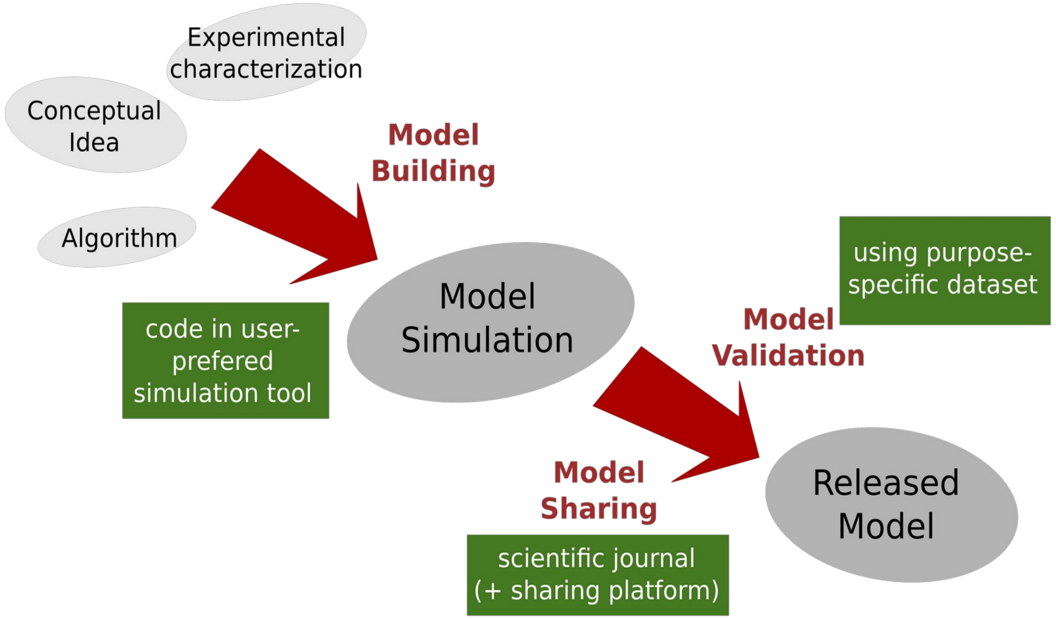
How model standardization enables new tools and applications in neuroscientific research

Insights from the HBP

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

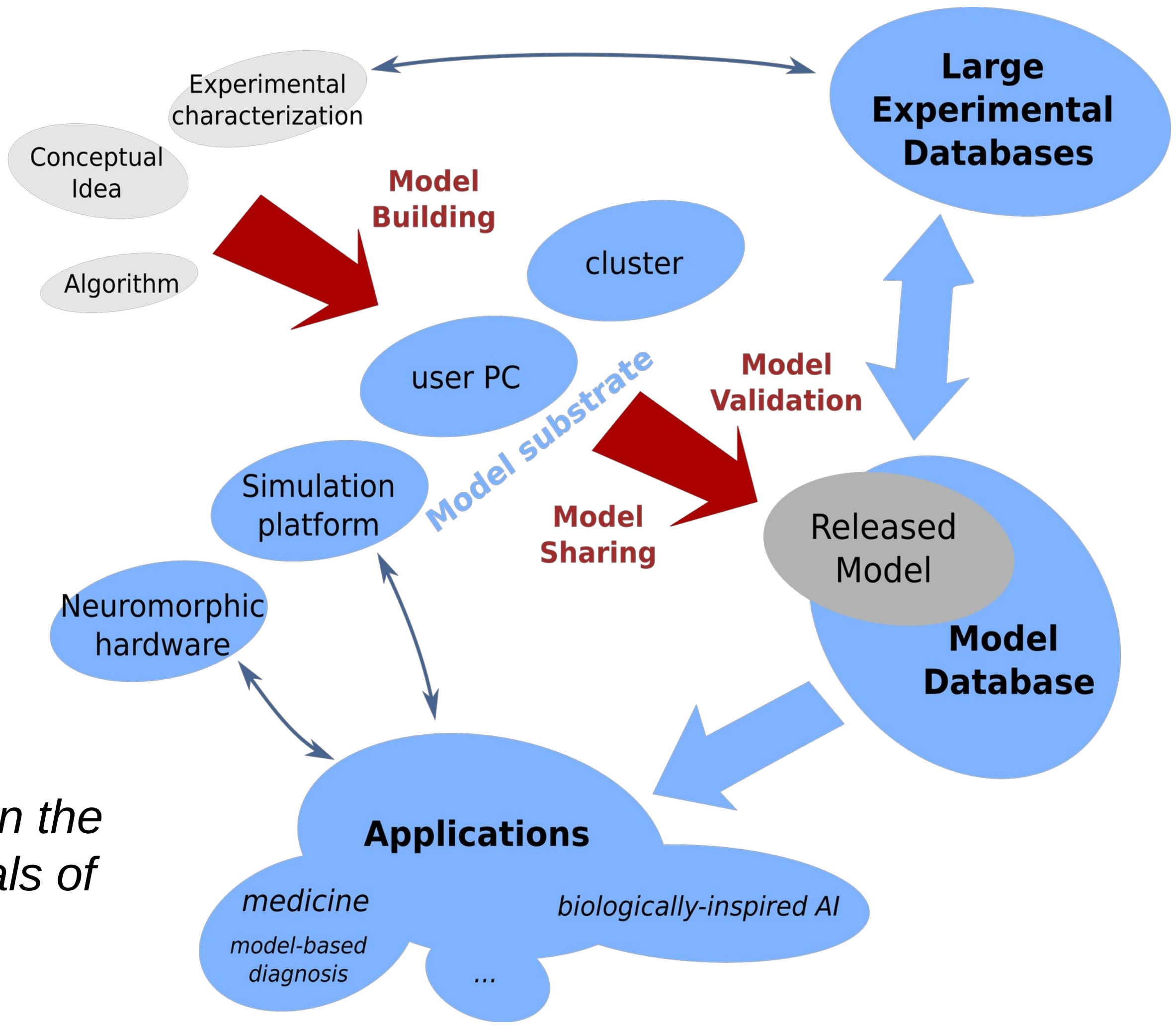
Open Source Brain Meeting 2019, Alghero

Motivation



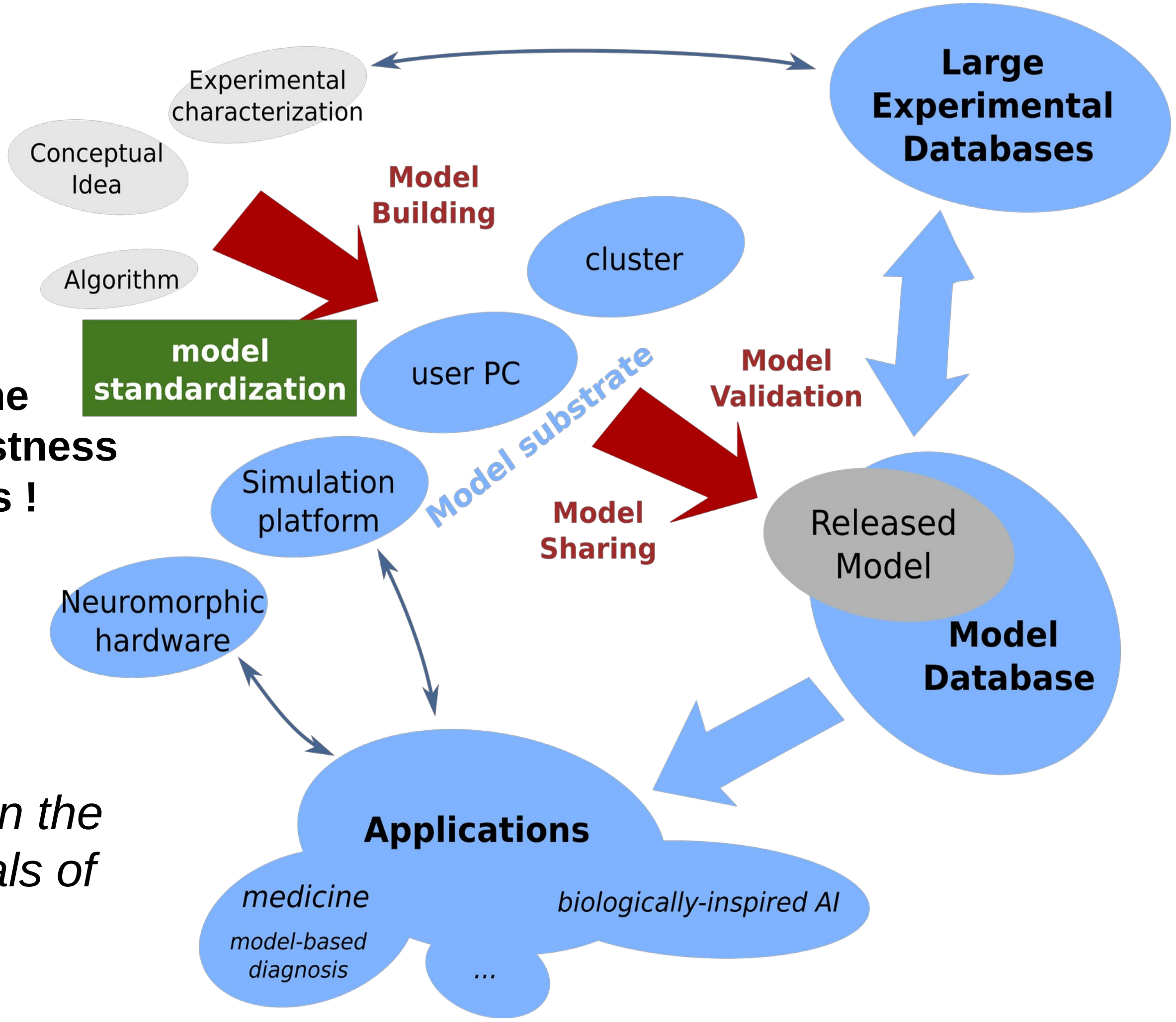
A typical model production pipeline in neuroscientific research

Motivation



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

Motivation



decisive to insure the feasibility and robustness of the entire process !

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

Overview

I) Model standardization:

- **model building**
  - PyNN
  - Sonata (pyNN.Sonata)
  - SciUnit
- **model release/sharing**
  - Model Catalog
  - Knowledge Graph
  - Live Papers

II) New tools and applications:

- Neuromorphic hardware
- Neurorobotics
- Brain Simulation Platform
- ***Model Validation framework***

PyNN A unified interface for neuronal network simulators

SenseLab ModelDB

Find Models by Simulation Environment

Click on a link to show a list of models implemented in that simulation environment or programming language.

