Standards and Tools in Neuroscience

A summary of the Open Source Brain workshop, September 2019

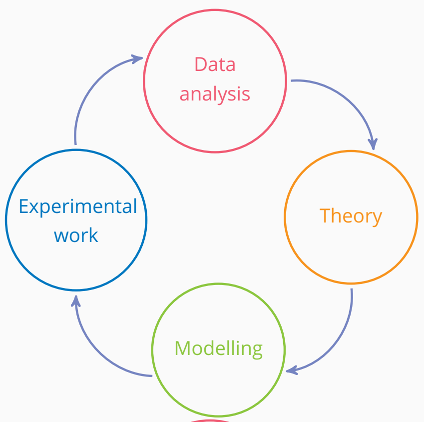
Ankur Sinha
Ph.D. candidate: UH Biocomputation Group, UK,
Volunteer: Fedora Project.

Notes

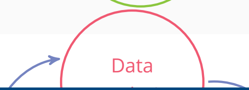
The problem statement

Notes

Neuroscience is complex, and massive



Notes



Free/Open Neuroscience

Free/Open science:
Scientific material should be easily, openly accessible to all.

Notes

Experimental neuroscience data is heterogeneous, multiscale and analysis is complex

Increasing spatial scale and complexity

Anatomy	Electrophysiology	Functional imaging	Behaviour
Receptor Immuno-histochemistry	Single/ensemble channel recordings	Synaptic imaging	Restricted task
Neuronal morphologies	Whole cell patch-clamp recordings	Single cell imaging	Freely moving
Brain mapping & Connectomes	Multielectrode array	Population imaging	Natural environment

How can we structure neuroscience data to facilitate reuse?

Notes

Models of brain function span multiple spatial scales

Behaviour

Brain

Network

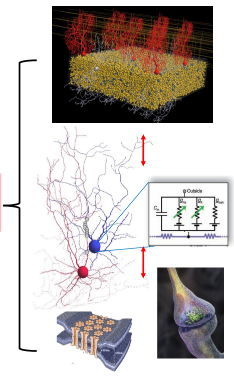
Neuron

Synapse

Molecular assemblies

Genes

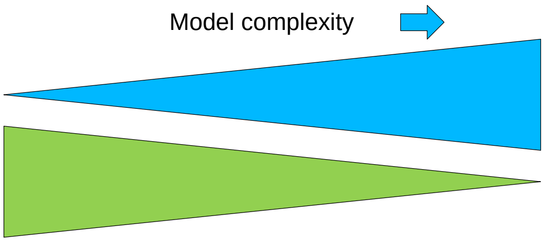
Biologically detailed multi-scale models of brain function.



Notes

A scaling problem

Model complexity



Transparency, accessibility, reproducibility, reuse…..and utility as a scientific tool

Notes

Standards: the common tongue

Notes

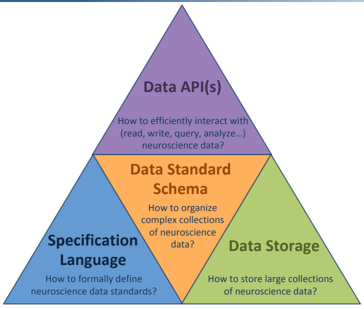
NWB:N 2.0: An Ecosystem for Neurophysiology Data Standardization

Oliver Rübel
Computational Research Division, Lawrence Berkeley National Laboratory

