Network Modeling at the Allen Institute for Brain Science

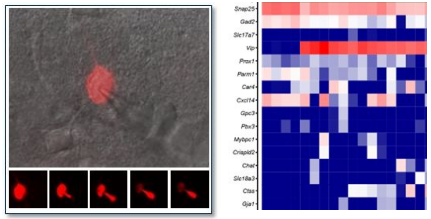
Kael Dai

HBP CodeJam

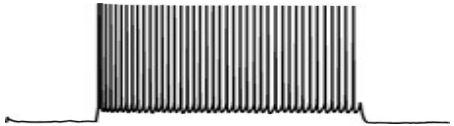
Nov 2018

Single Cell Modeling: Experimental Pipelines

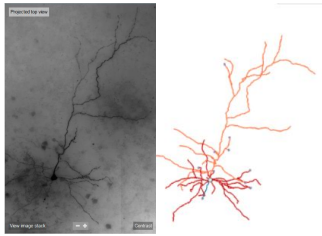
Multiples labs investigating cell types and properties in the mouse and human neocortex.



Transcriptomics (rna-seq)



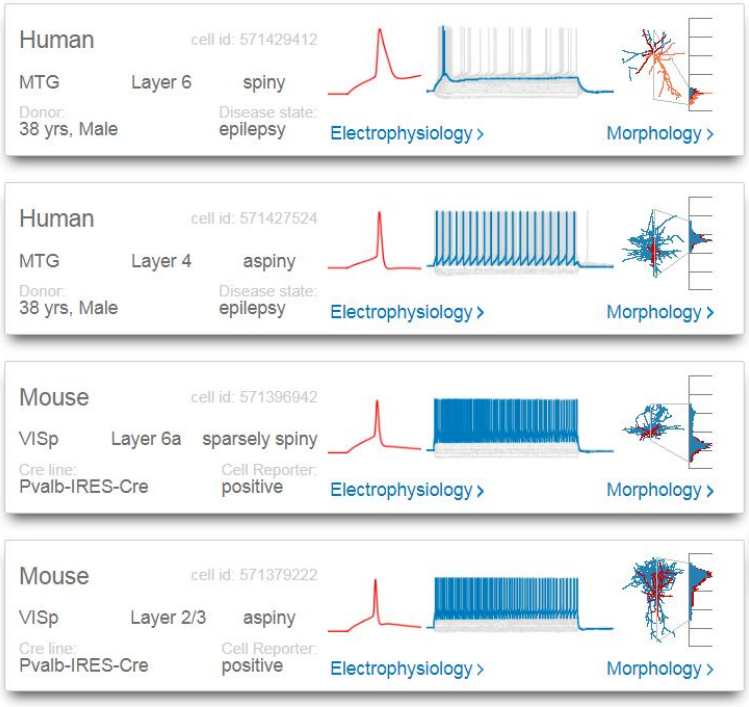
Electrophysiology



Histology and Morphology (Vaa3D)

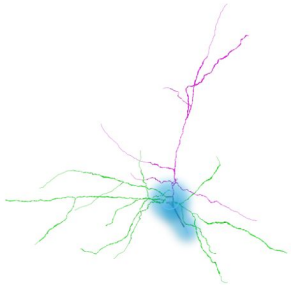
Data of thousands of cell can be downloaded from the Allen Cell Types Database:

http://celltypes.brain-map.org/



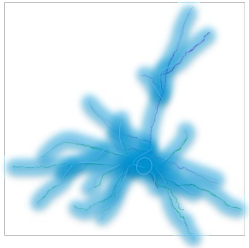
Generalized leaky integrate-and-fire (GLIF 1-5)

Teeter, Nat. Comm 2018



Passive dendritic compartments (perisomatic)

Gouwens, Nat. Comm 2018

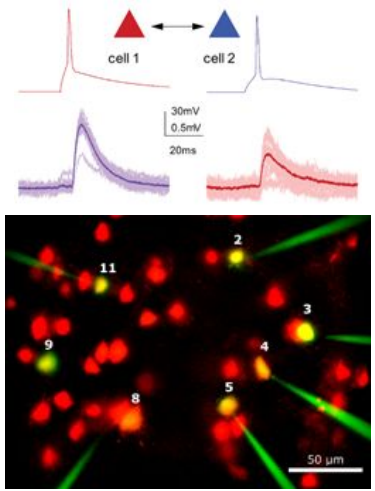


Active dendritic compartments (all-active)