Simulation Environment	Homepage	Number of models
BioPAX (web link to model)		1
Brian		4
C or C++ program		34
C or C++ program (web link to model)		19
CONTENT		1
CSIM		1
CSIM (web link to model)		3
CalC Calcium Calculator		1
CalC Calcium Calculator (web link to model)		7
Catacomb (web link to model)		1
CellExcite (web link to model)		1
CellML		0
CellML (web link to model)		1
Chemesis		2
Dynamics Solver		1
Emergent/PDP++		3
FORTRAN		4
FORTRAN (web link to a model)		1
GNUstep NeXTStep/OpenStep		1
Genesis		13
Genesis (web link to model)		7
IChMASCOT		0
IGOR Pro		3
IonChannelLab		0
Java		5
Java (web link to model)		2
KInNeSS (web link to model)		1
L-Neuron		0
MATLAB		67
MATLAB (web link to model)		34
MCell		1
MOOSE/PyMOOSE (web link to method)		1
MVASpike		1
MadSim		1
NCS		1
NEST (formerly BLISS/SYNOD)		2
NEURONPM (web link to tool)		2
NSL		0
Neosim		0
Network		1
NeuGen		0
Neuron		261
Neuron (web link to model)		14
NeuronC		0
NeuronetExperimenter (web link to model)		1
Octave		1
PCSIM		1
PSpice		2
Pascal (web link to model)		1
Pascal/Delphi		2
PyNN		2
Python		5
Python (web link to model)		1
QBasic/QuickBasic/Turbo Basic		2
QuB		1
R (web link to model)		1
SABER		1
SBML (web link to model)		1
SNNAP		21
SciLab		1
Scilab (web link to model)		2
Simulink		6
Sspice Symbolic SPICE		1
Surf-Hippo		0
Synthesis		0
Topographica		0
Topographica (web link to model)		1
Virtual Cell (web link to model)		3
XML (web link to model)		3
XNBC		0
XPP		50
XPP (web link to model)		7
neuroConstruct (web link to model)		1
parplex		2

Simulator diversity: problems and opportunities

Cons

- Considerable difficulty in translating models from one simulator to another...
- ...or even in understanding someone else's code.
- This:
  - impedes communication between investigators,
  - makes it harder to reproduce other people's work,
  - makes it harder to build on other people's work.

Pros

- Each simulator has a different balance between efficiency, flexibility, scalability and user-friendliness → can choose the most appropriate for a given problem.
- Any given simulator is likely to have bugs and hidden assumptions, which will be revealed by cross-checking results between different simulators → greater confidence in correctness of results.

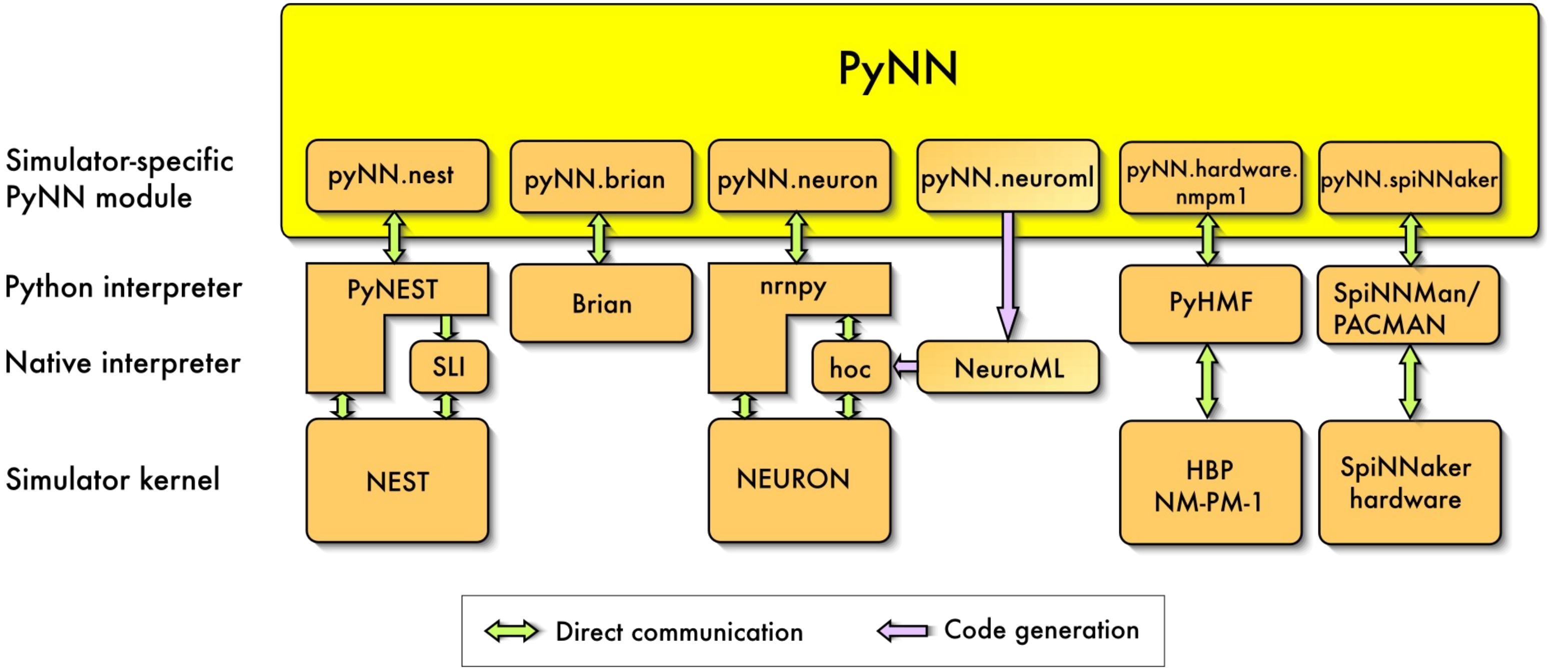
A unified interface for neuronal network simulators

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)



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

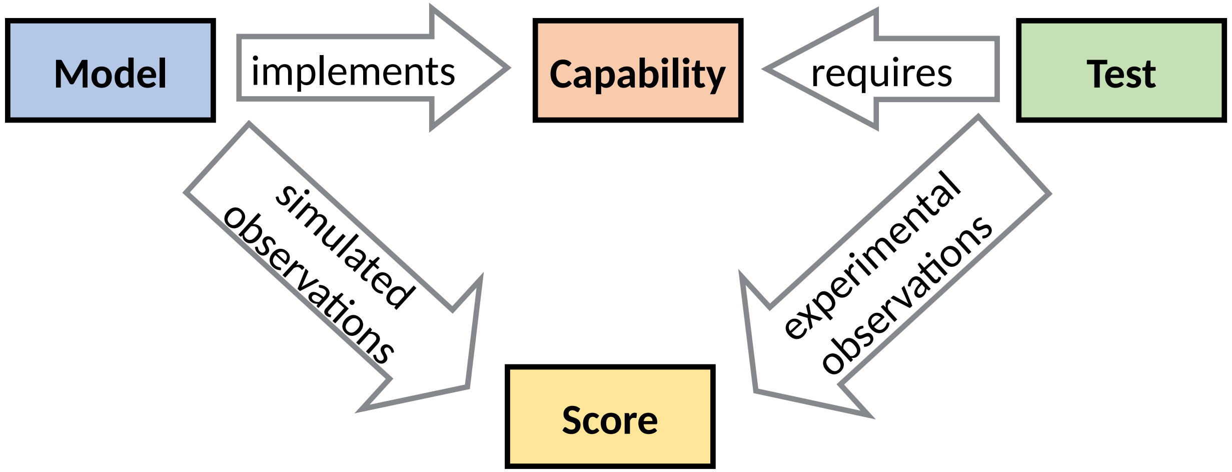
```
from pyNN.serialization import import_from_sonata, load_sonata_simulation_plan
import pyNN.neuron as sim

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)
```

SciUnit

https://github.com/scidash/sciunit

Include a validation framework in model development



See Part II

✓ **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*

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

Model Catalog App

Hosts the models developed and/or used within the ecosystem of tools provided by the HBP Platforms

Help

Model Catalog

Select: species ▾ | brain region ▾ | cell type ▾ | model scope ▾ | abstraction level ▾ | Select organization ▾ | Select privacy ▾ | Select collab ▾

Search:

Sort: Newest first

New model

Name	Alias	Species	Brain region	Cell type	Model scope	Abstraction level	Author(s)	Organization	Collab ID
Rall Morphology Model for Layer 5 Pyramidal Cells in Mouse V1	Rall-Morpho-L5PyrMouseV1	Mus musculus	cerebral cortex	L5 tufted pyramidal cell	single cell	spiking neurons	Zerlaut Yann	HBP-SP5	60843
msn_d2_morphologies		Mus musculus	striatum	medium spiny neuron (D2 type)	single cell	spiking neurons: biophysical	Alexander Kozlov	HBP-SP6	6351
msn_d1_morphologies		Mus musculus	striatum	medium spiny neuron (D1 type)	single cell	spiking neurons: biophysical	Alexander Kozlov	HBP-SP6	6351
fs_morphologies		Mus musculus	striatum	fast spiking interneuron	single cell	spiking neurons: biophysical	Alexander Kozlov	HBP-SP6	6351
str-fs-161205_FS1-BE37A-20180212		Mus	striatum	fast spiking	single cell	spiking	Ale		

Collaboration

Enables 3 levels of “sharing”:

- private: with user-defined collaborators
- public: visible to all users
- published: DOI + entry in the HBP Knowledge Graph

Two step process for publication:

1) Model submission by the users

2) Curation that the required “standards” are met

Model Catalog App

Model Versions

Name	ID	Created on
1.0	e1ee5736-7a29-42a6-aa1e-1ea56127c7d5	29/03/2018, 02:23

Description:

Parameters:

Morphology: ["https://object.cscs.ch/v1/AUTH_c0a333ecf7c045809321ce9d9ecdfdea/Migliore_2018_CA1/ View morphology

