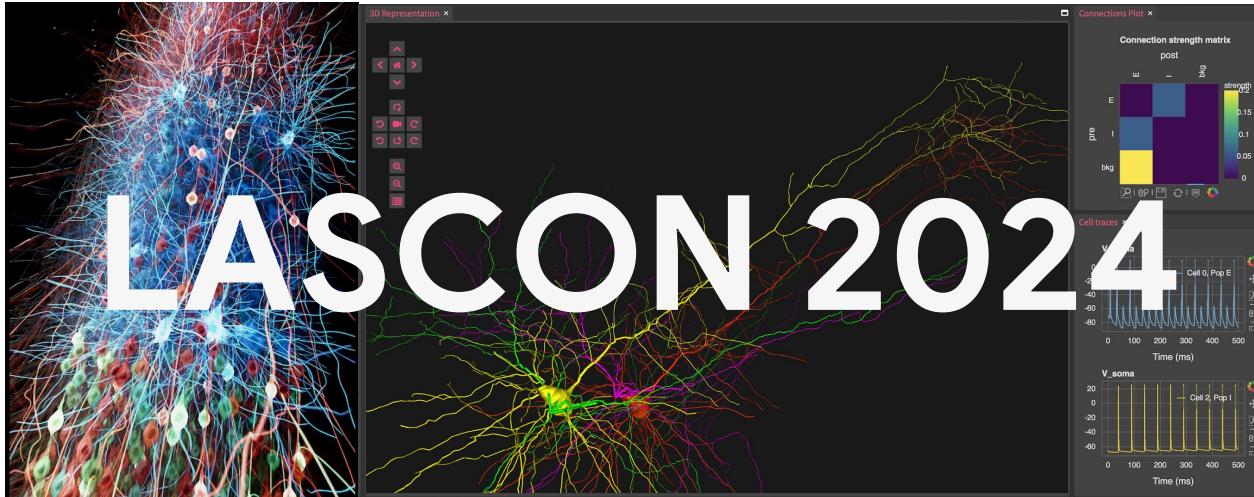


# Network Modelling with

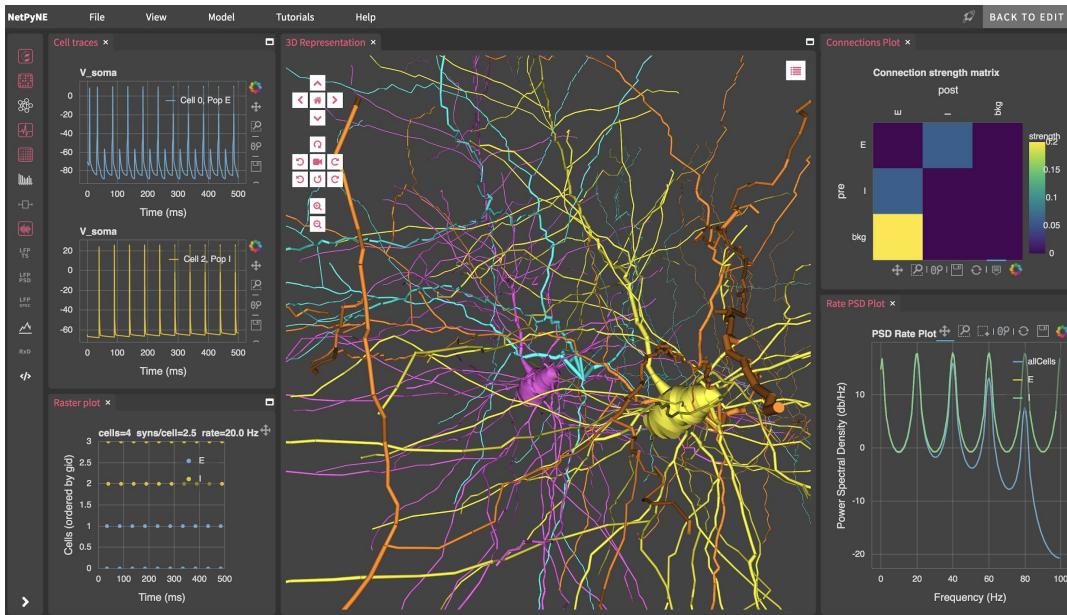


Valery Bragin, State University of New York (SUNY) Downstate [vbragin19@gmail.com](mailto:vbragin19@gmail.com)  
PI: Salvador Dura-Bernal, PhD  
Assistant Professor, State University of New York (SUNY) Downstate

Lab web: [dura-bernal.org](http://dura-bernal.org)



A python package to facilitate the development, parallel simulation, optimization and analysis of biological neuronal networks using the NEURON simulator.



