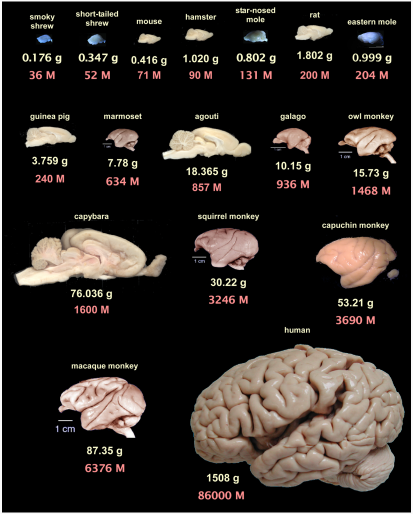
The NeuroML ecosystem for standardised multi-scale modelling in neuroscience

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

2024-02-26

An understanding of the brain

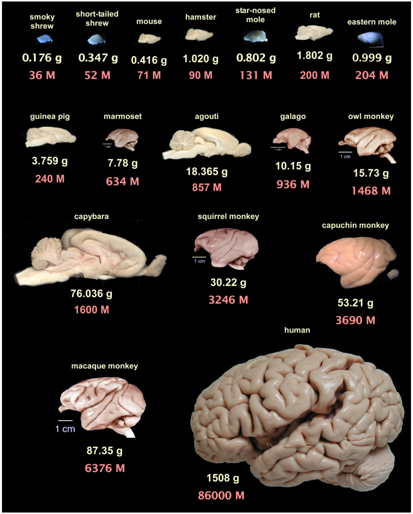


- ~86B neurons
- ~100T synapses
- also ~85B glia

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

[1] 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

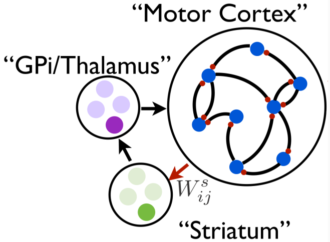
[1] Herculano-Houzel, S. **The human brain in numbers: a linearly scaled-up primate brain.** *Frontiers in human neuroscience* **3,** 31 (2009)

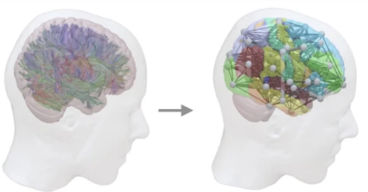
[1] von Bartheld, C. S. *et al.* **The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting.** *Journal of Comparative Neurology* **524,** 3865–3895. ISSN: 1096-9861 (June 2016)

Experiments provide a window into the brain

Multiple scales of experiments/data sources go here

Models test & unify experimental results; generate hypotheses





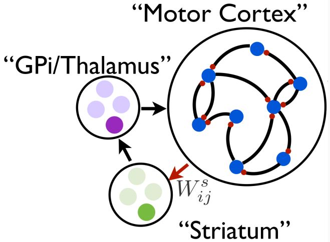
[1] Murray, J. M. **Local online learning in recurrent networks with random feedback.** *eLife* **8** (eds Latham, P. *et al.*) e43299. ISSN: 2050-084X (2019)

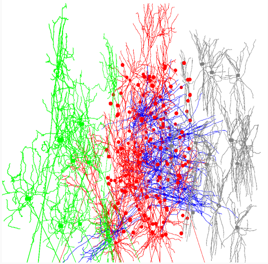
[1] Schirner, M. *et al.* **Learning how network structure shapes decision-making for bio-inspired computing.** *Nature Communications* **14.** ISSN: 2041-1723 (May 2023)

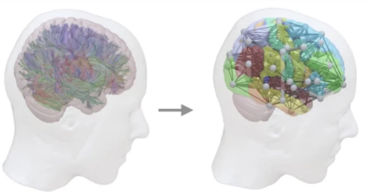
[1] Yao, H. K. *et al.* **Reduced inhibition in depression impairs stimulus processing in human cortical microcircuits.** *Cell Reports* **38.** ISSN: 2211-1247. https://doi.org/10.1016/j.celrep.2021.110232 (Jan. 2022)

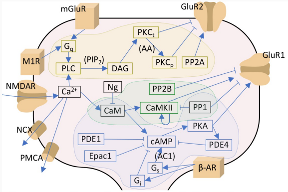
[1] Mäki-Marttunen, T. *et al.* **A unified computational model for cortical post-synaptic plasticity.** *eLife* **9** (eds Shouval, H. Z. *et al.*) e55714. ISSN: 2050-084X. https://doi.org/10.7554/eLife.55714 (July 2020)