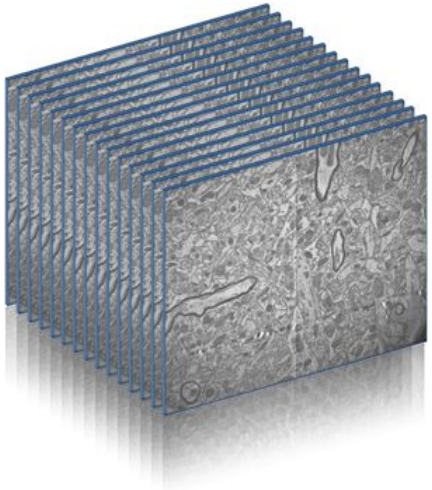
Connectivity

Using synaptic physiology, EM, and viral injections we are able to find patterns of connectivity within and between regions of the mouse visual cortex.

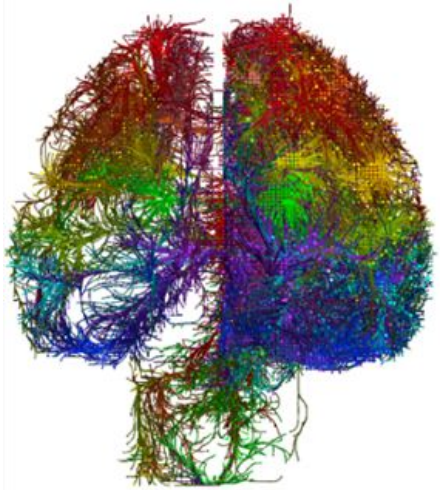
http://connectivity.brain-map.org/



Multi-patch Synaptic physiology



Electron microscope connectomics

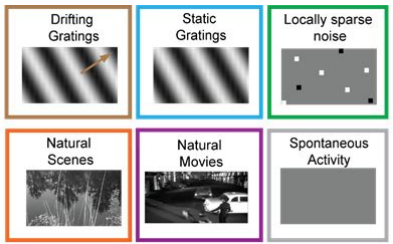


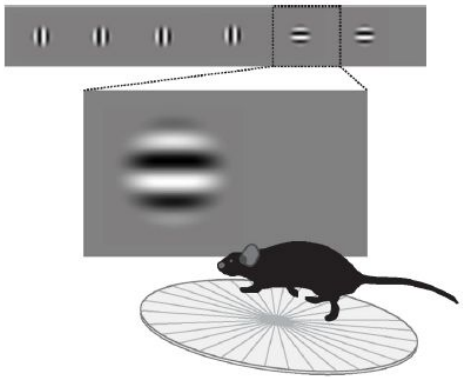
Mesoscale connectomics

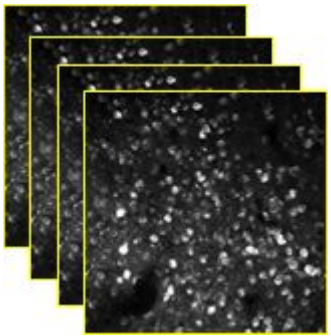
In-Vivo Behavioral Observatories

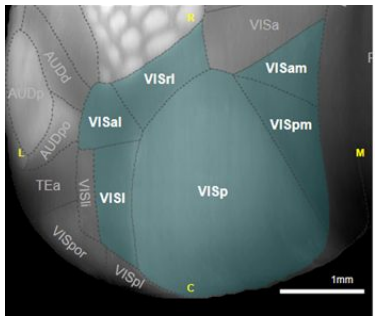
With ophys techniques we are able to record neural activity of the mouse visual cortex during passive viewing or behavioral experiments

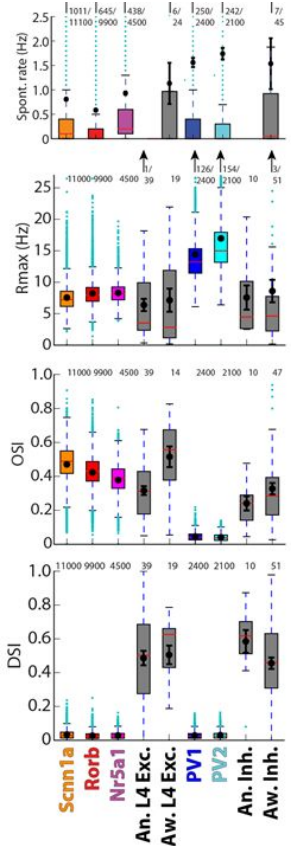
http://observatory.brain-map.org/visualcoding