Funded by:



National Institutes  
of Health

# Motivation

Separate model parameters from standardized implementation

Standardize format – easy to read, interpret, edit, share, reproduce, etc

```
popParams['EXC_L2'] = {  
    'cellType': 'PYR',  
    'yRange': [100, 400],  
    'numCells': 50}
```



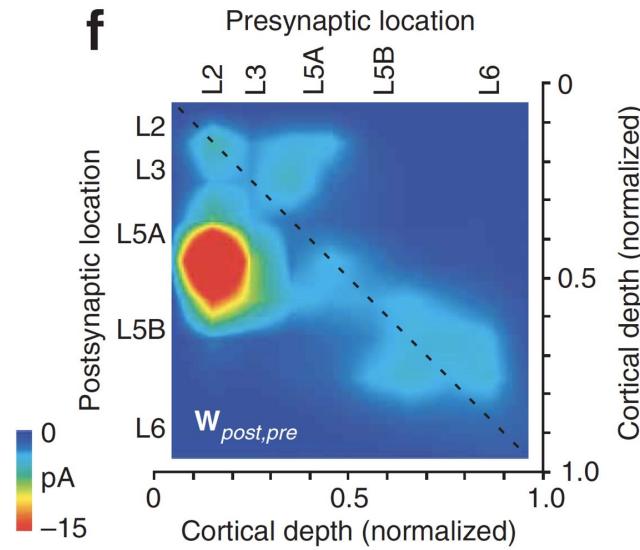
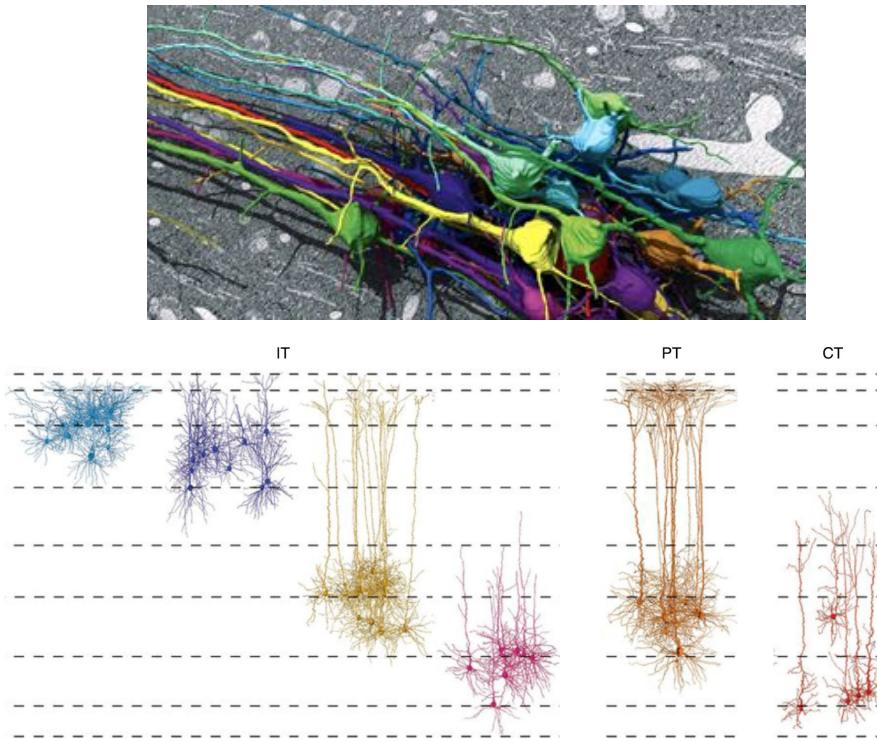
```
for gid in range(pop.numCells):  
    cell = sim.Cell()  
    cell.y = numpy.random(100,400)  
    cell.type = 'PYR'  
    pc.cell(gid, h.NetCon(v_soma, threshold))
```

Standard, efficient and tested  
backend implementation

Potentially very different implementations  
(arbitrary functions, variables, file names  
etc.)