Code format: asc, py, hoc, mod, json

Download location: https://object.cscs.ch/v1/AUTH_c0a333ecf7c045809321ce9d9ecdfdea/hippocampus_optimization/optimizations/CA1_pyr_cACpyr_oh

rkspace Model Catalog

CA1_pyr_cACpyr_oh140807_A0_idG_20170915113354

Author(s): Rosanna Migliore

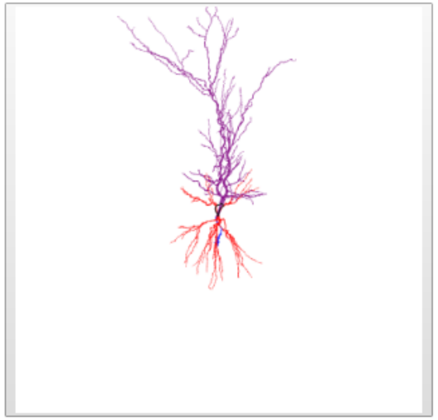
Permissions

Read and Write

Home Edit New version

Model Description

Single cell model optimized using BluePyOpt



Morphology

Model information

Public

ID: d4272705-bc99-4c4a-9e4d-3ce114382b90

Collab ID: 9821

Organization: HBP-SP6

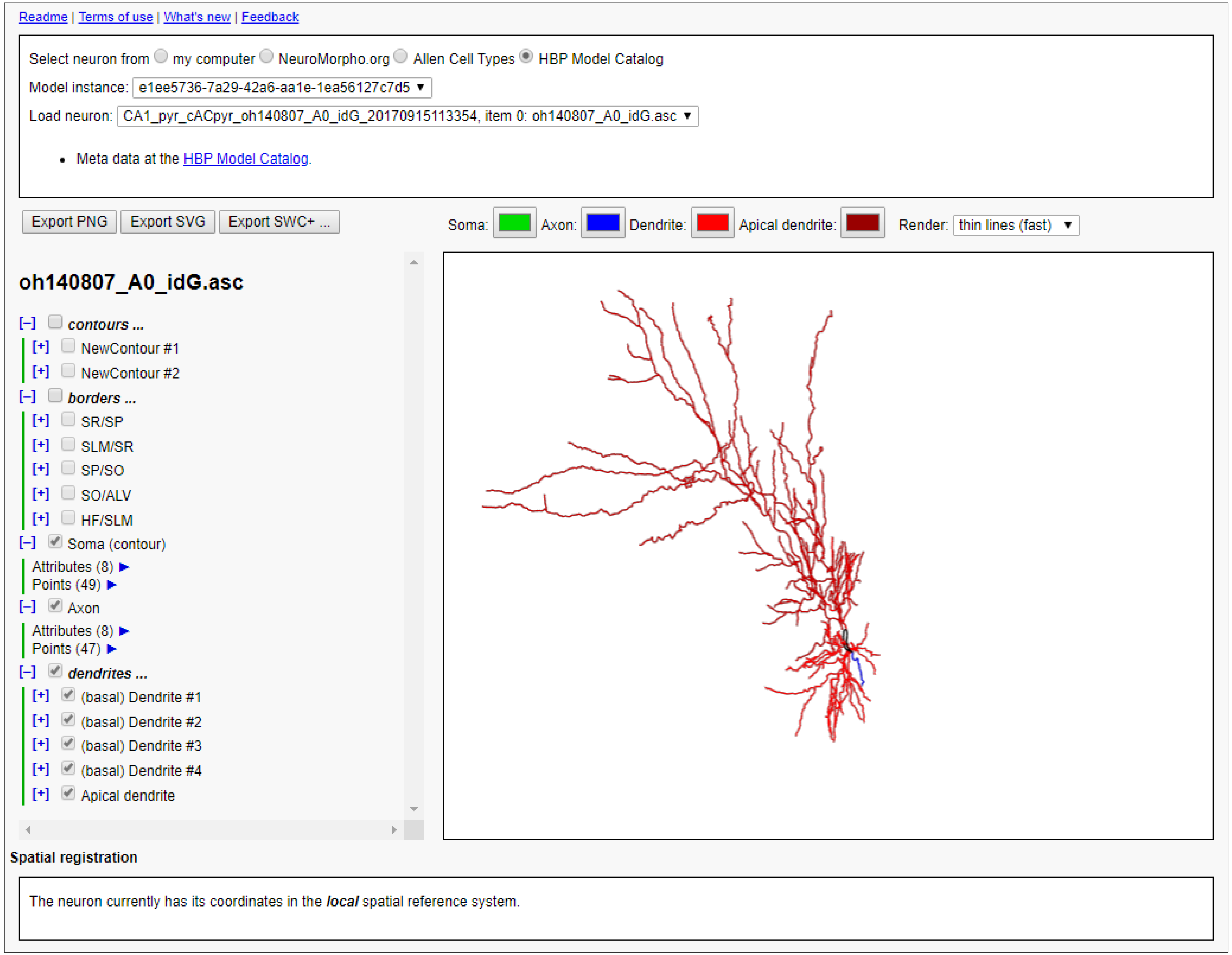
Species: Rat (Rattus rattus)

Brain region: Hippocampus

Model scope: Single cell model

Cell type: Pyramidal Cell

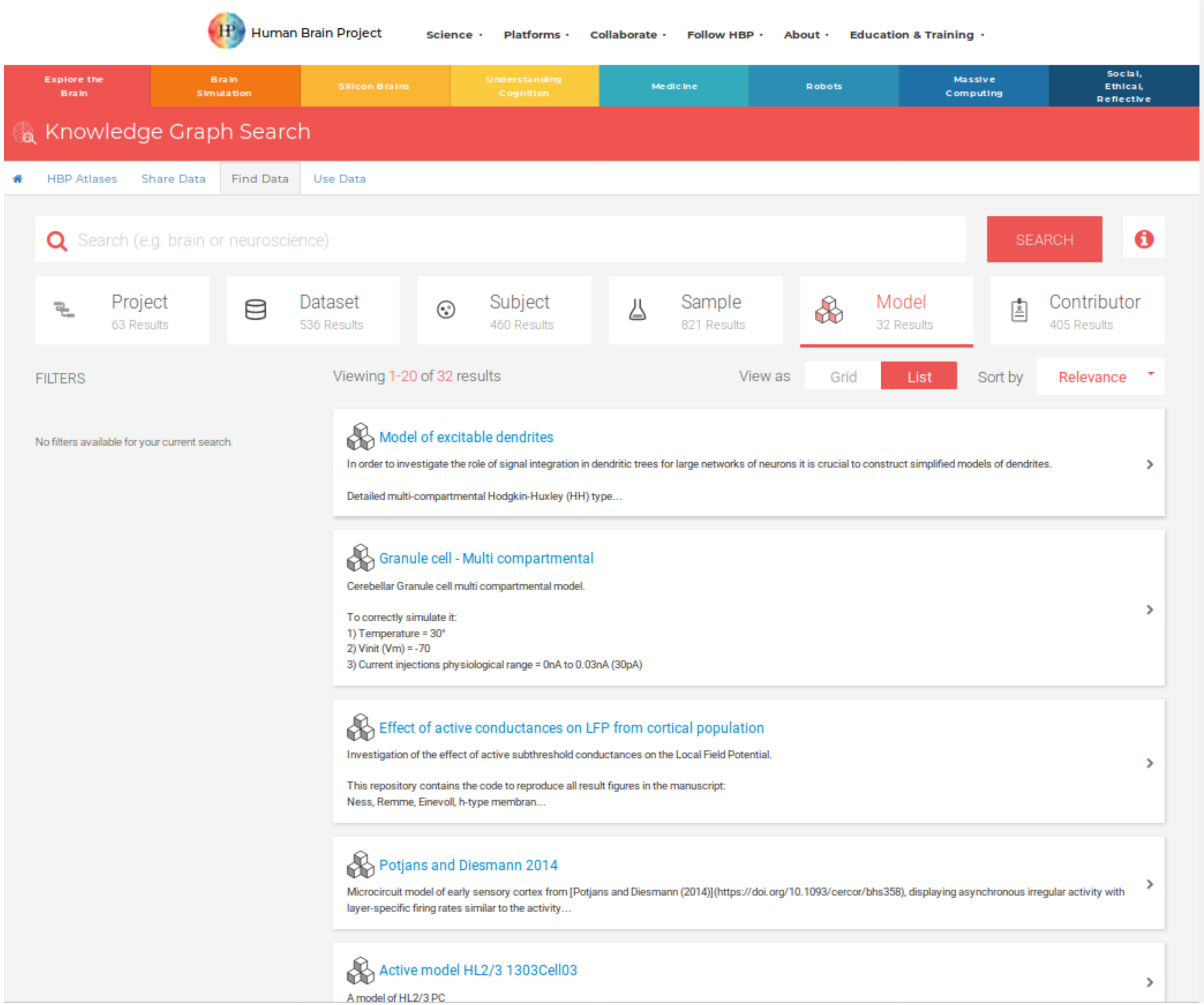
Model Catalog App – Morphology Viewer



*Thanks to **Rembrandt Bakker** (Jülich) for his support on this task.*

HBP Knowledge Graph

https://www.humanbrainproject.eu/en/explore-the-brain/search

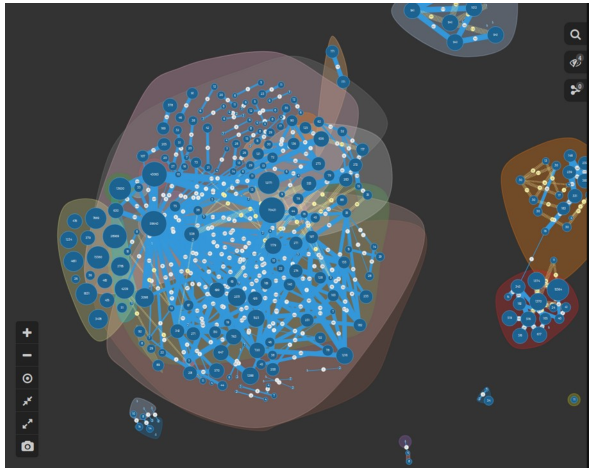


Technical support that provides the required information to make research data:

- Findable
- Accessible
- Interoperable
- Reusable

(FAIR)

Linked data, Relational database



Live papers

https://www.humanbrainproject.eu/en/brain-simulation/live-papers/

The Brain Simulation Platform "Live Papers"

The physiological variability of channel density in hippocampal CA1 pyramidal cells and interneurons explored using a unified data-driven modeling workflow

Authors: Rosanna Migliore [1], Carmen A. Lupascu [1], Luca L. Bologna [1], Armando Romani [2], Jean-Denis Courcol [2], Stefano Antonel [2], Werner A.H. Van Geit [2], Alex M. Thomson [3], Audrey Mercer [3], Sigrun Lange [3,4], Joanne Falck [3], Christian A. Rössert [2], Ying Shi [2], Olivier Hagens [5], Maurizio Pezzoli [5], Tamas F. Freund [6,7], Szabolcs Kali [6,7], Eilif B. Muller [2], Felix Schürmann [2], Henry Markram [2], and Michele Migliore [1]

Author information: [1] Institute of Biophysics, National Research Council, Palermo, Italy, [2] Blue Brain Project, École Polytechnique Fédérale de Lausanne, Campus Biotech, Geneva, Switzerland, [3] University College London, United Kingdom, [4] University of Westminster, London, United Kingdom, [5] Laboratory of Neural Microcircuitry (LNMC), Brain Mind Institute, EPFL, Lausanne, Switzerland, [6] Institute of Experimental Medicine, Hungarian Academy of Sciences, Budapest, Hungary, [7] Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Budapest, Hungary.

Corresponding author: Rosanna Migliore (rosanna.migliore@cnr.it)

Journal: Plos Computational Biology

Download Url: https://doi.org/10.1371/journal.pcbi.1006423

Citation: Migliore R, Lupascu CA, Bologna LL, Romani A, Courcol J-D, Antonel S, et al. (2018) The physiological variability of channel density in hippocampal CA1 pyramidal cells and interneurons explored using a unified data-driven modeling workflow. PLoS Comput Biol 14(9): e1006423.