Open Source Brain Workshop
Alghero, Sardinia
September 10, 2019

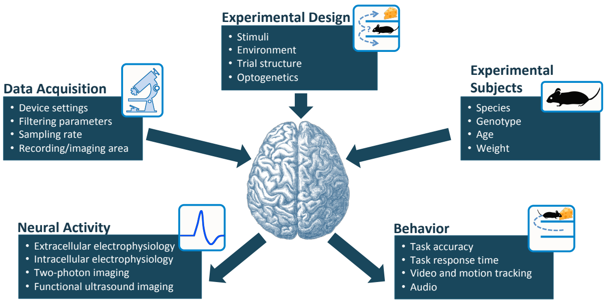
Notes

Main components of the NWB:N ecosystem



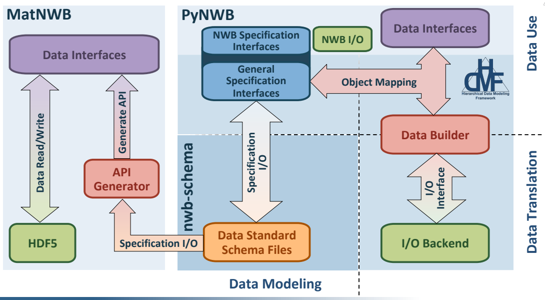
Notes

NWB:N supports complex collections of data required for understanding the brain



Notes

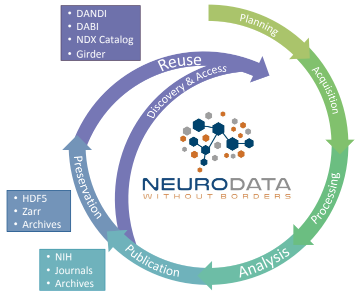
Advanced software architecture for data standardization



Notes

Neurophysiology Tools

NWB:N technologies at the heart of the neurodata lifecycle and applications



Electrophysiology	Optical Physiology
DataJoint, NWB Tutorials	
OpenEphys, Plexon, Neuralynx, Intan, TDT, SpikeGadgets, SpikeGLX	MiniScope ScanImage
PyNWB, MatNWB, HDMF, HDF5 tools/libs	
SpikeInterfaces MountainSort, KiloSort, npzsorting BrainStorm WaveClust, UltraMegaSort2000, KiloSort Kluster	NoRMCorre CNMF-E CELLMax EXTRACT
NWB-Explorer (OpenSourceBrain) NWB-JupyterWidgets	
BrainStorm, RAVE, Neo, SpikeWidgets, EcogVis, ephys-viz, ...	CalmAn

NEURODATA WITHOUT BORDERS

BERKELEY LAB

Notes

MetaCell

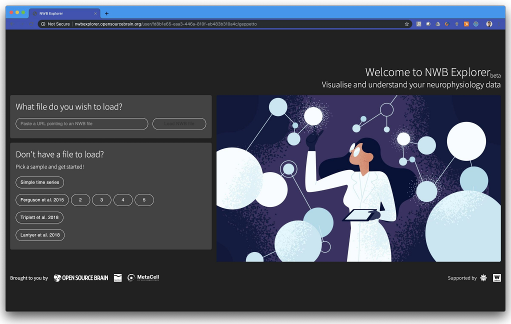
SOFTWARE FOR NEUROSCIENCE

NWB Explorer

Open Source Brain Meeting
Alghero, Italy
10th September 2019

OPEN SOURCE BRAIN

Notes



Notes

NWB EXPLORER
DEMO

Try it out at

https://nwbexplorer.opensourcebrain.org

Notes

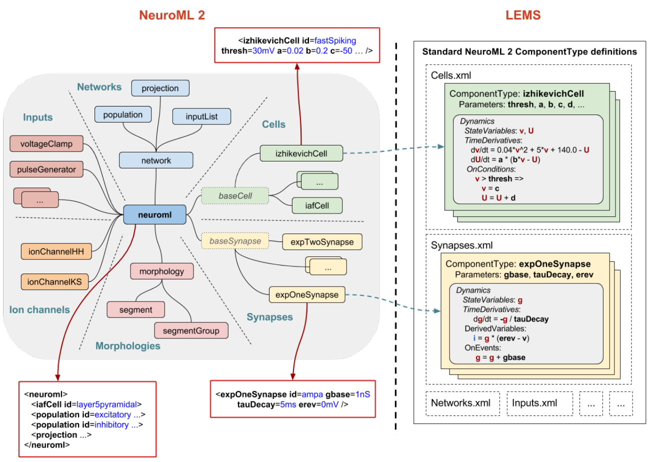
Creating cortical models across scales in NeuroML

