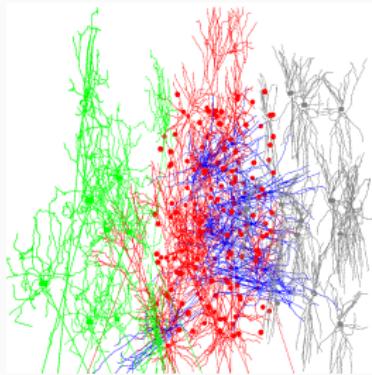
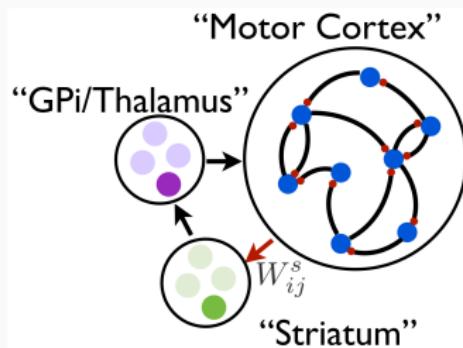
<sup>1</sup> Yao, H. K. et al. Reduced inhibition in depression impairs stimulus processing in human cortical microcircuits. *Cell Reports* 38. ISSN: 2211-1247.

<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)

# Models test & unify experimental results; generate hypotheses



<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.

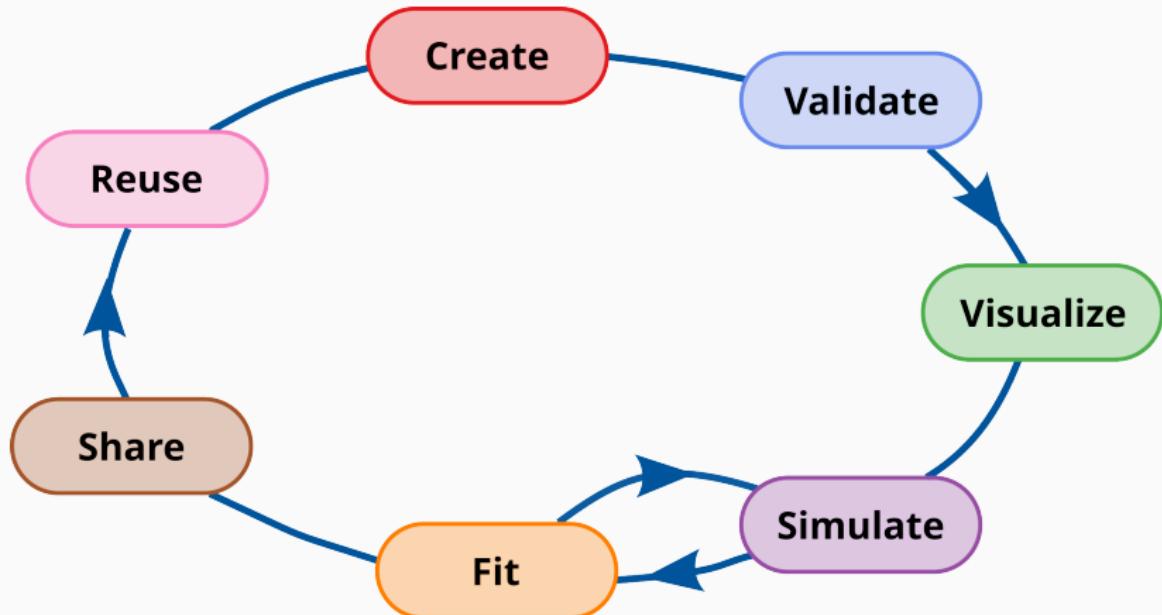
<https://doi.org/10.1016/j.celrep.2021.110232> (Jan. 2022)

<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...