DOI: https://doi.org/10.1371/journal.pcbi.1006423

Abstract:

The peak conductance of many ion channel types measured in any given animal is highly variable across neurons, both within and between neuronal populations. The current view is that this occurs because a neuron needs to adapt its intrinsic electrophysiological properties either to maintain the same operative range in the presence of abnormal inputs or to compensate for the effects of pathological conditions. Limited experimental and modeling evidence suggests this might be implemented via the correlation and/or degeneracy in the function of multiple types of conductances. To study this mechanism in hippocampal CA1 neurons and interneurons, we systematically generated a set of morphologically and biophysically accurate models. We then analyzed the ensembles of peak conductance obtained for each model neuron. The results suggest that the set of conductances expressed in the various neuron types may be divided into two groups: one group is responsible for the main characteristics of the firing behavior in each population and the other more involved with degeneracy. These models provide experimentally testable predictions on the combination and relative proportion of the different conductance types that should be present in hippocampal CA1 pyramidal cells and interneurons.

Resources

Data and models: all data and models used in the paper are available at the links reported below, grouped into the following categories:

- Morphologies
- Electrophysiological Traces
- ModelDB link and test simulations
- Optimizations

Tools and applications

Neuromorphic hardware

Neuromorphic computing implements aspects of biological neural networks as analogue or digital copies on electronic circuits.

- offers a tool to understand the dynamic processes of learning and development
- applying brain inspiration to generic cognitive computing

Key advantages:
- *energy efficiency*
- *execution speed*
- *robustness against local failures*
- *ability to learn*

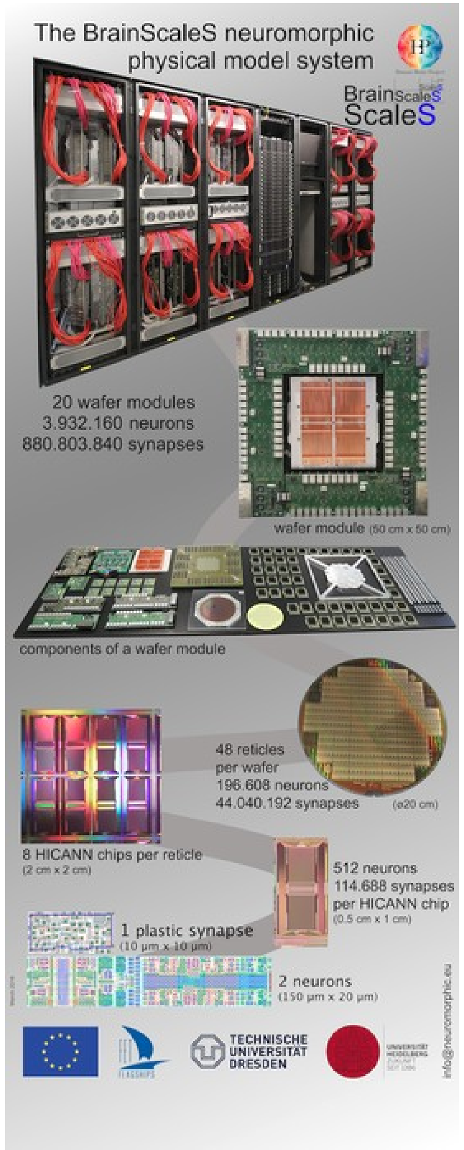
A common pyNN interface :

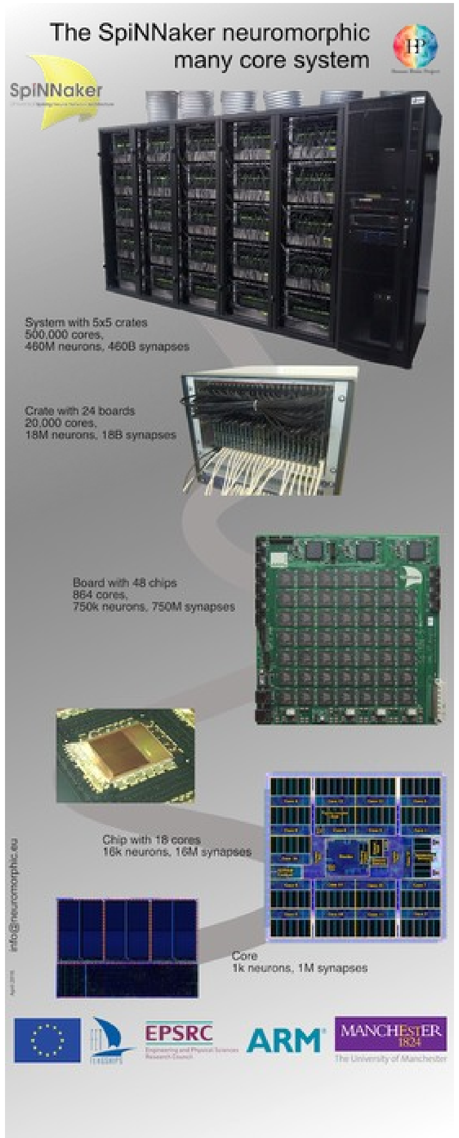
import pyNN.BrainScaleS as sim

//

import pyNN.SpiNNaker as sim

Two complementary technologies:

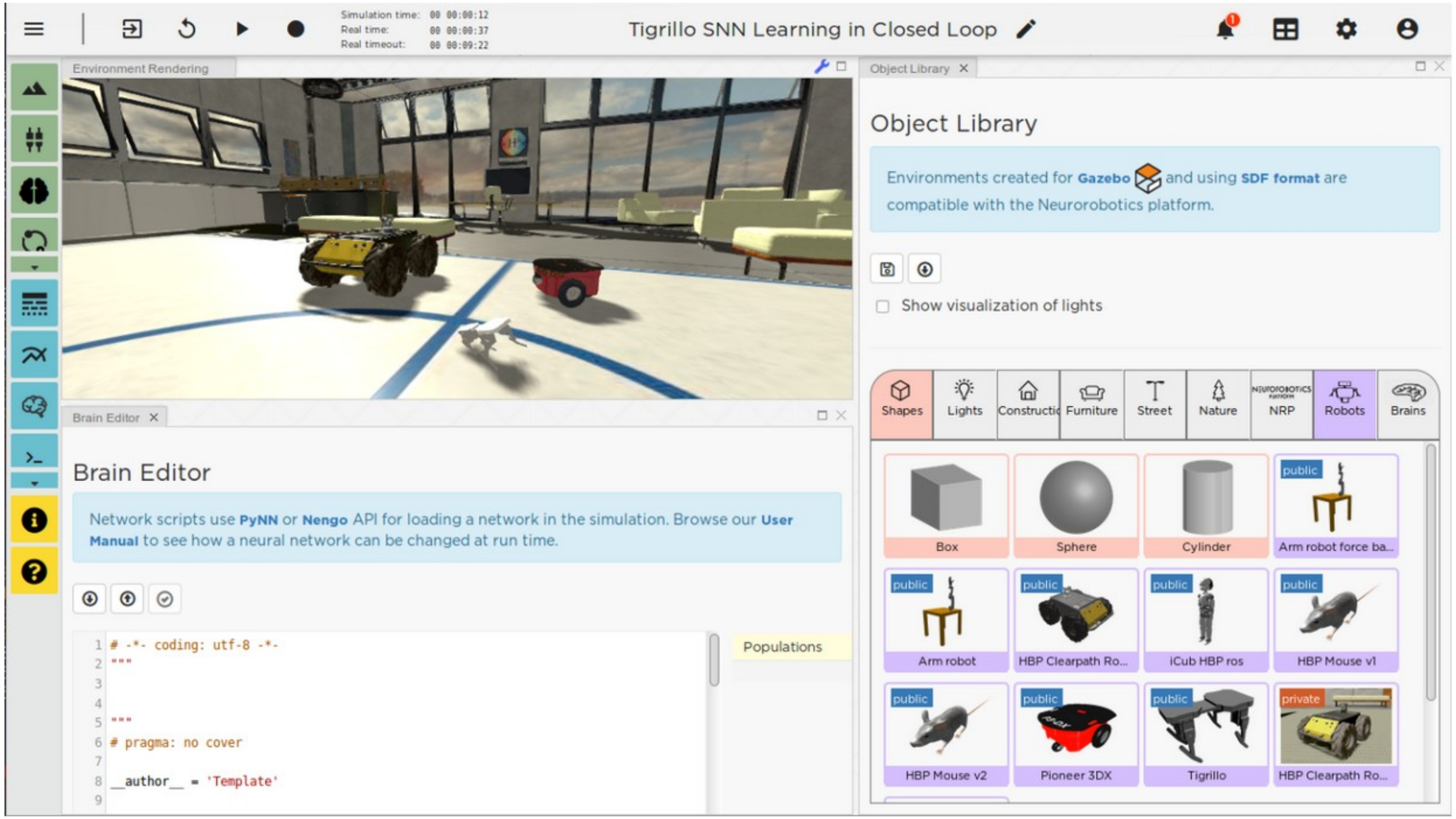




Neurorobotics

https://www.humanbrainproject.eu/en/robots/

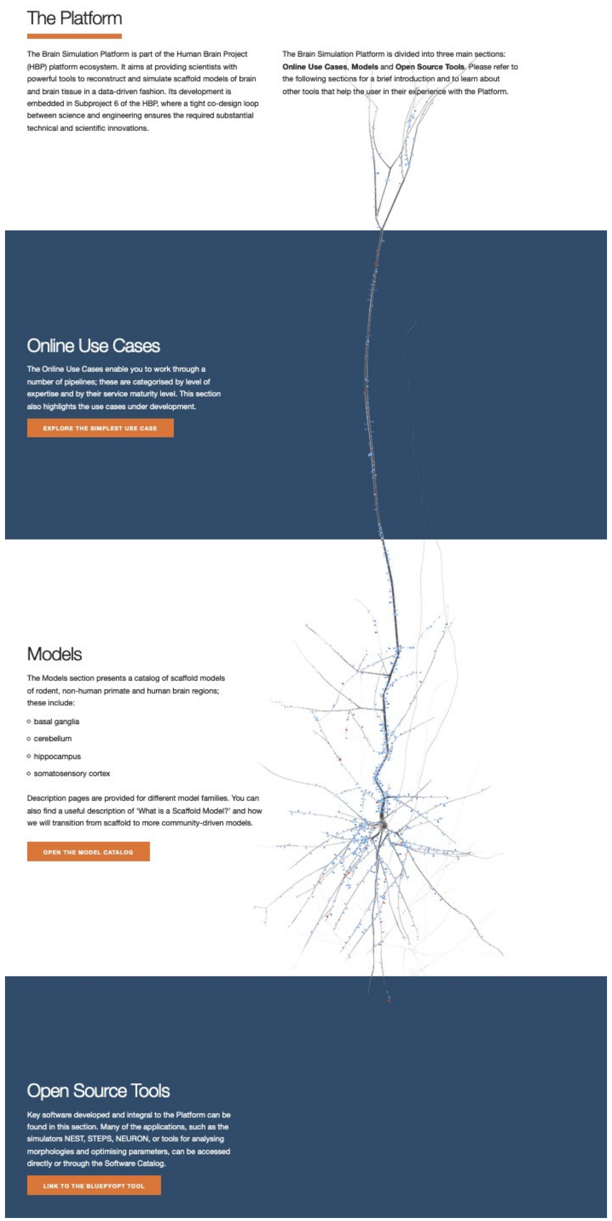
allows researchers to give any simulated brain model its own “body” — virtual or even real — and explore how it controls movement, reacts to stimulus and learns in a virtual environment



The Platform is public, online, and available for all researchers who want to test their brain models or to build the brain-inspired robots of the future.