# Motivation

Facilitate incorporation of experimental data at multiple scales



# NetPyNE High-Level Specifications

NEURON

```
# add exc connection
postSyn1 = h.ExpSyn(postCell.dend(0.5))
postSyn1.tau = 2
postSyn1.e = -90

pre1Con = h.NetCon(preCell1.soma(0.5)._ref_v,
                   postSyn1,
                   sec=preCell1.soma)
pre1Con.delay = 1
pre1Con.weight[0] = 0.001
pre1Con.threshold = 0
```

# NetPyNE High-Level Specifications

NetPyNE

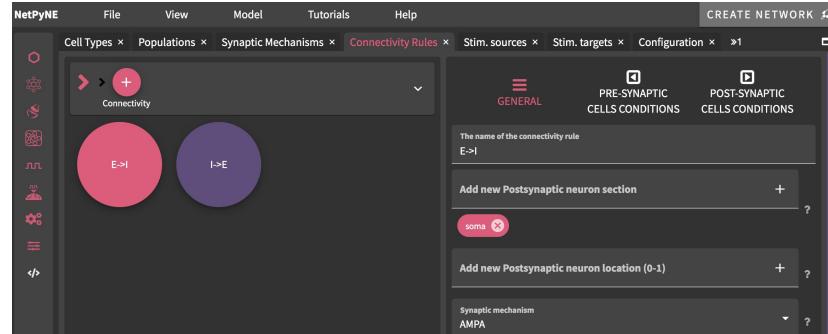
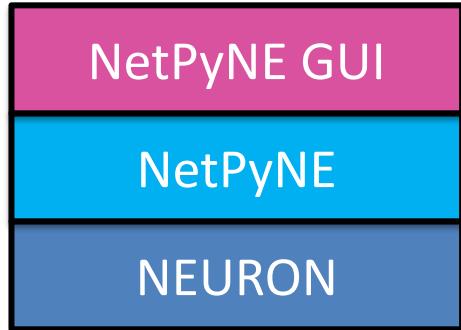
NEURON

```
## Cell connectivity rules
netParams.connParams['S->M'] = {
    'preConds': {'pop': 'S'},
    'postConds': {'pop': 'M'},
    'probability': 0.5,
    'weight': 0.01,
    'delay': 5,
    'synMech': 'exc'}
```

```
# add exc connection
postSyn1 = h.ExpSyn(postCell.dend(0.5))
postSyn1.tau = 2
postSyn1.e = -90

pre1Con = h.NetCon(preCell1.soma(0.5)._ref_v,
                   postSyn1,
                   sec=preCell1.soma)
pre1Con.delay = 1
pre1Con.weight[0] = 0.001
pre1Con.threshold = 0
```

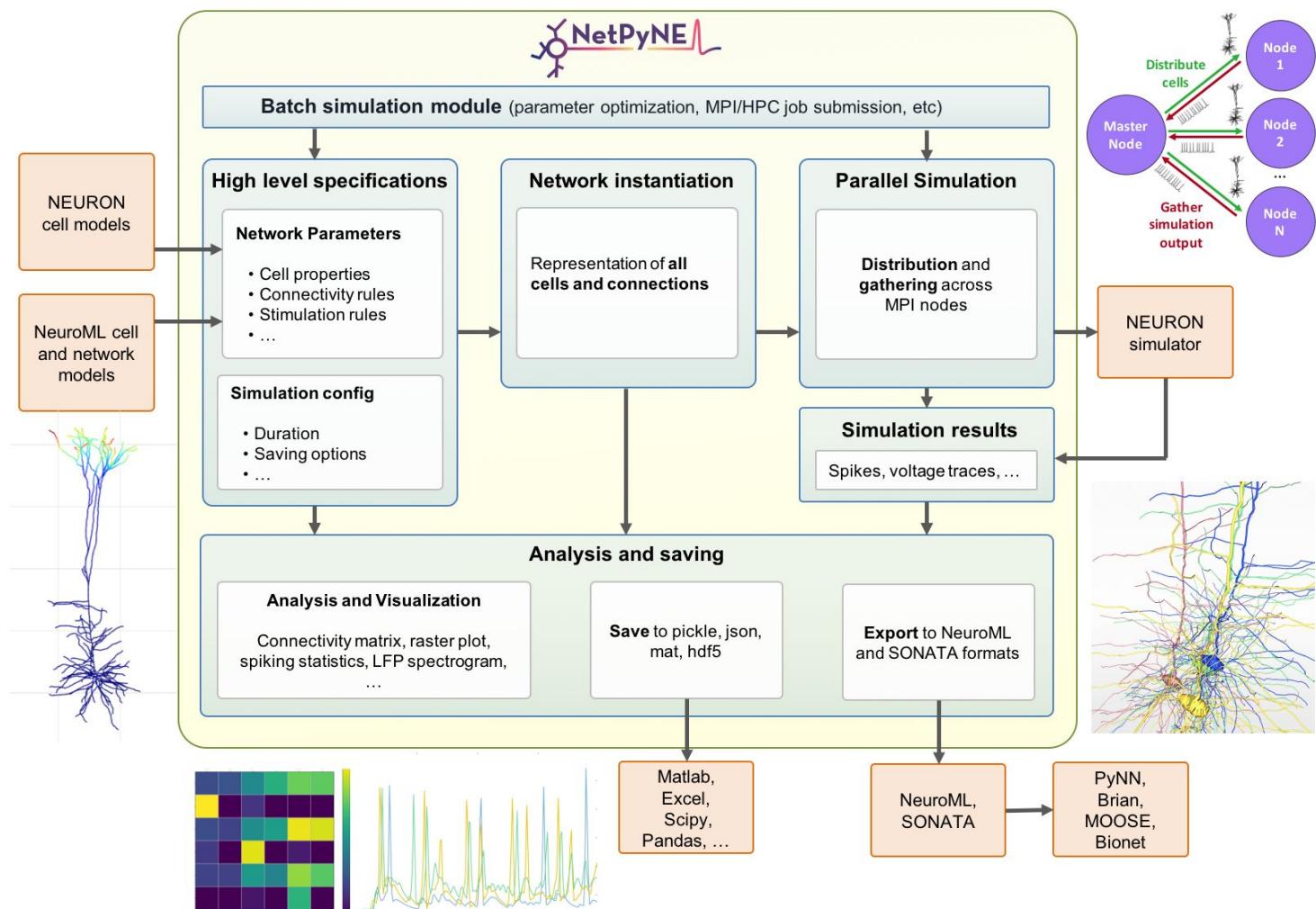
# NetPyNE High-Level Specifications



```
## Cell connectivity rules
netParams.connParams['S->M'] = {
    'preConds': {'pop': 'S'},
    'postConds': {'pop': 'M'},
    'probability': 0.5,
    'weight': 0.01,
    'delay': 5,
    'synMech': 'exc'}
```

```
# add exc connection
postSyn1 = h.ExpSyn(postCell.dend(0.5))
postSyn1.tau = 2
postSyn1.e = -90

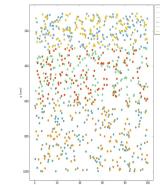
pre1Con = h.NetCon(preCell1.soma(0.5)._ref_v,
                   postSyn1,
                   sec=preCell1.soma)
pre1Con.delay = 1
pre1Con.weight[0] = 0.001
pre1Con.threshold = 0
```