Padraig Gleeson
p.gleeson@ucl.ac.uk
University College London

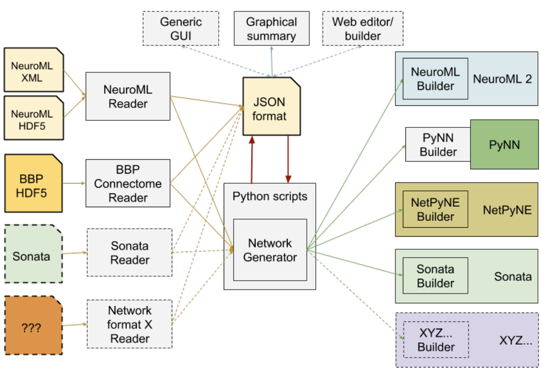
Open Source Brain workshop 2019

11th Sept 2019

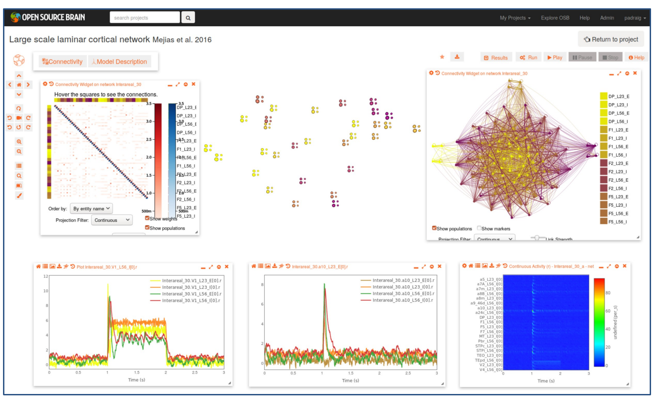
Notes



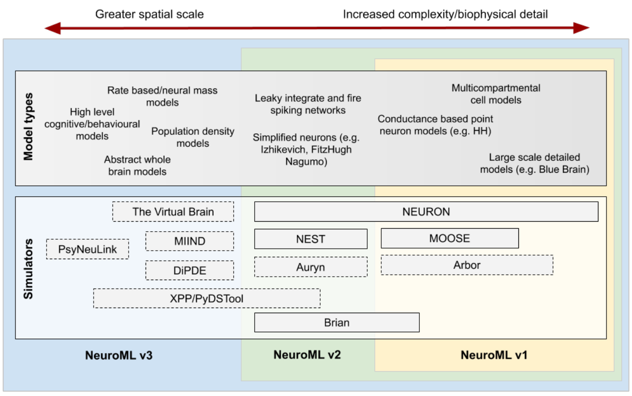
Notes



Notes



Notes



OPEN SOURCE BRAIN [NeuroML]

Notes

incf

About Network Activities Resources search

Home > Activities > Standards and best practices > INCF Special Interest Groups > INCF SIG on Standardised Representations of Network Structures

INCF SIG on Standardised Representations of Network Structures

This SIG deals with the various tools and formats for creating and sharing representations of biological neuronal networks, and will work towards ensuring these are as interoperable and usable as possible for computational neuroscientists.

Contact info: p.gleeson@uc.ac.uk

Members

Anton Arkhipov, Allen Institute, USA
Tom Close, Monash University, Australia
Sharon Crook, Arizona State University, USA
Kael Dai, Allen Institute, USA
Andrew Davison, UNIC, CNRS, France
Lia Domide, Codemart, Romania & Aix-Marseille Université, France
Salvador Durá-Bernal, SUNY Downstate Medical Center, USA
Viktor Jirsa, Aix-Marseille Université, France
Padraig Gleeson, University College London, UK

Notes

Converting simulator specific formats to NeuroML2

Open Source Brain Meeting 2019

Boris Marin

boris.marin@ufabc.edu.br

Universidade Federal do ABC

Notes

Converting NMODL to NeuroML

The Simple™, OSB sponsored way of converting models to NeuroML2

mailto:p.gleeson@ucl.ac.uk

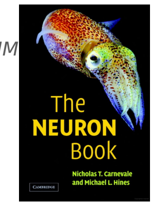
Notes

The *NEURON* simulator

Defining models in *NEURON*

https://www.neuron.yale.edu/neuron/

- Cells, Networks: *hoc* language (accessible from Python)
  - morphologies
  - synaptic connections
  - *.hoc* files
- Ion Channels (membrane mechanisms): *NM*
  - *.mod* files

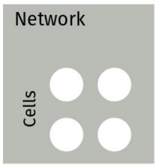


Notes

What is NeuroML, and why should I care?