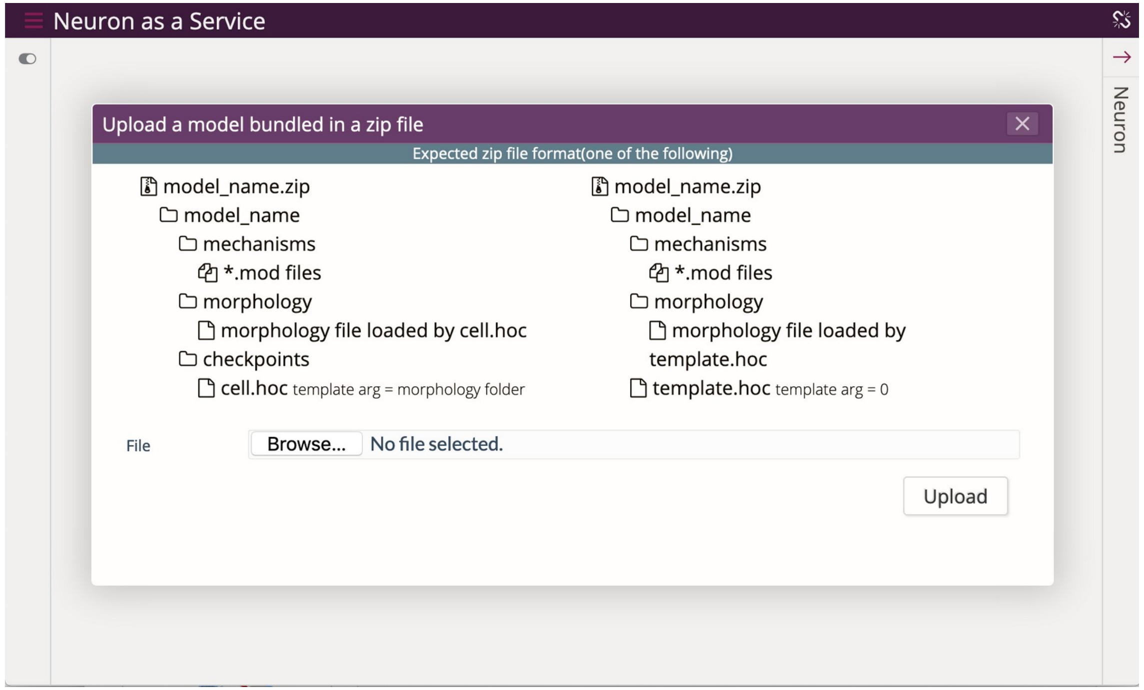
The Brain Simulation Platform

https://www.humanbrainproject.eu/en/brain-simulation/brain-simulation-platform/



The Brain Simulation Platform is an internet-accessible collaborative Platform designed for reconstruction and simulation of brain models.

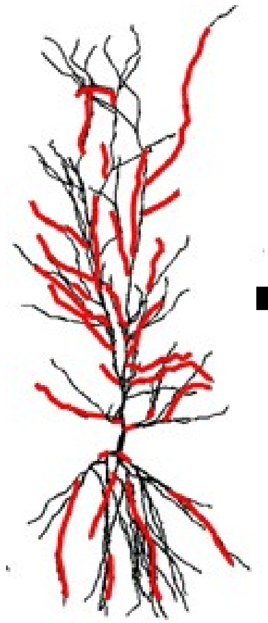
Neuron as a Service (Blue NaaS)



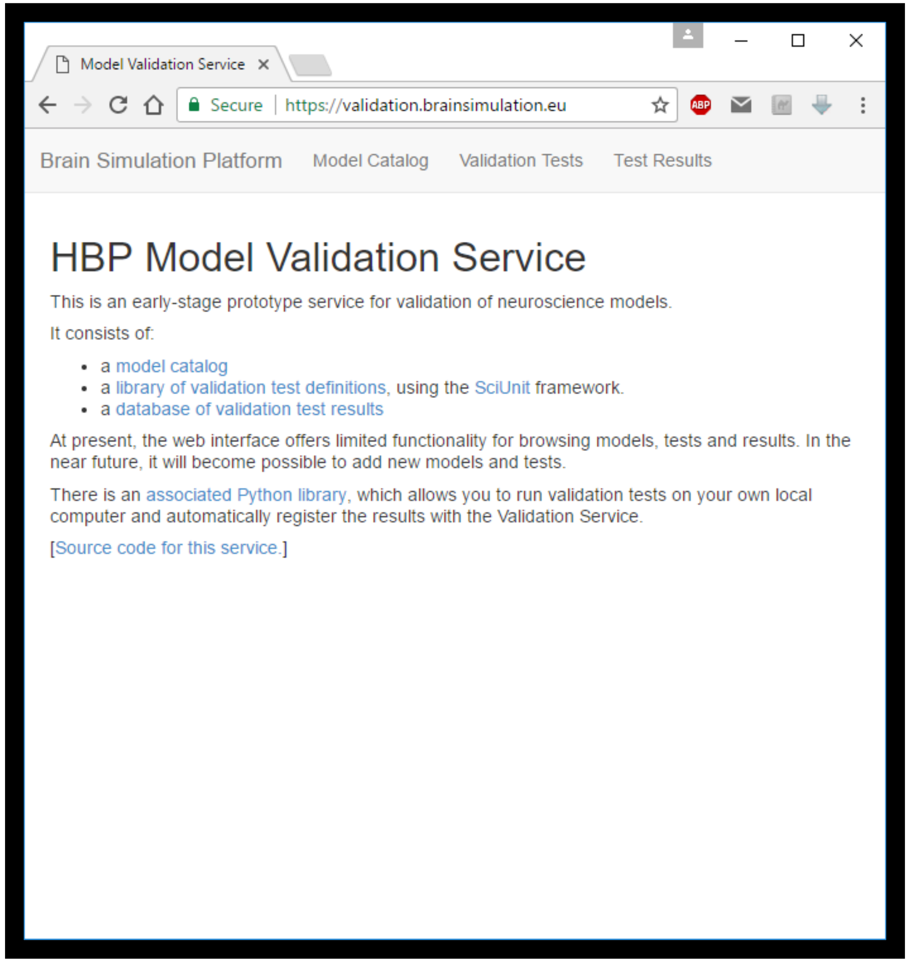
https://blue-naas.humanbrainproject.eu/

The Model Validation Framework

My_model_v2



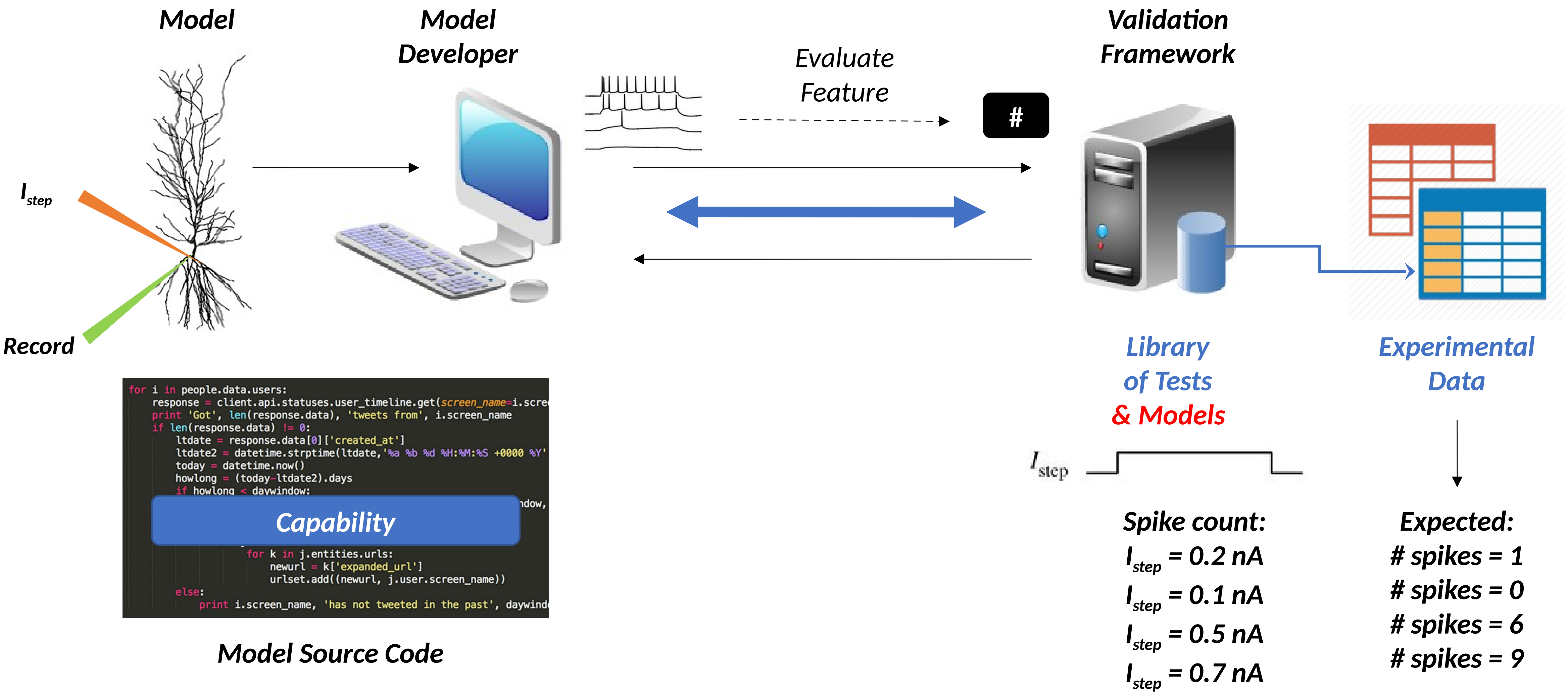
Adds a new type of ionic channels with respect to: <u>My_model_v1</u>



Test (against data)	Score	Result
Resting Membrane Potential	0.01	Pass
Input Resistance	0.20	Pass
Action Potential Height	0.05	Pass
Oblique Integration	4.73	Fail
		ass
		ail

Model Name	Net Score	Tests Passed
CodeJam et al., 2018	2.73	18
Xyz et al., 2010	3.25	15
My_model_v2	3.51	15
Random et al., 2014	4.73	12
My_model_v1	5.02	10
Abcde et al., 2017	5.09	10
⋮		