# 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 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 validated
  - custom machine readable internal representations:
    - 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 validated
  - custom machine readable internal representations:
    - 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

<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/>

# Standards enable FAIR neuroscience



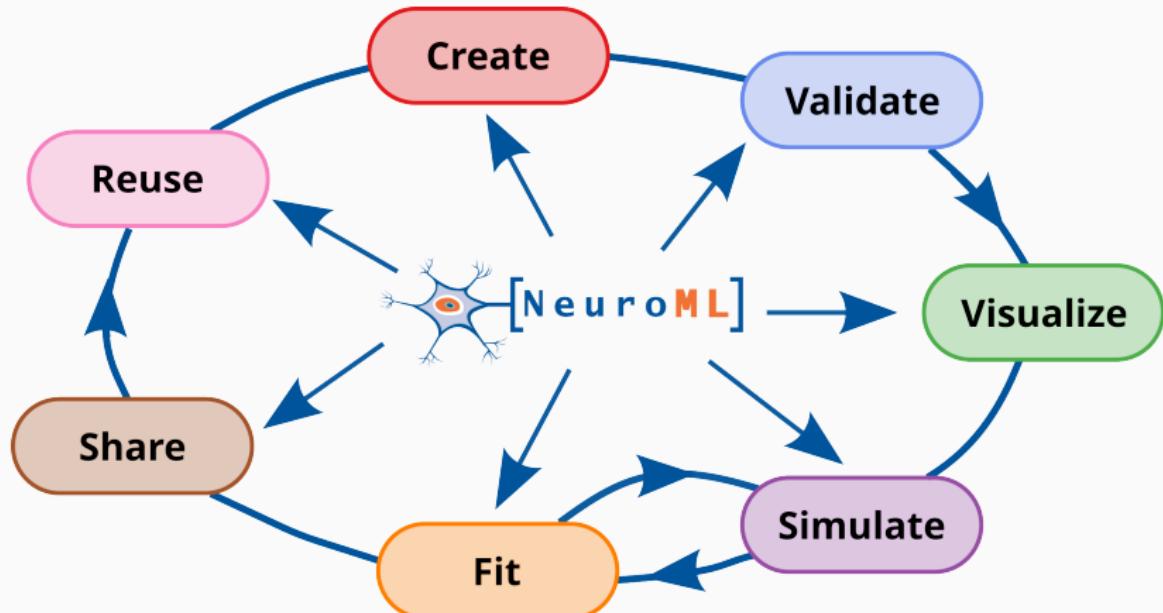
COMBINE



<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

## Model specification (schema: XSD)

- elements
- attributes
- hierarchical relationships

## Dynamics (LEMS)

- 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**

<sup>1</sup><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**

One can validate a model description 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

### Low Entropy Model Specification language

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

<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)"/>

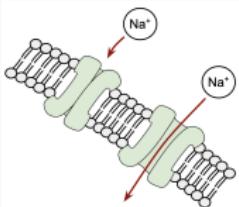
    <OnStart>
      <StateAssignment variable="v" value="v0"/>
      <StateAssignment variable="u" value="0"/>
    </OnStart>