# High level specifications

A **standardized, declarative** Python format (JSON-like, lists and dicts) to define:

**Populations:** cell type, number of neurons or density, spatial extent, ...



**Cell properties:** Morphology, biophysics, molecular processes ...



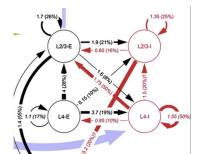
**Synaptic mechanisms:** Time constants, reversal potential, ...



**Stimulation:** Spike generators, current clamps, spatiotemporal patterns, ...



**Connectivity rules:** conditions of pre- and post-synaptic cells, functions, ...



**Simulation configuration:** duration, saving and analysis, visualization, ...



# High level specifications

i) `popParams['EXC_L2'] = {  
 'cellType': 'PYR',  
 'cellModel': 'simple',  
 'yRange': [100, 400],  
 'numCells': 50}`

ii) `popParams['EXC_L5'] = {  
 'cellType': 'PYR',  
 'cellModel': 'complex',  
 'yRange': [700, 1000],  
 'density': 80e3}`

iii) `cellParams['PYR_simple'] = {  
 'conds': {'cellType': 'PYR',  
 'cellModel': 'simple'},  
 'secs': {'soma':  
 'geom': {'diam': 18, 'L': 18},  
 'mechs': {'hh':  
 {'gnabar': 0.12,  
 'gkbar': 0.036,  
 'gl': 0.003,  
 'el': -70}}}}`

iv) `importCellParams(  
 label = 'PYR_complex',  
 conds = {'cellType': 'PYR',  
 'cellModel': 'complex'},  
 fileName = 'L5_pyr_full.hoc',  
 cellName = 'PYR_L5')`

v) `rxParams['regions'] = {  
 'cyt': {'cells': 'all',  
 'secs': 'soma',  
 'nrn_region': 'i'}}  
  
rxParams['species'] = {  
 'ca': {'regions': 'cyt',  
 'charge': 2, 'initial': 1e-4},  
 'buf': {'regions': 'cyt', 'initial': 1e-4},  
 'cabuf': {'regions': 'cyt', 'initial': 0}}  
  
rxParams['reactions'] = {  
 'buffering': {'reactant': '2*ca+buf',  
 'product': 'cabuf',  
 'rate_f': 1e6, 'rate_b': 1e-2}}  
  
rxParams['rates'] = {  
 'degradation': {'species': 'buf',  
 'rate': '-1e-3*buf'}}`

vi) `synMechParams['AMPA'] = {  
 'mod': 'Exp2Syn',  
 'taul': 0.8, 'tau2': 5.3, 'e': 0}`