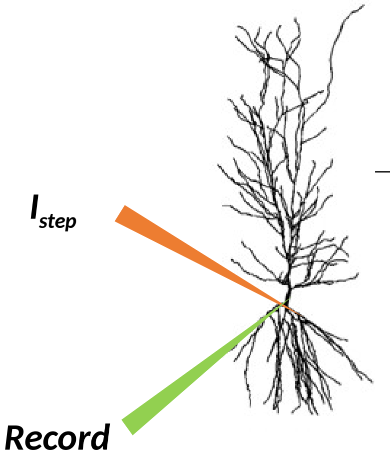
Validation Framework – Quick Overview



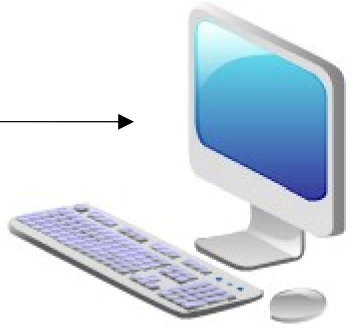
Images (adapted from): https://www.linkedin.com/pulse/source-codes-required-handover-govt-china-la-isms-dlip-ciseh-itil-

Validation Framework – Python Client

Model

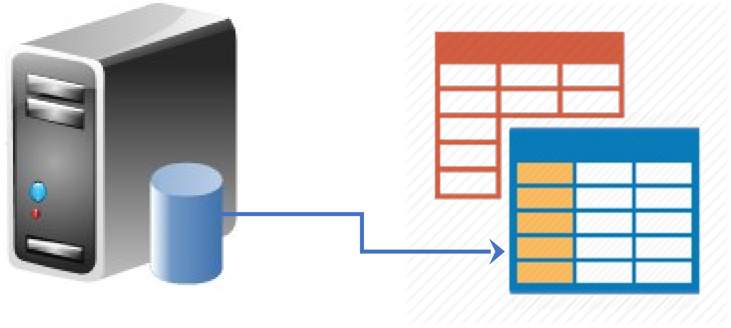


Model Developer



python

Validation Framework



Leveraging the functionality offered by two existing tools:

Framework for test-driven validation of scientific model

For data-driven validation of neuron and ion channel models using SciUnit

MyModel

Implements Cap.:
Inject Current
Record Potential

Test_SpikeCount

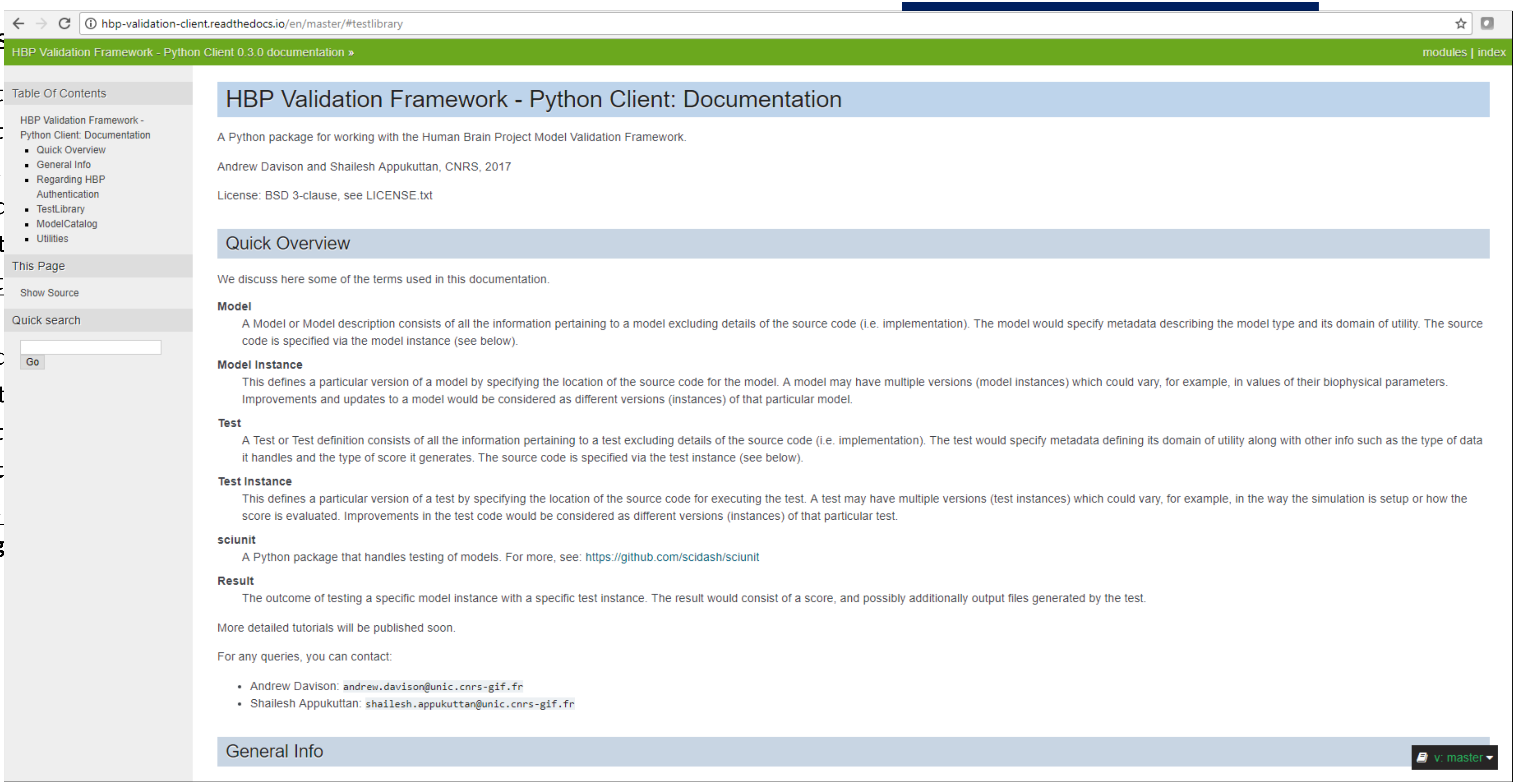
Req. Capabilities:
Inject Current
Record Potential

Python Client

http://www.hbp-validation-client.readthedocs.io

Tes

Get
Get
List
Add
Edit
Get
List
Add
Edit
Get
Get
List
Reg


hbp-validation-client.readthedocs.io/en/master/#testlibrary

HBP Validation Framework - Python Client 0.3.0 documentation »

modules | index

Table Of Contents

HBP Validation Framework - Python Client: Documentation

- Quick Overview
- General Info
- Regarding HBP Authentication
- TestLibrary
- ModelCatalog
- Utilities

This Page

Show Source

Quick search

Go

HBP Validation Framework - Python Client: Documentation

A Python package for working with the Human Brain Project Model Validation Framework.

Andrew Davison and Shailesh Appukuttan, CNRS, 2017

License: BSD 3-clause, see LICENSE.txt

Quick Overview

We discuss here some of the terms used in this documentation.

Model
A Model or Model description consists of all the information pertaining to a model excluding details of the source code (i.e. implementation). The model would specify metadata describing the model type and its domain of utility. The source code is specified via the model instance (see below).

Model Instance
This defines a particular version of a model by specifying the location of the source code for the model. A model may have multiple versions (model instances) which could vary, for example, in values of their biophysical parameters. Improvements and updates to a model would be considered as different versions (instances) of that particular model.

Test
A Test or Test definition consists of all the information pertaining to a test excluding details of the source code (i.e. implementation). The test would specify metadata defining its domain of utility along with other info such as the type of data it handles and the type of score it generates. The source code is specified via the test instance (see below).

Test Instance
This defines a particular version of a test by specifying the location of the source code for executing the test. A test may have multiple versions (test instances) which could vary, for example, in the way the simulation is setup or how the score is evaluated. Improvements in the test code would be considered as different versions (instances) of that particular test.

sciunit
A Python package that handles testing of models. For more, see: https://github.com/scidash/sciunit

Result
The outcome of testing a specific model instance with a specific test instance. The result would consist of a score, and possibly additionally output files generated by the test.

More detailed tutorials will be published soon.

For any queries, you can contact:

- Andrew Davison: `andrew.davison@unic.cnrs-gif.fr`
- Shailesh Appukuttan: `shailesh.appukuttan@unic.cnrs-gif.fr`

General Info

v: master

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`

HippoUnit

Targeted at electrophysiological validations on detailed hippocampal CA1 pyramidal cell model. Simulation outcomes are tested closely against experimental findings. It currently comprises the following tests:

- **Somatic Features Test**
  Experimental data: Migliore et al. 2018

- **Depolarization Block Test**
  Experimental data: Bianchi et al. 2012

- **Backpropagating Action Potential (AP) Test**
  Experimental data: Golding et al. 2001

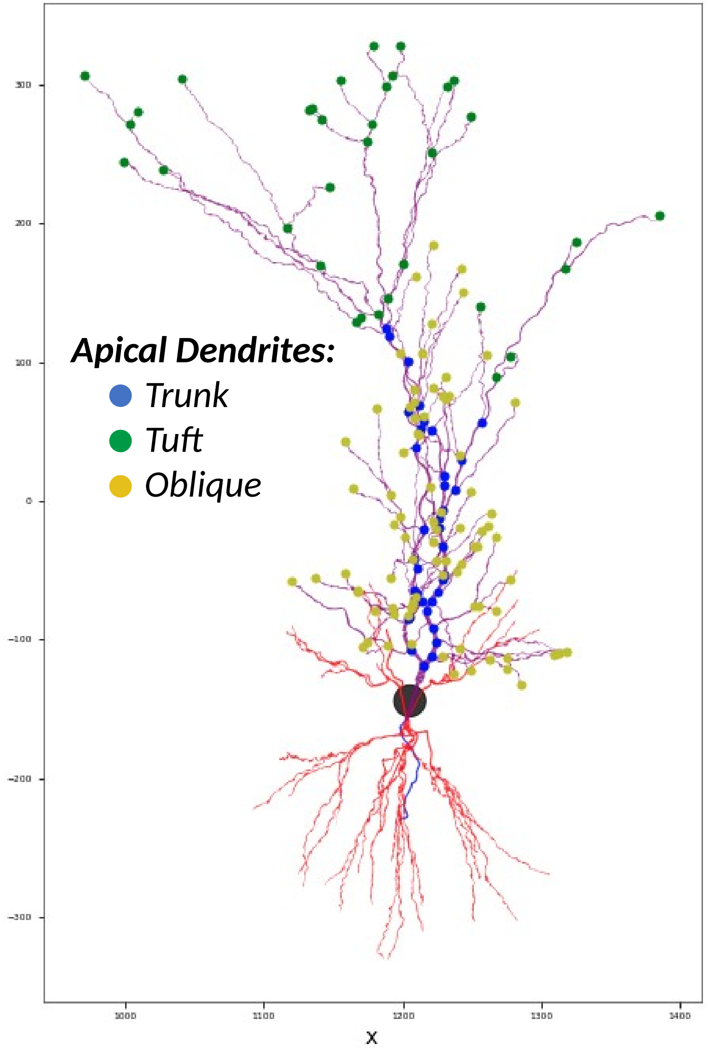
- **Post-Synaptic Potential (PSP) Attenuation Test**
  Experimental data: Magee & Cook 2000