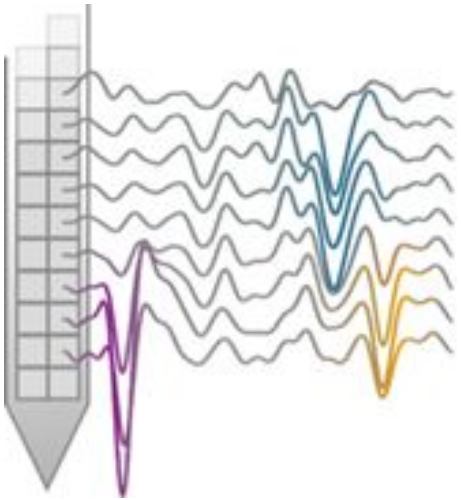








Also developing a similar pipeline with ephys techniques using NeuroPixel probes

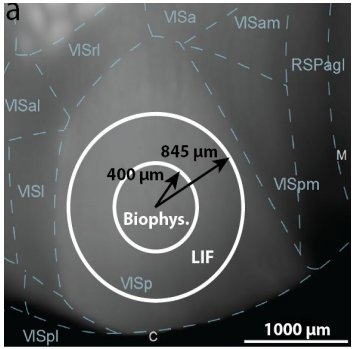


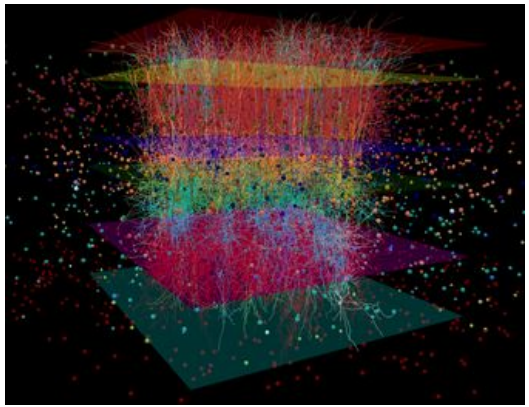
Current Modeling Efforts

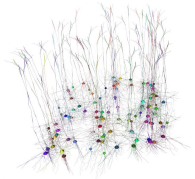
Detailed Models and simulations of a mouse visual V1 column

- >280,000 cells (114+ unique models)
- > 70 million connections
- Feedforward stimuli from the LGN and higher cortical areas

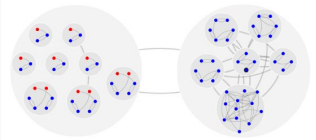
Layer 4 Model:
Arkhipov, et al, Plos Comp. Bio 2018
https://github.com/AllenInstitute/arkhipov2018_layer4/

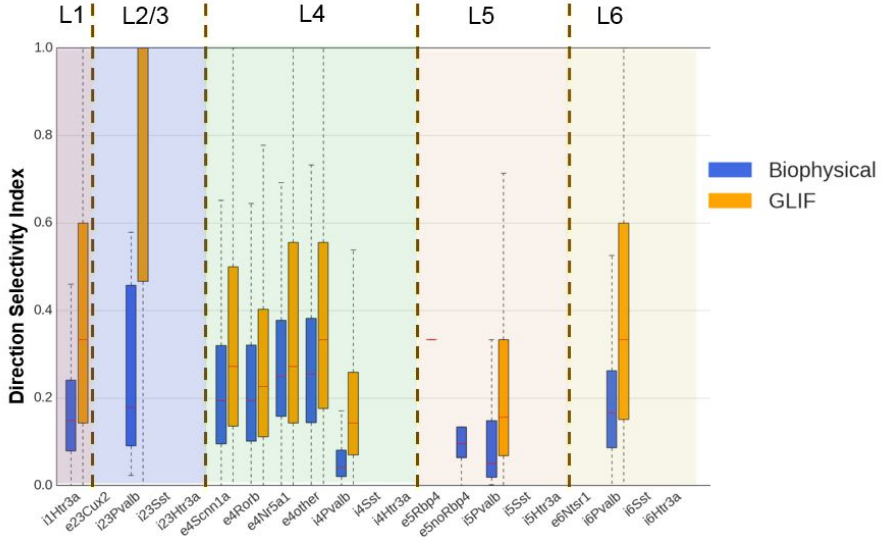






vs.