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

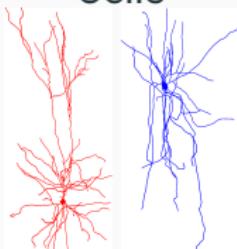
  </Dynamics>
</ComponentType>
```

# NeuroML is modular, structured, and hierarchical

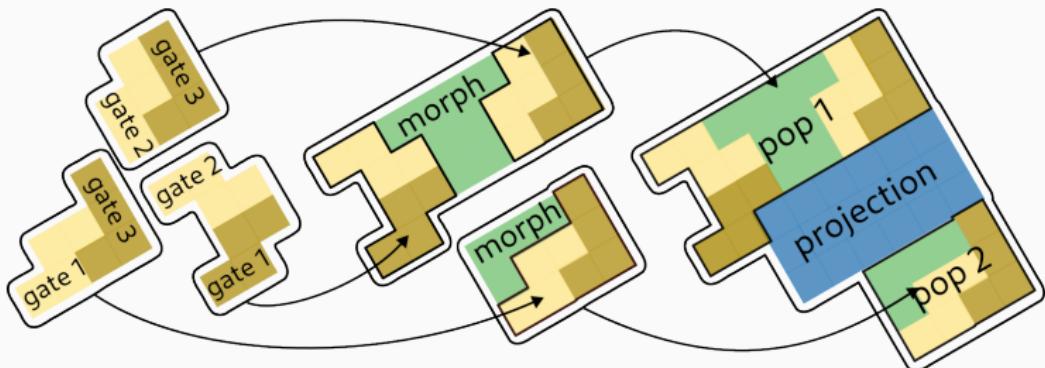
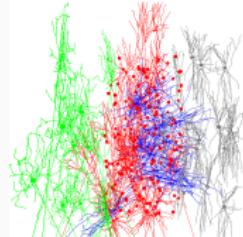
Conductances



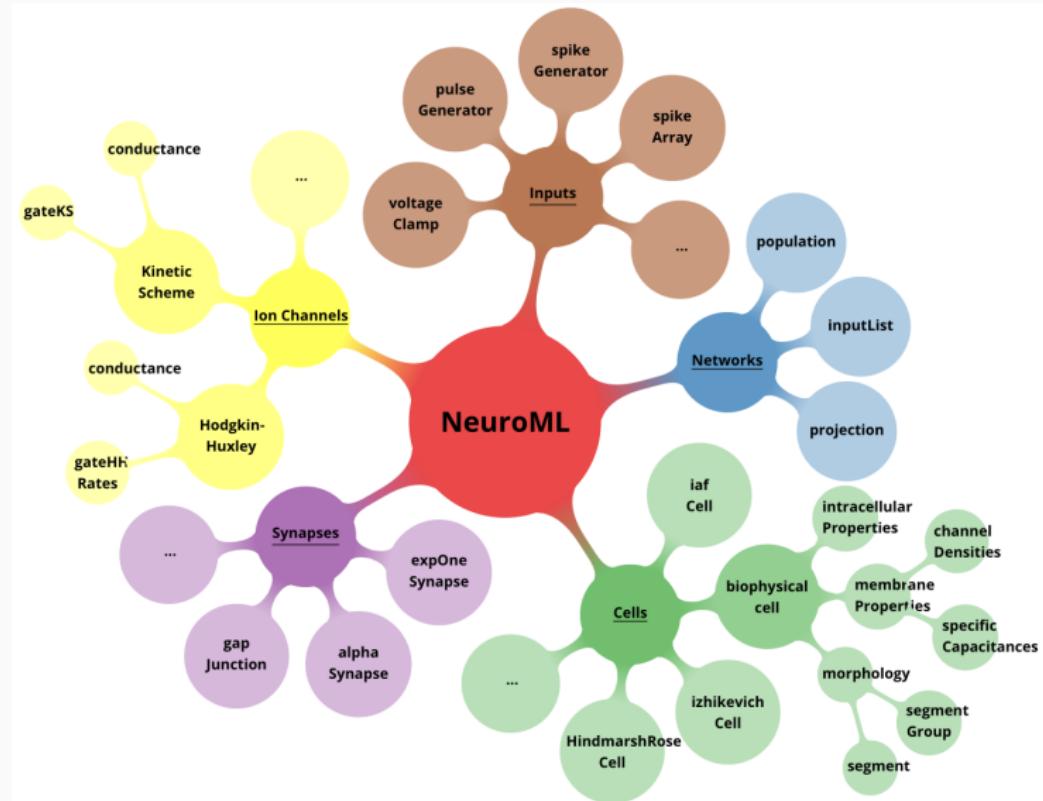
Cells



Networks

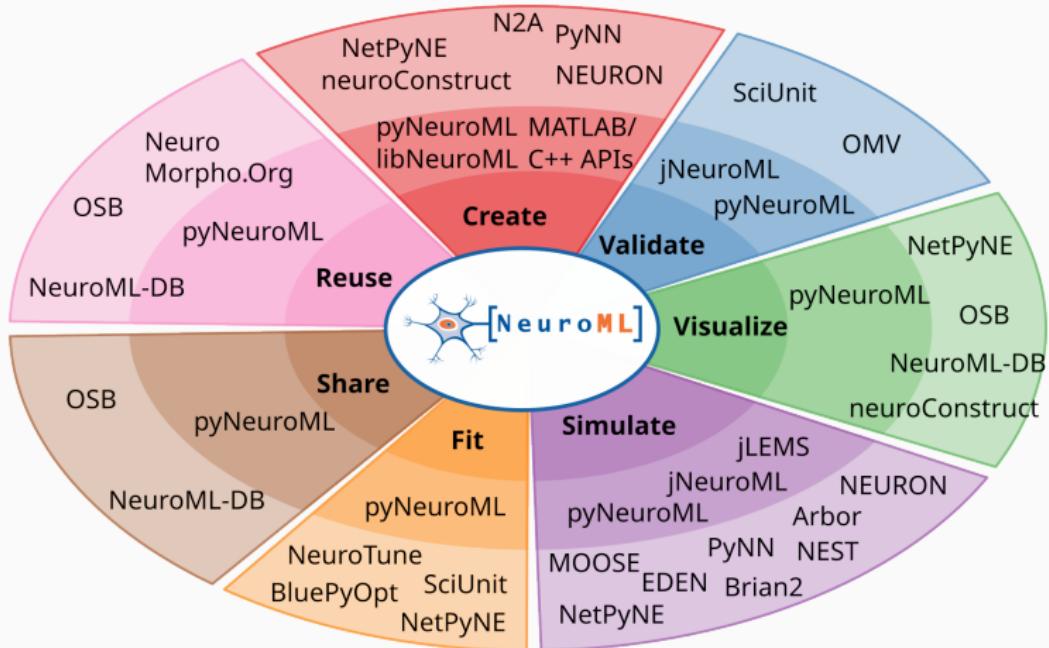


# NeuroML provides users with a set of curated model elements



<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

# NeuroML: create 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)

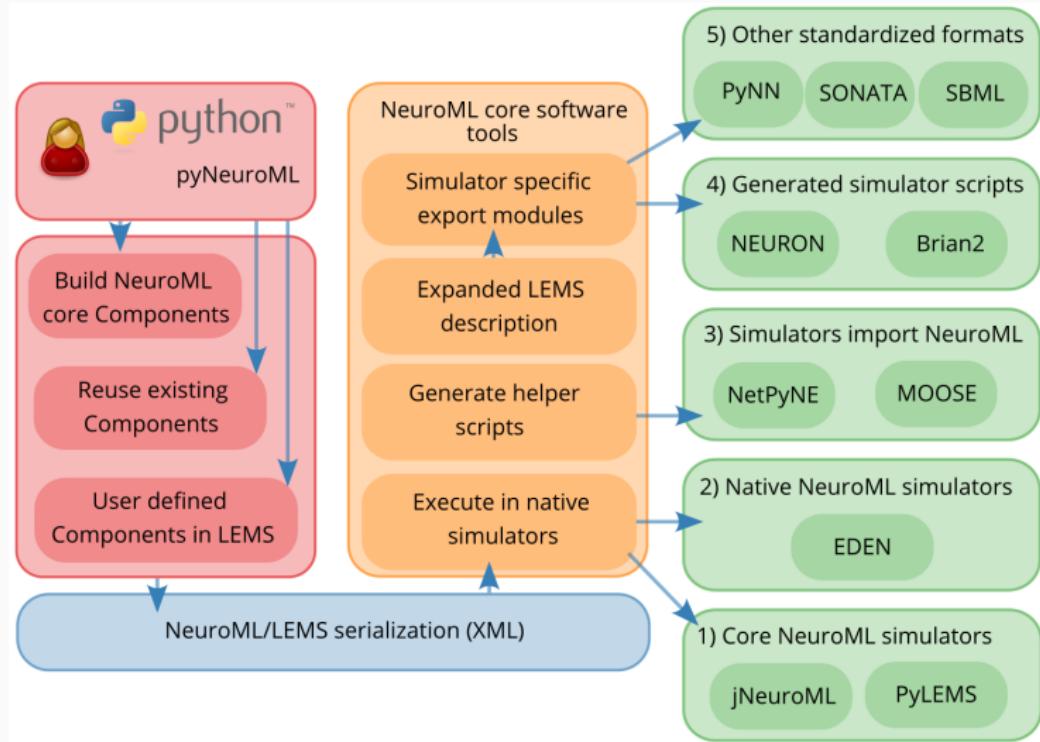