- **Oblique Integration Test**
  Experimental data: Losonczy & Magee 2006

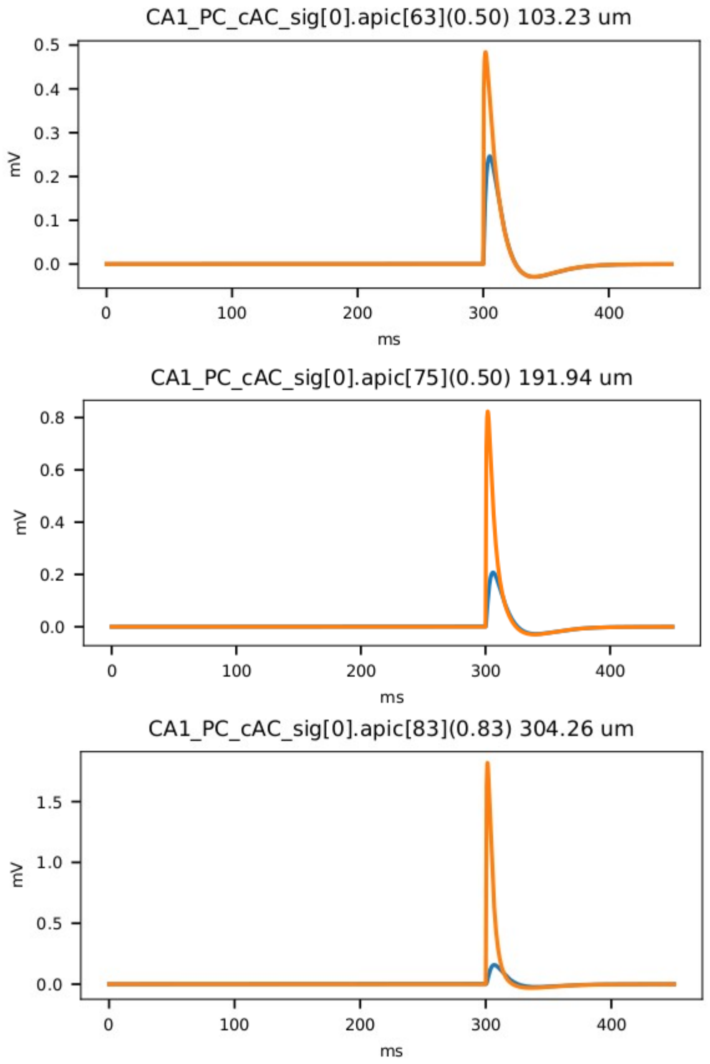
Developed by
***Sára Saray** and **Szabolcs Káli** (IEM, HAS)*