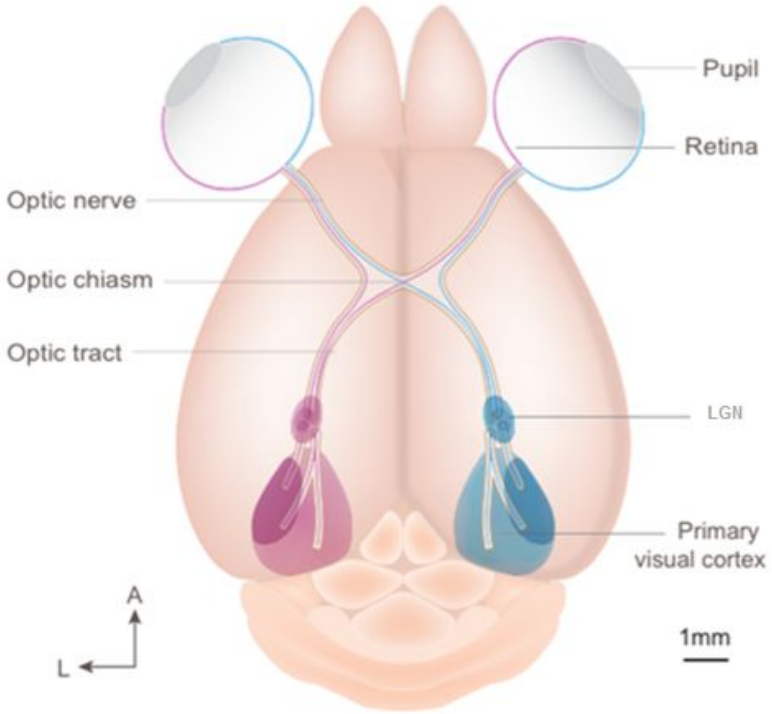


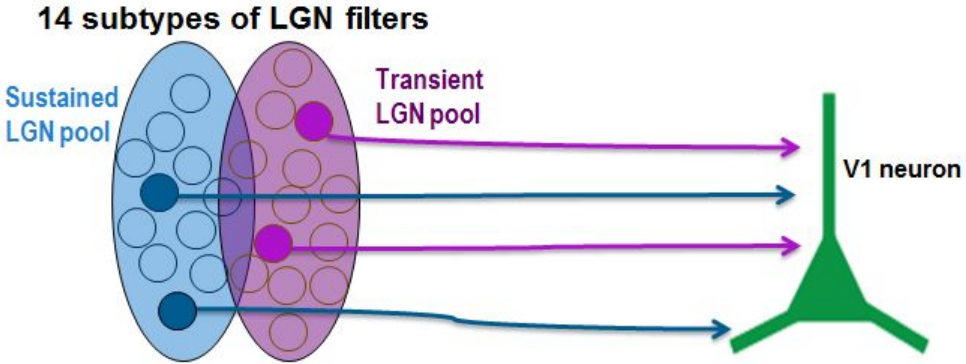


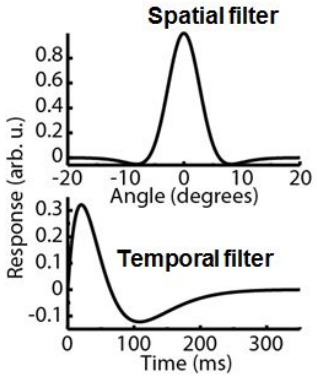
Use both compartmental models (with NEURON) and point neuron models (with NEST) - explore the difference between various levels of resolutions

Current Modeling Efforts

Modeling visual stimuli onto the LGN.

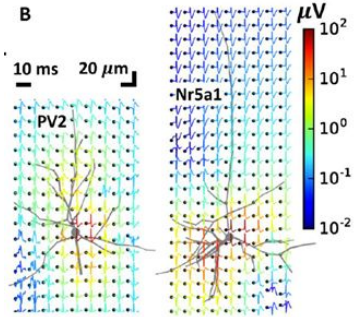




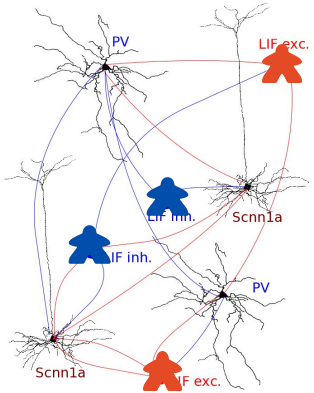


Modeling Projects

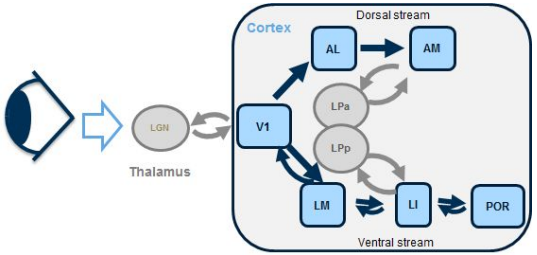
Extracellular stimulations and **LFP** recordings from small networks