Why can OSB process any NeuroML2 file?

- NML is *structured* (not unlike a *Type System*)

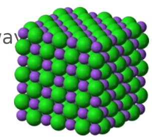


Notes

It is all about **Structure**

Structure in NeuroML / NMODL

- A *Type System* (composability rules) is what grants NML its superpowers
- `nmodl` is also powerful, but can be used as a general purpose language
  - `VERBATIM` blocks
  - many different ways of achieving same goal
  - prone to *unstructuredness*
- OSB could in theory treat `nmodl` the same wa NML...
  - .... if only people stuck to "good practices"!



Notes

Levels of Abstraction

NeuroML2

```xml
<ionChannelHH id="kChan" conductance="10pS" species="k">

    <gateHHrates id="n" instances="4">
      <forwardRate type="HHExpLinearRate" rate="0.1per_ms" midpoint="-55mV" scale="10mV"/>
      <reverseRate type="HHExpRate" rate="0.125per_ms" midpoint="-65mV" scale="-80mV"/>
    </gateHHrates>

  </ionChannelHH>
```

NMODL

```
BREAKPOINT {
        SOLVE states METHOD cnexp
        gk = gkbar * n ^ 4
        ik = gk * (v-ek)
}
INITIAL{
        n = alpha(v) / (alpha(v) + beta(v))
}
DERIVATIVE states{
        n' = (1 - n) * alpha(v) - n * beta(v)
}
```

```
FUNCTION alpha(Vm(mV))(/ms){
        LOCAL x
        UNITSOFF
        x = (Vm + 55) / 10
        if(fabs(x) > 1e-6){
                alpha=0.1*x/(1-exp(-x))
        }else{
                alpha=0.1/(1-0.5*x)
        }
        UNITSON
}
```

Notes

Levels of Abstraction

Declarative vs Imperative

- NeuroML2 operates (at least syntactically) closer to the level of abstraction employed by electrophysiologists
- The gory details exist, but elsewhere: *LEMS*
  - i.e. what to do with α, β; the definition of an `ExpRate`; how all of that is converted to conductances/currents…
- But we seldom need (want!) to interact with that level (look under the hood)

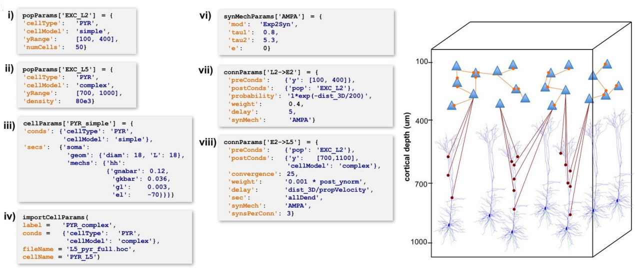
```
<Network ...>
    <Cell ...>
        <Channel ...>
            <Gate ...>
                 <Rate ...>
```

```
SOLVE{...} METHOD euler
...
DERIVATIVE {...}
...
FUNCTION trap(v){...}
```

Notes

Levels of Abstraction

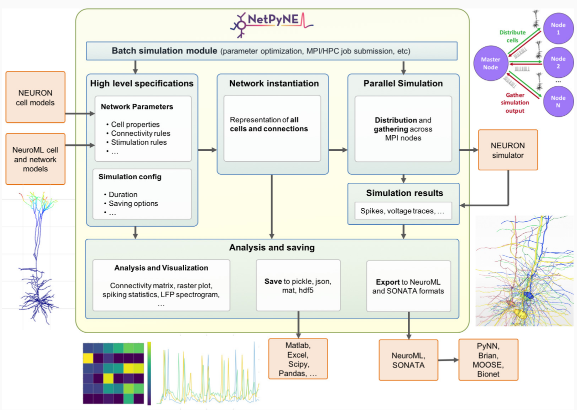
NetPyNE: structured network specification



Dura-Bernal, Salvador, et al. "NetPyNE, a tool for data-driven multiscale modeling of brain circuits." Elife 8