Models test & unify experimental results; generate hypotheses









[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)

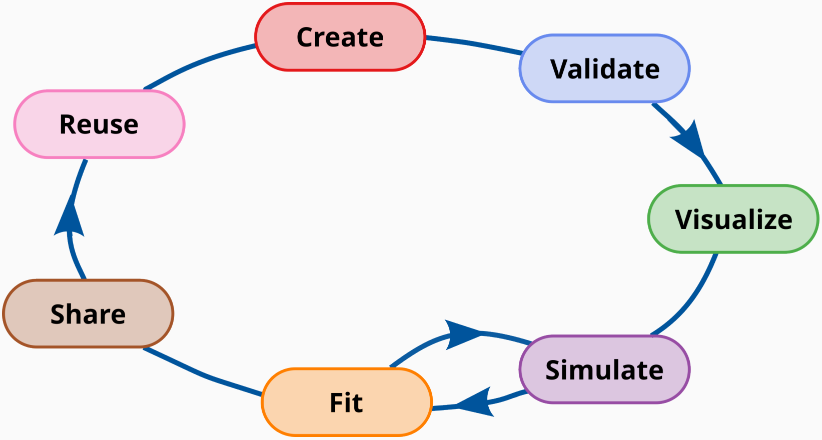
[1] Schirner, M. *et al.* **Learning how network structure shapes decision-making for bio-inspired computing.** *Nature Communications* **14.** ISSN: 2041-1723 (May 2023)

[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 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

incf 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://co.mbine.org/

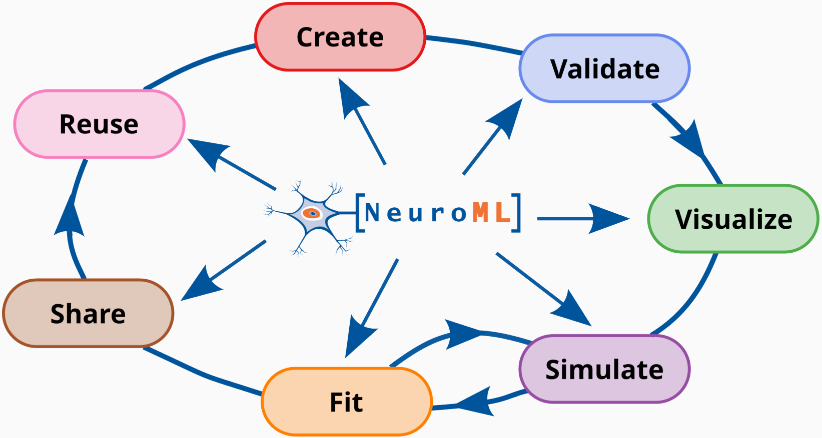
Standards enable FAIR neuroscience

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://co.mbine.org/

NeuroML ecosystem supports all stages of the model cycle



NeuroML ecosystem

- standard/specification
- software ecosystem

NeuroML schema/standard

Model specification (XSD)

- elements
- attributes
- hierarchical relationships

NeuroML schema/standard

Model specification (XSD)

- elements
- attributes
- hierarchical relationships

Dynamics (LEMS)

- dynamical behaviour

NeuroML schema/standard: XSD

Way of specifying the structure of an XML document.

- allows defining types and extensions/restrictions on types to create new types.

[1] https://www.w3.org/TR/xmlschema-1/

NeuroML schema/standard: XSD

Way of specifying the structure of an XML document.

- allows defining types and extensions/restrictions on types to create new types.