Network models using **Human cell** models



Predictive encoding models



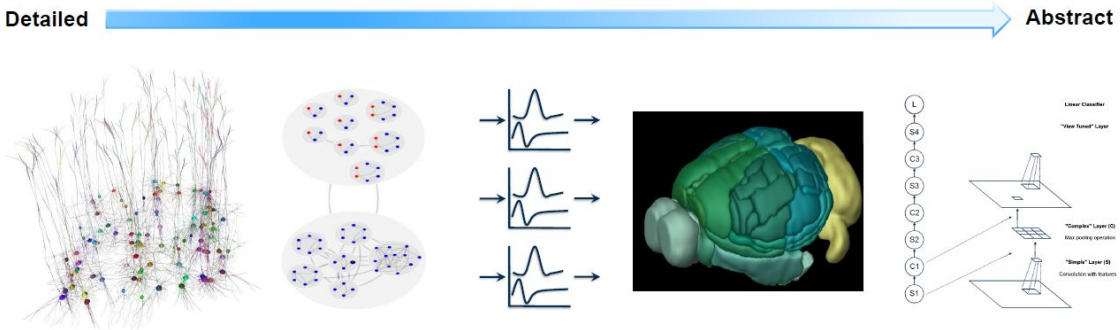
STP synaptic models in multicompartmental and point networks

Epileptic networks

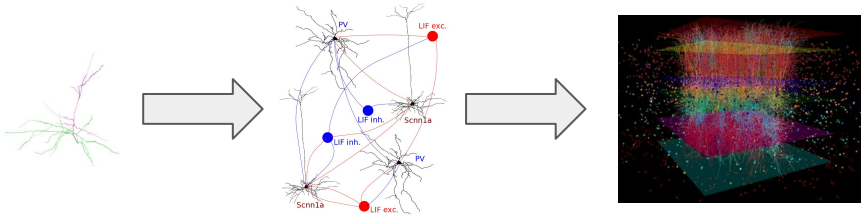
Single Platform to Consolidate our Modeling Projects

Needed a software platform that:

- Work across different levels of resolution (and across different simulators)



- Scalable from single-cell simulations to large dense networks.



- Easy to modify and update as our data changes.

- Allows for sharing of models and simulation results (both internally and externally).

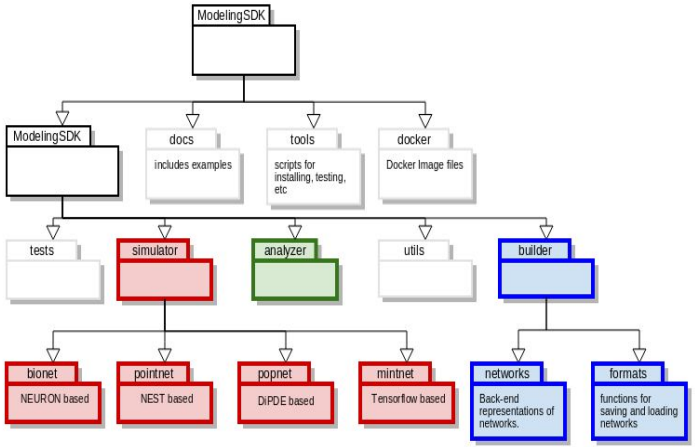
Solution: The Brain Modeling Toolkit (BMTK)

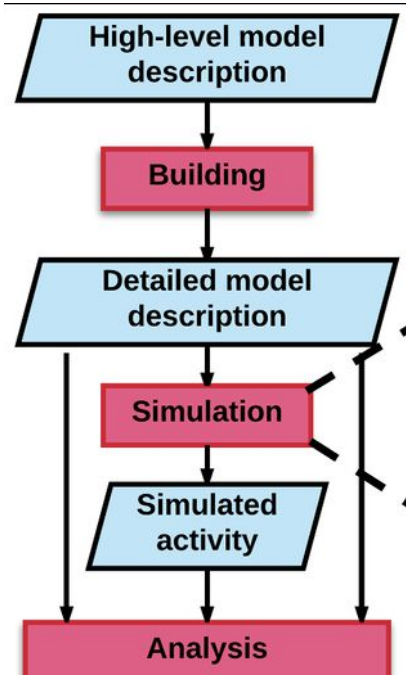
https://github.com/AllenInstitute/bmtk

A software package for building, simulating and analyzing large-scale network models at different resolutions.