network = newdoc.add(Network, id="new_net", validate=False)
population = network.add(Population, id="new_pop", size=10, component=newcell.id)

# 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: create models



# NeuroML: validate 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: validate 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?

### LEMS checks

- Are all model elements mappable to simulation back-ends?
- Are all of the units and dimensions consistent?

### Model simulation

# NeuroML: validate 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?

### LEMS checks

- Are all model elements mappable to simulation back-ends?
- Are all of the units and dimensions consistent?

## Model simulation

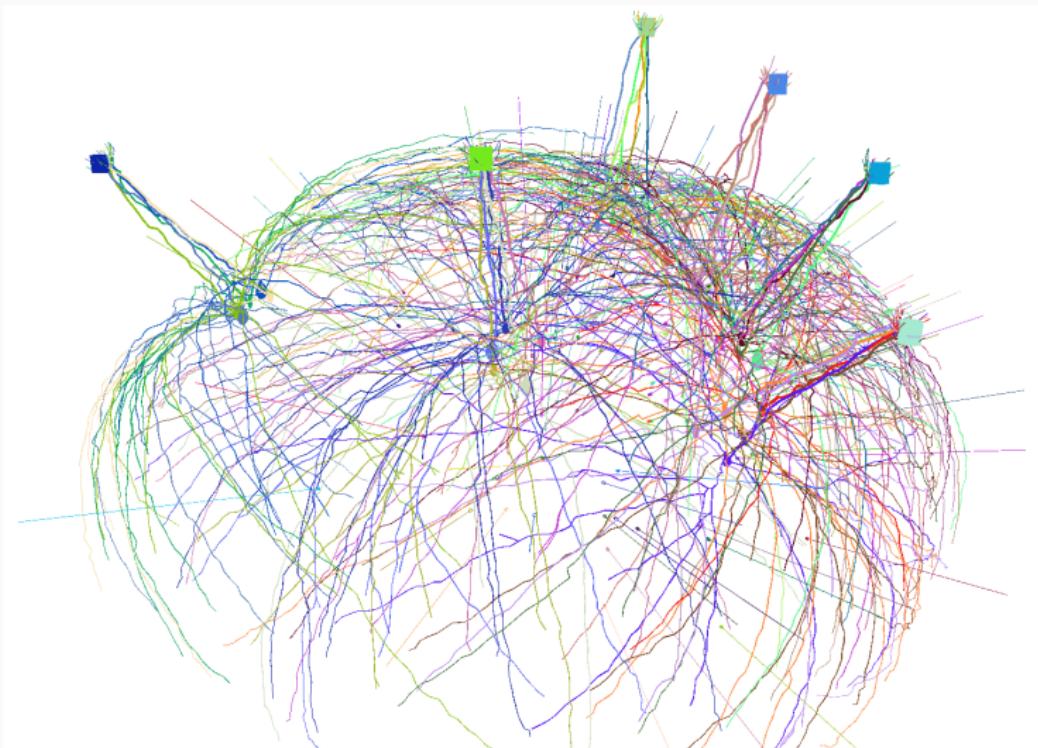
### OMV checks

- Does the model produce the same results on all simulators?

### Behavioral checks (SciUnit)

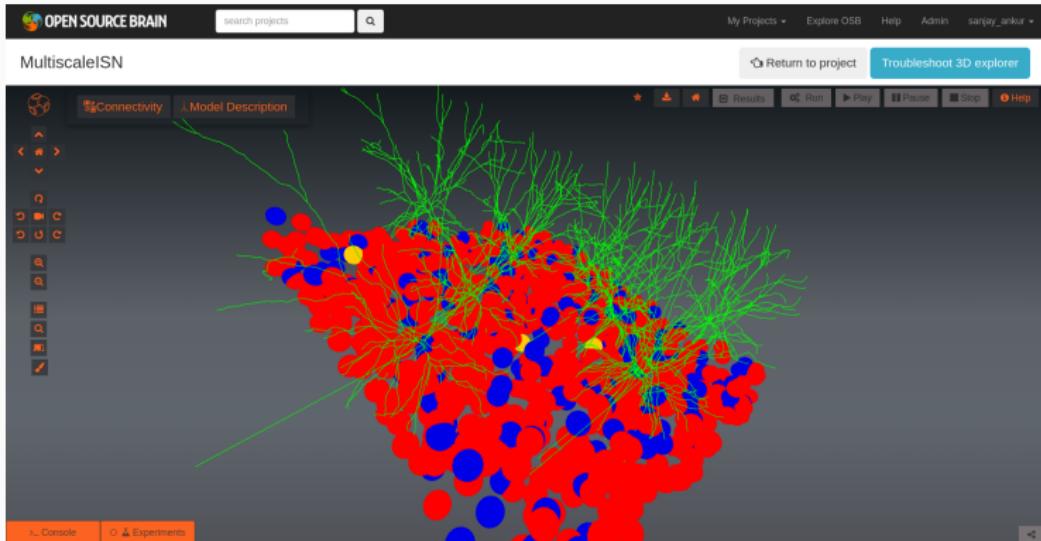
- Do the simulation results match experimental data?

# NeuroML: visualise/analyse models



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

# NeuroML: visualise/analyse models



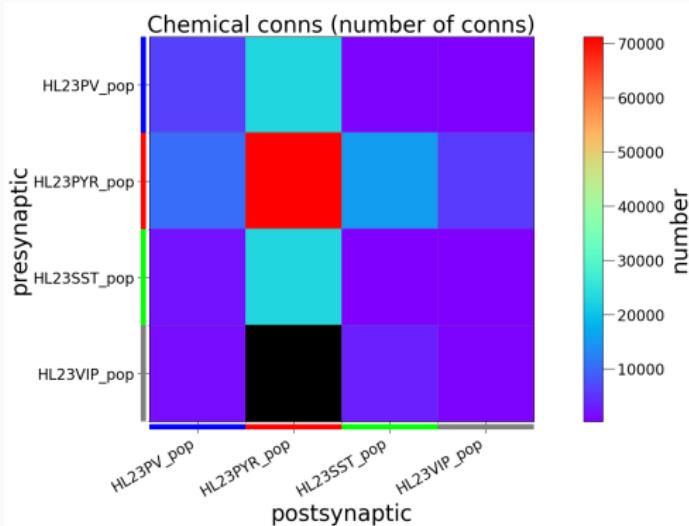
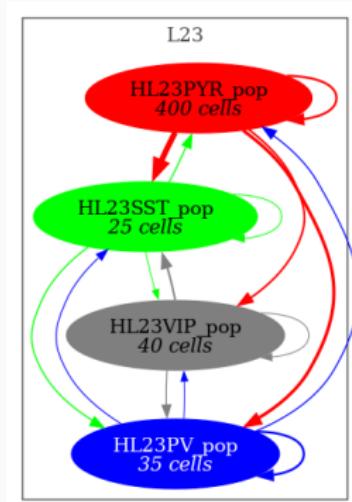
<sup>1</sup> 3D interactive visualisation of Sadeh et al. [11] on Open Source Brain: <https://vi.opensourcebrain.org>

# NeuroML: visualise/analyse models



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

# NeuroML: visualise/analyse models



## NeuroML: simulate models

- Example simulation: neuron/netpyne

## NeuroML: fit models

- Figure from docs
- Mention inspyred

## NeuroML: share and re-use models

- GitHub, OSBv1, OSBv2, NeuroML-DB

# NeuroML: Documentation

- Jupyterbook

## NeuroML: projects

- GSoC, Outreachy, good computer science students

## 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)

<https://docs.neuroml.org>

# NeuroML: the APIs

- Python API