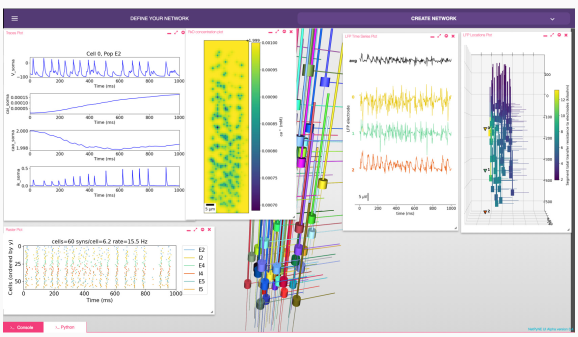
Notes

Netpyne workflow



Notes

Netpyne GUI

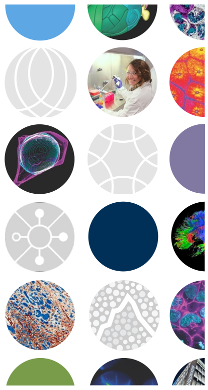


Notes

ALLEN INSTITUTE

Large-scale Datasets and Modeling Tools from the Allen Institute for Brain Science

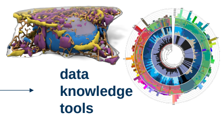
Yazan N. Billeh
yazanb@alleninstitute.org



Notes

ALLEN INSTITUTE | ALLEN INSTITUTE for BRAIN SCIENCE | ALLEN INSTITUTE for CELL SCIENCE | ALLEN INSTITUTE for IMMUNOLOGY | THE PAUL G. ALLEN FRONTIERS GROUP

- Established 2003 by Paul G. Allen
- South Lake Union, Seattle, WA
- 500 employees++



hard problems **complexity** **foundational biology** → **big science** **team science** **open science** → **data** **knowledge** **tools**

Notes

CORE PRINCIPLES

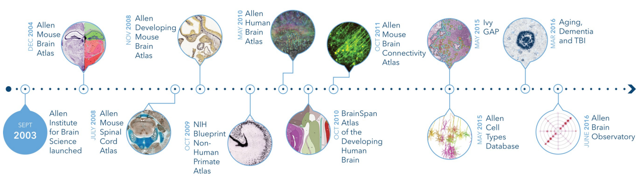
Team Science
Interdisciplinary teams working towards common goal

Big Science
Large-scale projects with robust, massive data

Open Science
All resources available online at brain-map.org or allencell.org

Notes

Allen Institute - Online Public Resources
www.brain-map.org



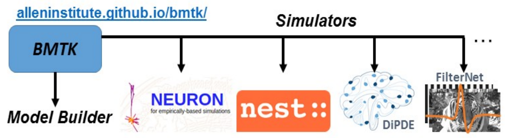
All data are:

- publicly accessible via API as soon as they pass QC
- freely available without any commercial restrictions

Notes

Our Models and Modeling Software Are Freely Available to the Community

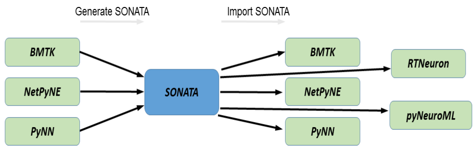
Brain Modeling ToolKit (BMTK): https://alleninstitute.github.io/bmtk/



Notes

Our Models and Modeling Software Are Freely Available to the Community

Scalable Open Network Architecture TemplAte (SONATA):
https://github.com/AllenInstitute/sonata
An interface between SONATA and the NWB format has been developed as well



Notes

Human Brain Project

How model standardization enables new tools and applications in neuroscientific research

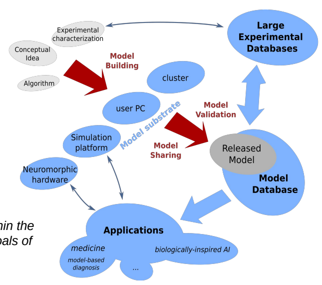
Insights from the HBP

Yann Zerlaut
Neuroinformatics team / group of A. Davison
Centre National de la Recherche Scientifique, France

Open Source Brain Meeting 2019, Alghero

Notes

Motivation



Model production pipeline within the infrastructure and research goals of the Human Brain Project

Notes

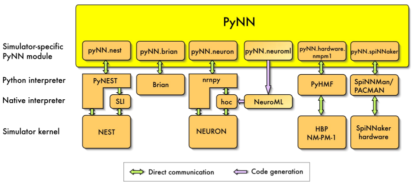
PyNN A unified interface for neuronal network simulators

Simulator-independent environments for developing neuroscience models:

- keep the advantages of having multiple simulators or hardware devices
- but remove the translation barrier.