Post-Synaptic Potential (PSP) Attenuation Test

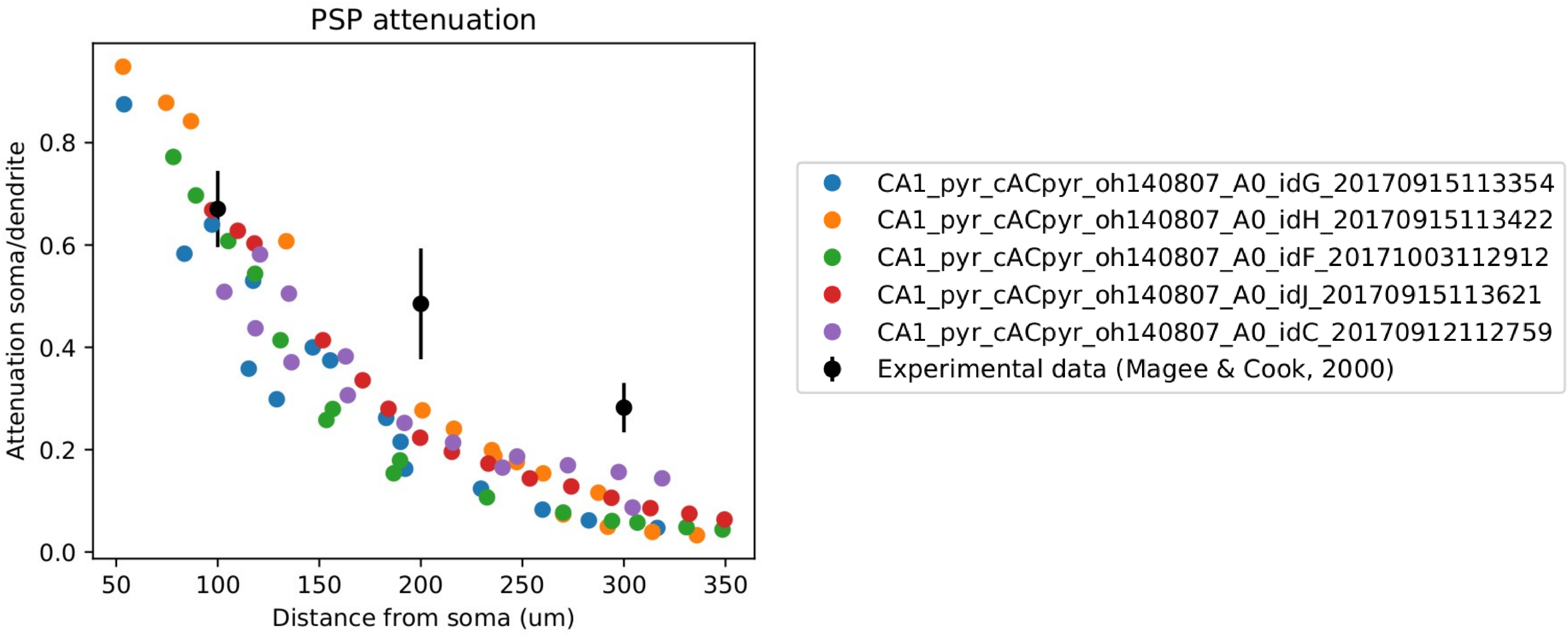
Neuronal Morphology



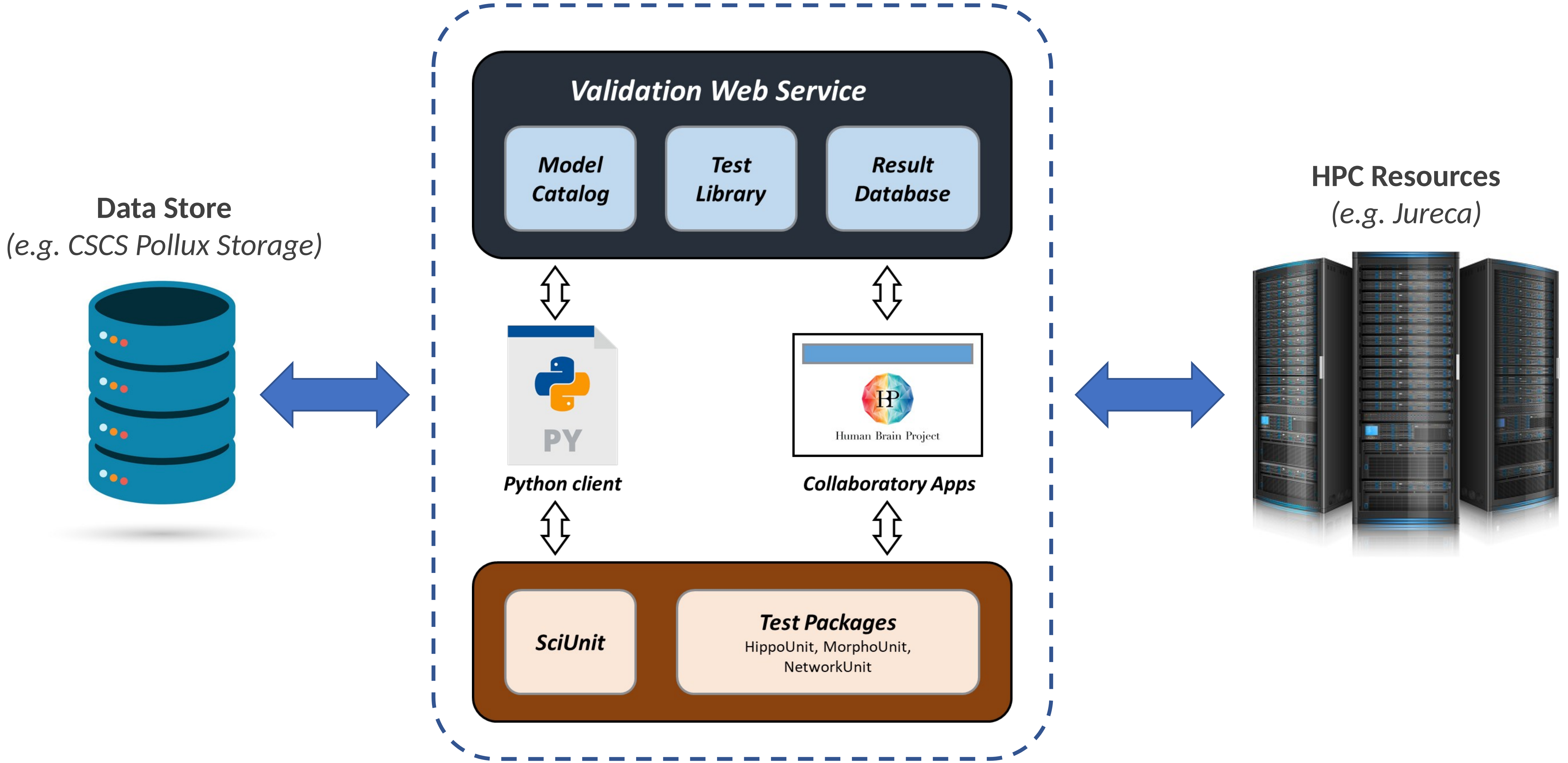
Model Responses



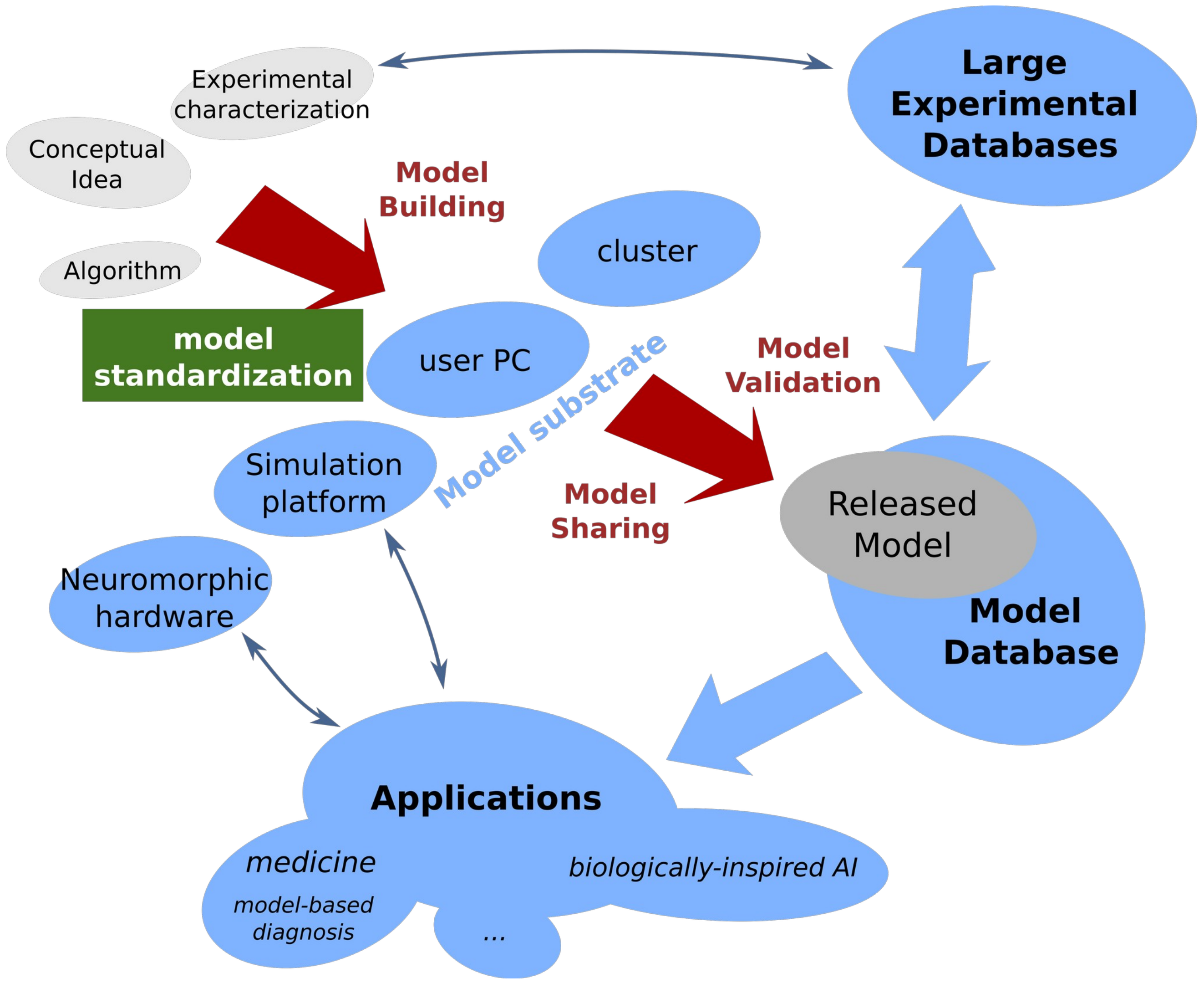
Post-Synaptic Potential (PSP) Attenuation Test



Validation Framework – Various Components

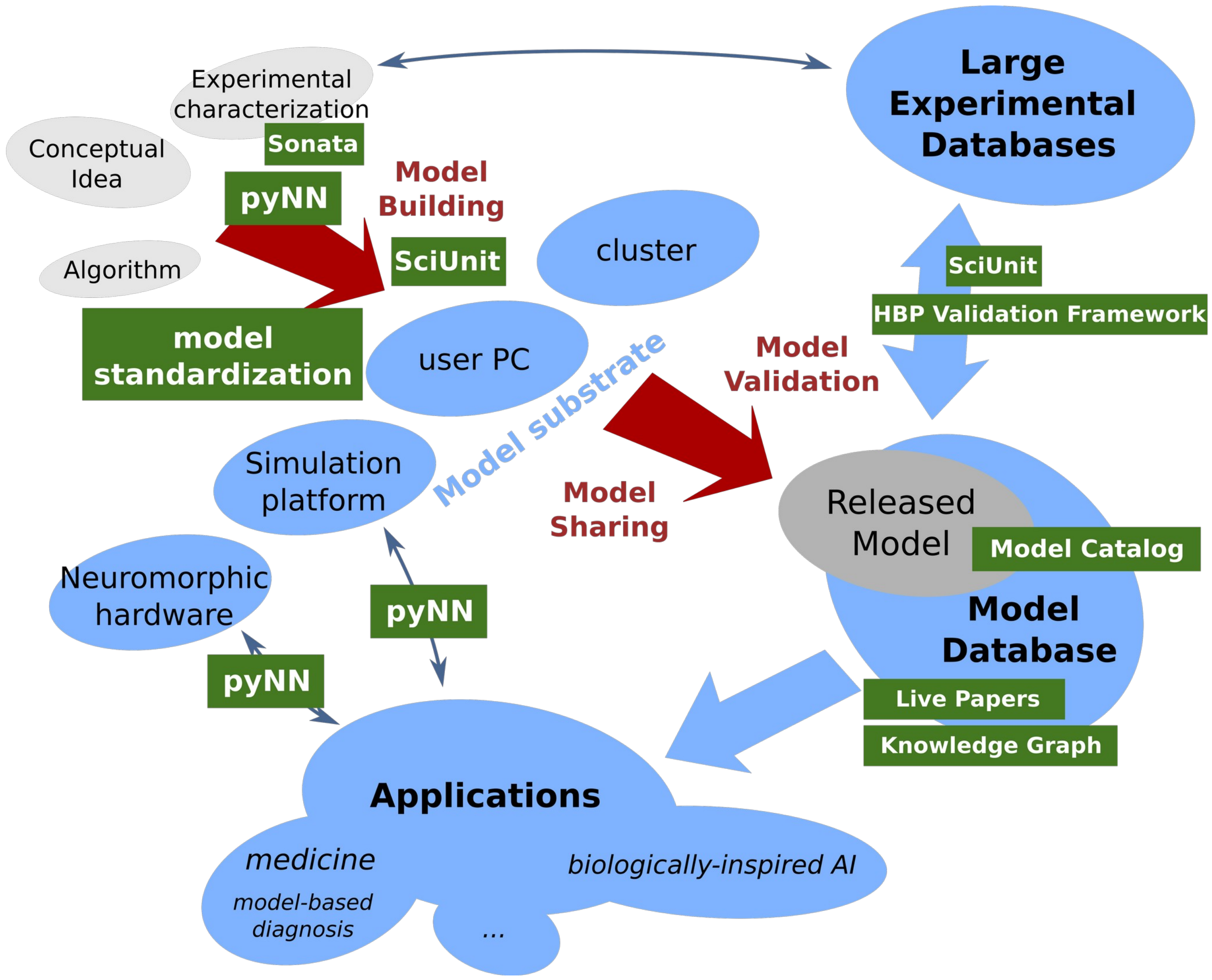


Summary



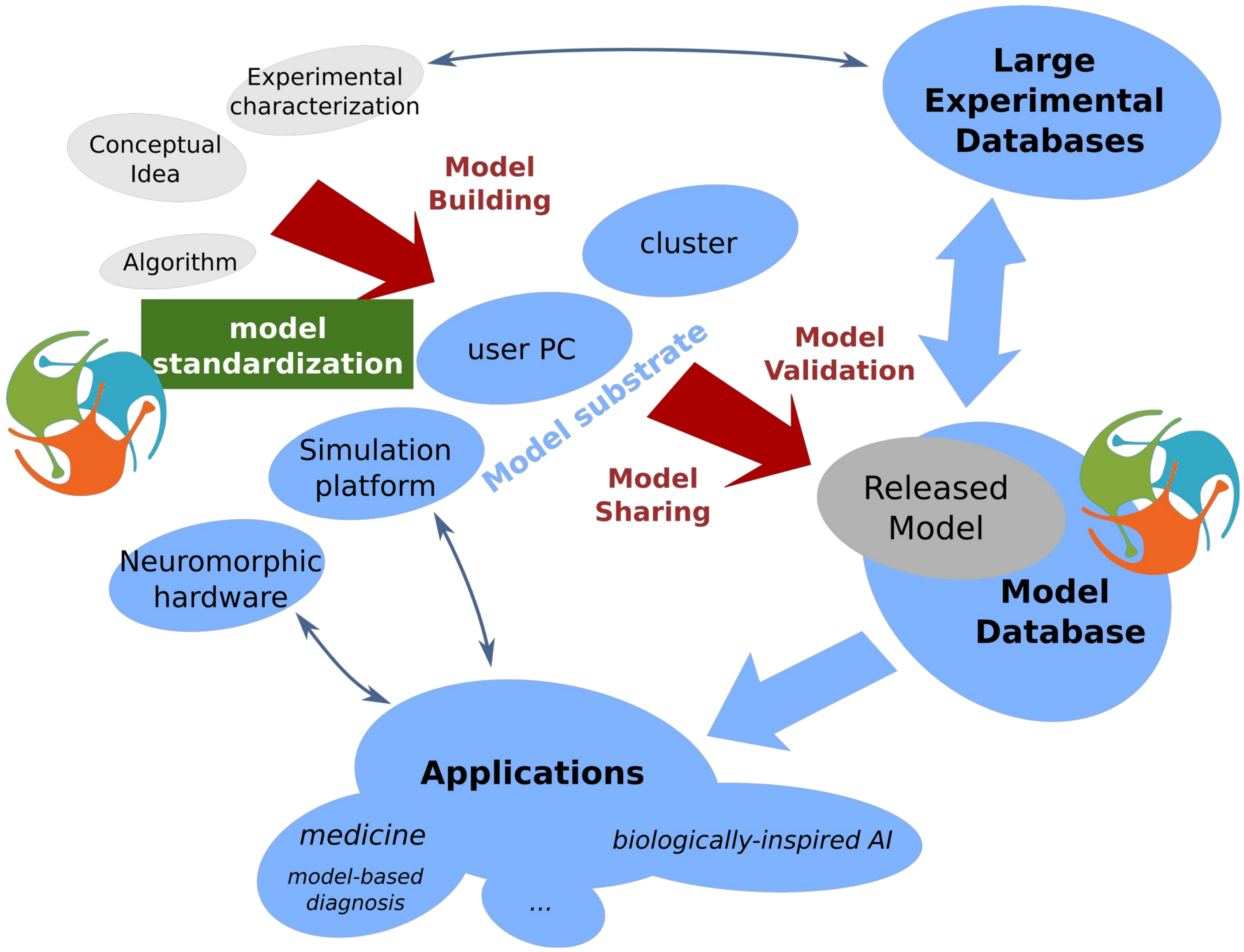
Experimental characterization
Large Experimental Databases
Conceptual Idea
Model Building
Algorithm
cluster
model standardization
user PC
Model substrate
Model Validation
Simulation platform
Model Sharing
Released Model
Neuromorphic hardware
Model Database
Applications
medicine
model-based diagnosis
biologically-inspired AI
...

Summary



Experimental characterization
Conceptual Idea
Sonata
pyNN
Model Building
Algorithm
SciUnit
cluster
model standardization
user PC
Model substrate
Large Experimental Databases
SciUnit
HBP Validation Framework
Model Validation
Simulation platform
Model Sharing
Released Model
Model Catalog
Neuromorphic hardware
pyNN
pyNN
Model Database
Live Papers
Knowledge Graph
Applications
medicine
model-based diagnosis
biologically-inspired AI
...

Summary


Experimental characterization
Conceptual Idea
Algorithm
Model Building
model standardization
cluster
user PC
Model substrate
Simulation platform
Neuromorphic hardware
Large Experimental Databases
Model Validation
Model Sharing
Released Model
Model Database
Applications
medicine
model-based diagnosis
...
biologically-inspired AI

Human Brain Project

UNIC

Thank you

cnrs
dépasser les frontières