Open-Source (BSD)

Support Python 2.7 and 3.5+





BMTK: Network Builder

A Python API for building large-scale networks.

- Can build different resolutions of brain networks (biophysical, point, ANN) using a graph structure.
- Modelers can choose whatever parameters they require for representing nodes, can create custom connectivity rules.
- Can be used as a stand-alone tool (i.e. does not require NEURON, NEST, etc.)

```python
import numpy as np
from bmtk.builder import SynNetwork

net = SynNetwork('V1')
net.add_nodes(N=10500, name='Scnn1a',
              tuning_angle=tuning_angles_1,
              rotatons=np.random.uinform(10500, 3),
              positions=scnn1a_pos(),
              model_type='Biophysical',
              model_params='472363762_fit.json',
              model_morphology='Scnn1a_473845048_m.swc',
              potential='e')

net.add_nodes(N=800, name='PV1',
              positions=pvalb_pos(),
              model_type='InfFire1',
              model_params='IntFire1_inh_1.json',
              potential='i')


net.add_edges(targets={'name': 'Scnn1a'}, sources={'potential': 'i'},
              func=synaptic_filter,
              delay=2.0,
              min_weight=-4.5e-05,
              weight_function='gaussian',
              target_regions=['soma', 'basal'],
              template='Exp2Syn')

net.build()
net.save_nodes('v1_nodes.h5', 'v1_nodes_metadata.csv')
net.save_edges('v1_edges.h5', 'v1_edges_metadata.csv')
```

BMTK: Simulator Engines

An interface for running large-scale network models across different simulators of different levels-of-resolutions.

Initialize and run a simulation with little-to-no programming needed (using SONATA config files).

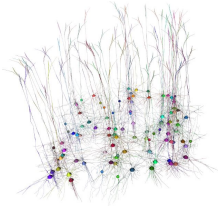
Built-in parallel computing support

Formats and standardizes the output files.

*Bio*Net

A NEURON based Interface for running multicompartment models

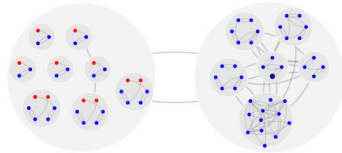
NEURON
for empirically-based simulations



*Point*Net

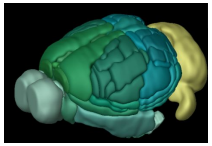
A NEST based Interface for running point-process models

nest ::



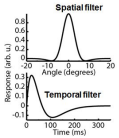
*Pop*Net

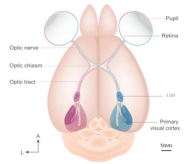
Uses The DiPDE simulator to model firing rate dynamics of neuronal populations