One can validate a model description against the schema before simulation

[1] https://www.w3.org/TR/xmlschema-1/

NeuroML schema/standard: specification: 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>
```

NeuroML schema/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

[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 schema/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
- but is not aware of the cable equation (required for multi-compartmental neuron models)

[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 schema/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 schema/standard: XSD and LEMS

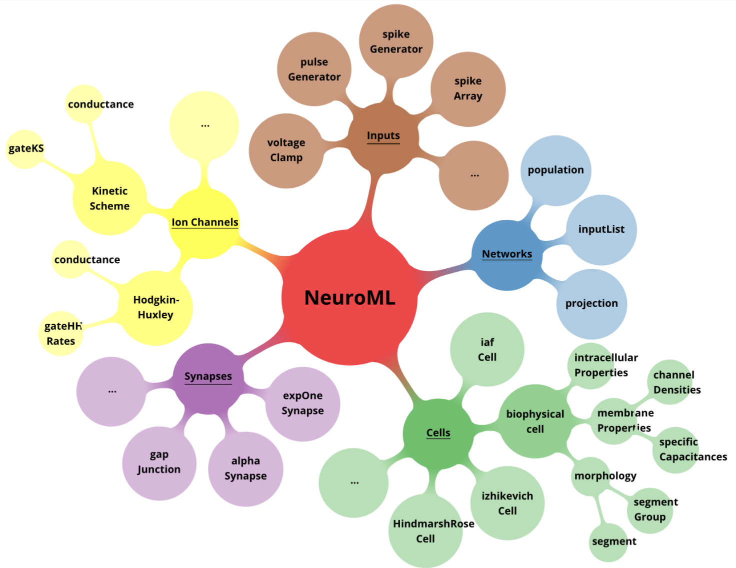
```
<xs:attribute name="v0" type="Nml2Quantity_voltage" use="required"/>
<xs:attribute name="k" type="Nml2Quantity_conductancePerVoltage" use="required"/>
<xs:attribute name="vr" type="Nml2Quantity_voltage" use="required"/>
<xs:attribute name="vt" type="Nml2Quantity_voltage" use="required"/>
<xs:attribute 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"/>
```

```
<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 schema/standard: dynamics (LEMS)

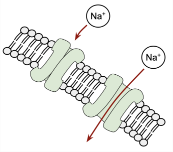
```
<ComponentType name="izhikevich2007Cell" extends="baseCellMembPotCap"
    description="Cell based ...">
    <!-- snipped -->
    <Dynamics>
        <StateVariable name="v" dimension="voltage" exposure="v"/>
        <StateVariable name="u" dimension="current" exposure="u"/>

        <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 provides a set of curated model elements

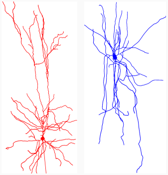


NeuroML is modular, structured and hierarchical

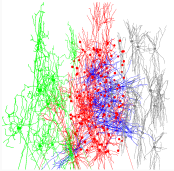
Conductances

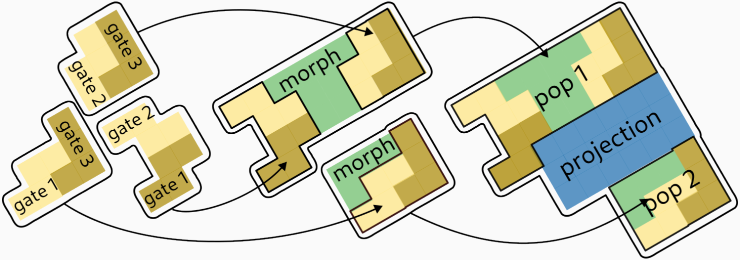


Cells

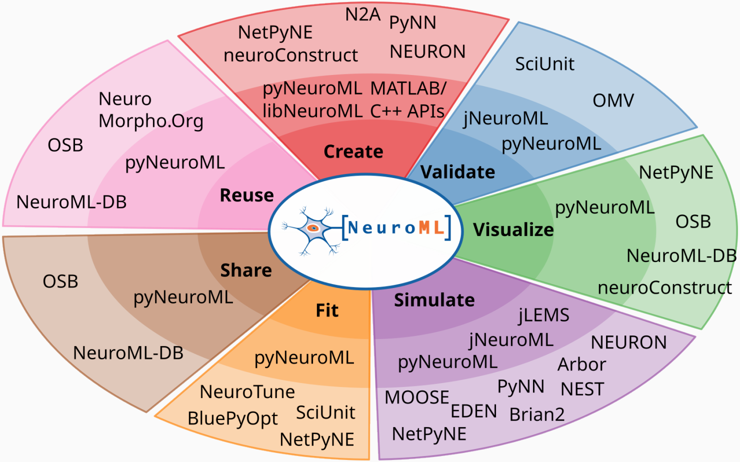


Networks





NeuroML: software ecosystem



NeuroML: software ecosystem: core tools

- Figure 4

NeuroML: create models

- Figure 5
- Code example

NeuroML: validate models

- Figure 6

NeuroML: visualise models

- Figure 7
- Figure 8
- Figure 9

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: the APIs

- Python API

NeuroML: Documentation

- Jupyterbook

NeuroML: projects

- GSoC, Outreachy, good computer science students

But, too many standards?

- XKCD here.