Three (complementary) approaches:

- GUI (e.g. neuroConstruct)
- XML-based language (e.g. NeuroML, NineML)
- interpreted language (e.g. Python)



Notes

Sonata (pyNN support)

Large-scale simulation of biophysically-detailed neuronal circuits
→ sets specific constraints

the SONATA Data Format emerges as the standard optimized for performance for simulation, analysis and visualization of large-scale circuits
(joint initiative of Blue Brain Project and the Allen Institute for Brain Science)

Export to Sonata format

```
from pyNN.network import Network
from pyNN.serialization import export_to_sonata

sim.setup()
...
# create populations, projections, etc.
...

# add populations and projections to a Network
net = Network(pop1, pop2, ...., prj1, prj2, ...)

export_to_sonata(net, "sonata_output_dir")
```

Import from Sonata format

```
from pyNN.serialization import import_from_sonata, load_sonata_simulation_plan
import pyNN.neuron as sim

simulation_plan = load_sonata_simulation_plan("simulation_config.json")
simulation_plan.setup(sim)
net = import_from_sonata("circuit_config.json", sim)
simulation_plan.execute(net)
```

Notes

SciUnit *https://github.com/scidash/sciunit*

Include a validation framework in model development

✓ **What is SciUnit?**

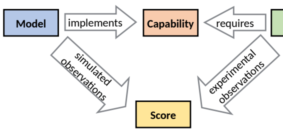
*A **Test**-driven framework for formally validating scientific models against data. It employs the concept of **Capabilities**.*

✓ **What are Tests?**

A procedure intended to establish the quality, performance, or reliability of a model

✓ **What are Capabilities?**

- *interfaces through which the model and the validation framework communicate*
- *implemented as methods (functions) within the model*



See Part II

Requires the participation of modellers:

- ✓ support to wrap your models for SciUnit
- ✓ add/request new tests to the library
- ✓ critique existing tests
- ✓ suggest new features

Notes

Test Packages

The overall test suite has been divided into a number of components, some containing validation tests specific to particular brain regions, others more generic. All validation tests are written in Python, using the SciUnit framework. Some of these are listed below:

Test suites for specific brain regions

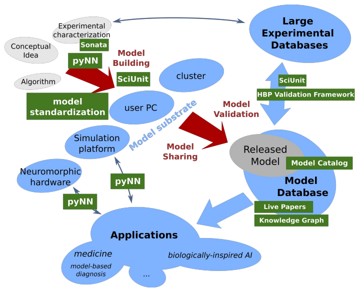
- ***HippoUnit***: `https://github.com/KaliLab/hippounit`
- ***HippoNetworkUnit***: `https://github.com/pedroernesto/HippoNetworkUnit`
- ***CerebUnit***: `https://github.com/lungsi/cerebellum-unit`
- ***BasalUnit***: `https://github.com/appukuttan-shailesh/basalunit`

Test suites for model features, independent of cell type or brain region

- ***MorphoUnit***: `https://github.com/appukuttan-shailesh/morphounit`
- ***NetworkUnit***: `https://github.com/mvonpapen/simrest_validation`
- ***eFELUnit***: `https://github.com/appukuttan-shailesh/eFELunit`

Notes

Summary



Notes

NeuroFedora: marketing pitch

Notes

Liaison between developers and users

Developers (upstream)

support | code — feedback | code

Distributions: Fedora/Debian . . .

support | binaries — feedback

End users (downstream)

Notes

Search: “NeuroFedora”

“Live” ISO now ready to download (demo)
Mailing list: neuro-sig@lists.fedoraproject.org
IRC: #fedora-neuro on Freenode
Telegram: t.me/NeuroFedora
Documentation neuro.fedoraproject.org
Blog: neuroblog.fedoraproject.org
Pagure.io (FOSS Git forge): neuro-sig/NeuroFedora

Notes

License

Notes

Notes

Notes

Notes