*Filter*Net

Uses filters to convert stimuli to firing rates and spikes on a receptive field





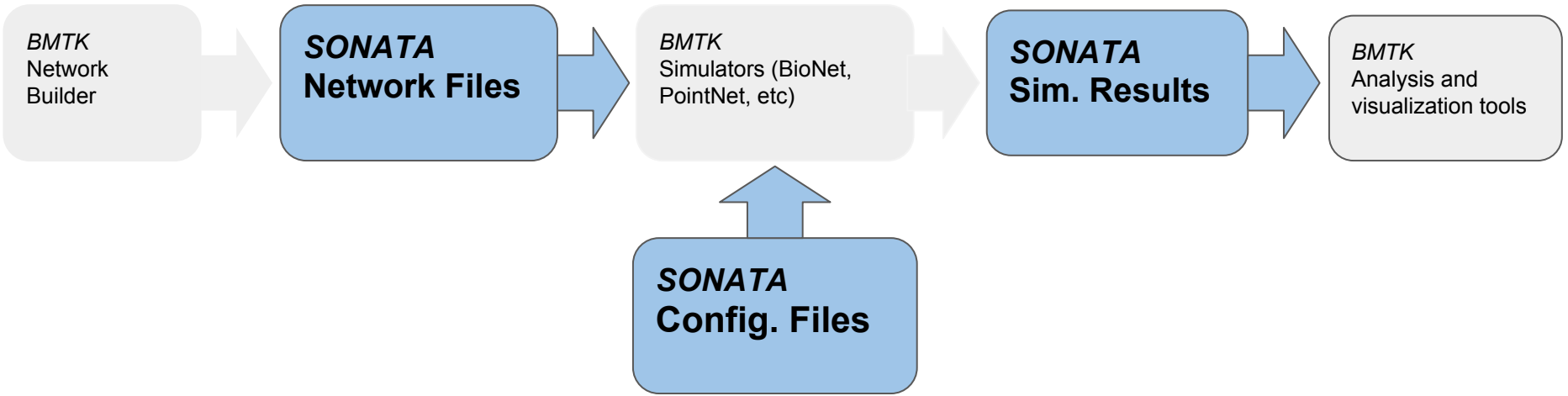
Adding more in 2019

SONATA Data Format

A set of standardized format for representing brain circuit models and simulation. Includes formats for

- Network Representation
- Simulation configuration
- Simulation results

SONATA integration into the BMTK



Representations of Nodes and Edges

Uses HDF5 and CSV files to store nodes (cells) and edges (synapses, junctions)

Highly normalized and optimized table structure.

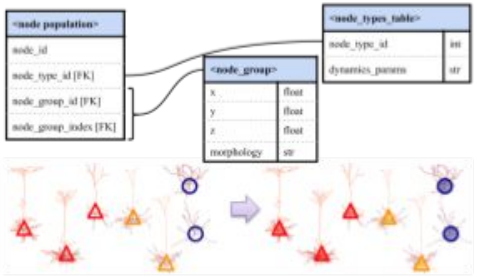
Saves both individual node properties and shared node “types” properties (same with edges).

Allows for heterogeneous networks with a mixture of different model types

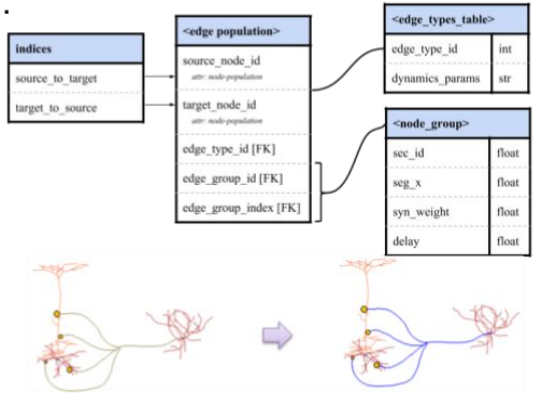
Includes reserved dictionary of attributes for cross-software compatibility.

But also allows modelers to add and remove attributes as required by their models.

Reserved node attributes	
Attribute	**Description**
model_type	Used to describe the type of model for each node. Currently there are four valid values; *biophysical* for morphologically detailed models, *single_compartment* for non-morphological models, *point_process* for point neuron models, and *virtual* for non-simulated spikes or rate generating objects
model_template	The name of the template file or built-in model used to instantiate the object.
model_processing	A directive or function name providing additional instructions on how to processes each individual model (for example how to handle axon).
dynamics_params	A reference to a file or hdf5 group with software specific parameter values used to instantiate a given model.
morphology	Name of file or description used to build a biophysical model_type node
x, y, z	Coordinates of the node (for biophysical, point_soma and point_process models).
rotation_angle_x, _y, _z	Rotation of cell in space (for biophysical type models)



nodes



edges (with indexing)

Simulation Configuration Files

A JSON-based text file to define all the parameters needed to run a full network simulation.

Including

- Run-time parameters
  simulation time, step time, initial conditions
- Network and model parameters
  What nodes/edges files are being simulated, mod/hoc/NeuroML/swc files to build cells and synapses
- Network stimuli
  current clamp, spike trains, voltage clamps
- Recorded parameters
  spike-trains, membrane voltage/Ca++, LFP

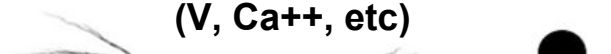
sonata_sim_config.json

```
{
  "run": {
    "tstop": 3000.0,
    "dt": 0.1,
    "dL": 20.0,
    "spike_threshold": -15,
    "nsteps_block": 5000
  },
  "conditions": {
    "celsius": 34.0,
    "v_init": -80
  },

  "output":{"log_file": "$OUTPUT_DIR/log.txt"...},

  "inputs": {
    "spikes": {
      "input_type": "spikes",
      "module": "h5",
      "input_file": "inputs/exc_spike_trains.h5",
      "node_set": "biophysical"
    },

    "current_clamp": {
      "input_type": "current_clamp",
      "module": "IClamp",
      "node_set": "biophysical",
      "amp": 0.2200,
      "delay": 500.0,
      "duration": 1000.0
    }
```

Simulation Output

Standards for storing network simulation results into HDF5 based, optimized tables.

Single and multi-compartment traces (V, Ca++, etc)

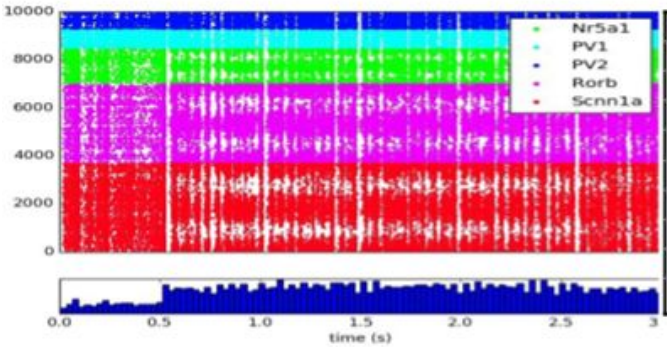


	gid 0						gid 1	gid 2
0	-80.123	-80.123	-80.022	-80.001	-79.124	....	0.000	
1	-80.023	-80.123	-79.322	-80.000	-81.294	....	0.012	
2	-80.011	-80.110	-79.211	-79.052	-79.615	....	0.019	
3	-79.950	-80.070	-78.999	-78.799	-77.118	....	0.019	
4	-78.550	-80.001	-77.999	-77.999	-77.118	....	0.021	
5	-78.450	-79.070	-77.819	-77.559	-77.118	....	0.021	
6	-75.150	-80.070	-78.999	-79.991	-77.118	....	0.022	

Time

voltage traces per cell/compartments

Spike Trains



gid	spikes (ms)
0	1.02
0	11.10
0	22.98
0	28.01
0	....
1	9.12
1	15.22
1	19.98

Extracellular voltage potentials

https://www.nature.com/articles/nature24636#f1

	0	1	2	4	5	6
0	-1.001	-3.140	-3.177	-3.511	-3.212	....
1	-1.022	-4.709	-4.976	-5.267	-4.677	
2	-2.550	-5.485	-5.796	-6.135	-7.125	
3	-3.001	-5.855	-6.188	-6.551	-7.252	
4	-5.548	-6.011	-6.353	-6.726	-7.300	
5	-5.730	-6.047	-6.399	-6.766	-9.598	
6	....					

Time

extracellular voltage per channel

SONATA: Growing Support

https://github.com/AllenInstitute/sonata

SONATA is an open-standard with the goal of increasing interoperability and reproducibility in the modeling and simulation community, and continually looking for feedback for ways to improve the standard in the future.

Other software Implementing SONATA support

RTNeuron https://github.com/BlueBrain/RTNeuron

Brion https://github.com/BlueBrain/Brion

PyNN http://neuralensemble.org/PyNN/

NeuroML https://www.neuroml.org/

NetPyNE https://github.com/Neurosim-lab/netpyne

Acknowledgements

BMTK

Anton Arkhipov
Sergey Graity
Yazan Billeh
David Feng
Ram Iyer
Sarah Naylor
Michael Buice
Stefan Mihalas
Nicholas Cain
Jung Hoon Lee
Nathan Gouwens
Michael Hines
Fahimeh Baghal
Yina Wei

SONATA

Yazan N. Billeh
Jean-Denis Courcol
Sergey L. Graiy
Juan Hernando
Adrian Devresse
Mike Gevaert
James King
Daniel Nachbauer
Arseniy Povolotskiy
Werner Van Geit
Anton Arkhipov
Eilif Muller
Andrew Davison
Padraig Gleeson
Salvador Dura

PAUL G. ALLEN
1953 - 2018

THANK YOU
We wish to thank the Allen Institute founder, Paul G. Allen, for his vision, encouragement and support.

We honor his legacy today, and every day into the long future of the Allen Institute, by carrying out our mission of tackling the hard problems in bioscience and making a significant difference in our respective fields.

alleninstitute.org
brain-map.org

ALLEN INSTITUTE for BRAIN SCIENCE