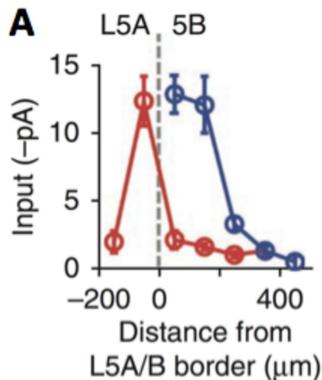
vii) `connParams['L2->E2'] = {  
 'preConds': {'y': [100, 400]},  
 'postConds': {'pop': 'EXC_L2'},  
 'probability': '1*exp(-dist_3D/200)',  
 'weight': 0.4, 'delay': 5, 'synMech': 'AMPA'}`

viii) `connParams['E2->L5'] = {  
 'preConds': {'pop': 'EXC_L2'},  
 'postConds': {'y': [700, 1100], 'cellModel': 'complex'},  
 'convergence': 25, 'weight': '0.001 * post_ynorm',  
 'delay': 'dist_3D/propVelocity', 'sec': 'allDend',  
 'synMech': 'AMPA', 'synsPerConn': 3}`

# High level specifications

## Connectivity

- **Flexible connectivity rules** based on pre- and post-synaptic cell properties (eg, type or location).
- Connectivity **functions** available: probabilistic, convergent, divergent, custom, ...
- Parameters (eg, probability, weight, delay) as a **function of pre/post-synaptic spatial properties**, eg, delays or probability that depend on distance between cells or cortical depth.
- Easily add synapses with **learning** mechanisms (STDP and RL) and **gap junctions**.

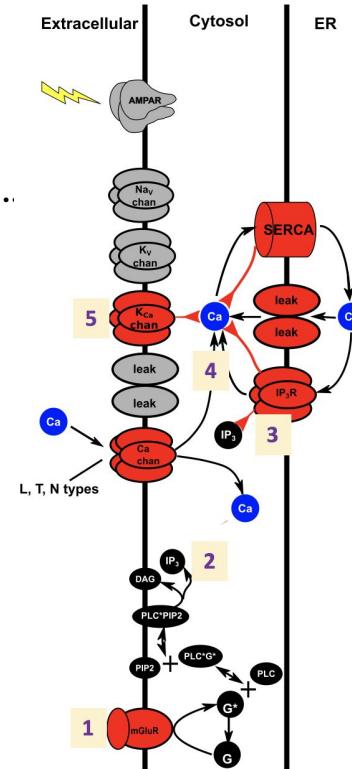
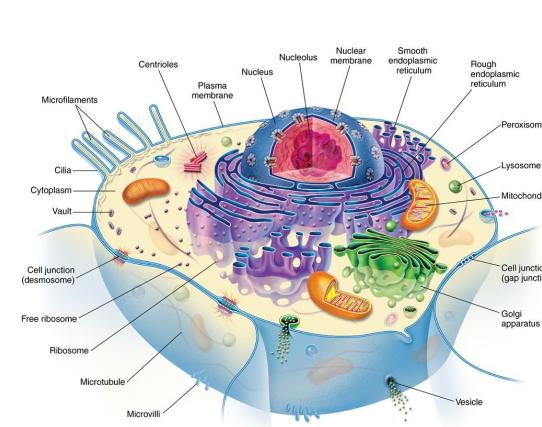
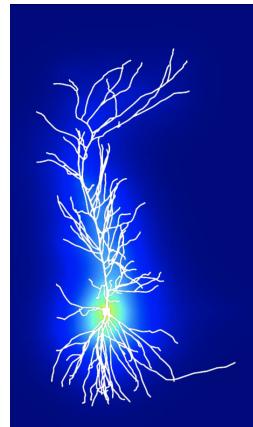
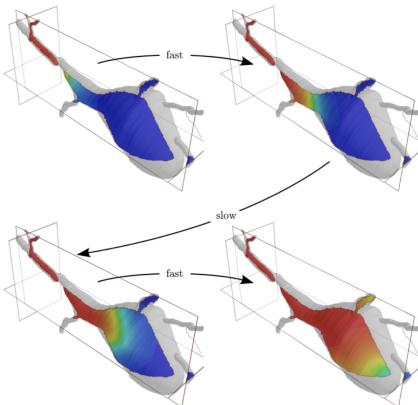


NetPyNE facilitates  
building models  
based on  
experimental data

# High level specifications

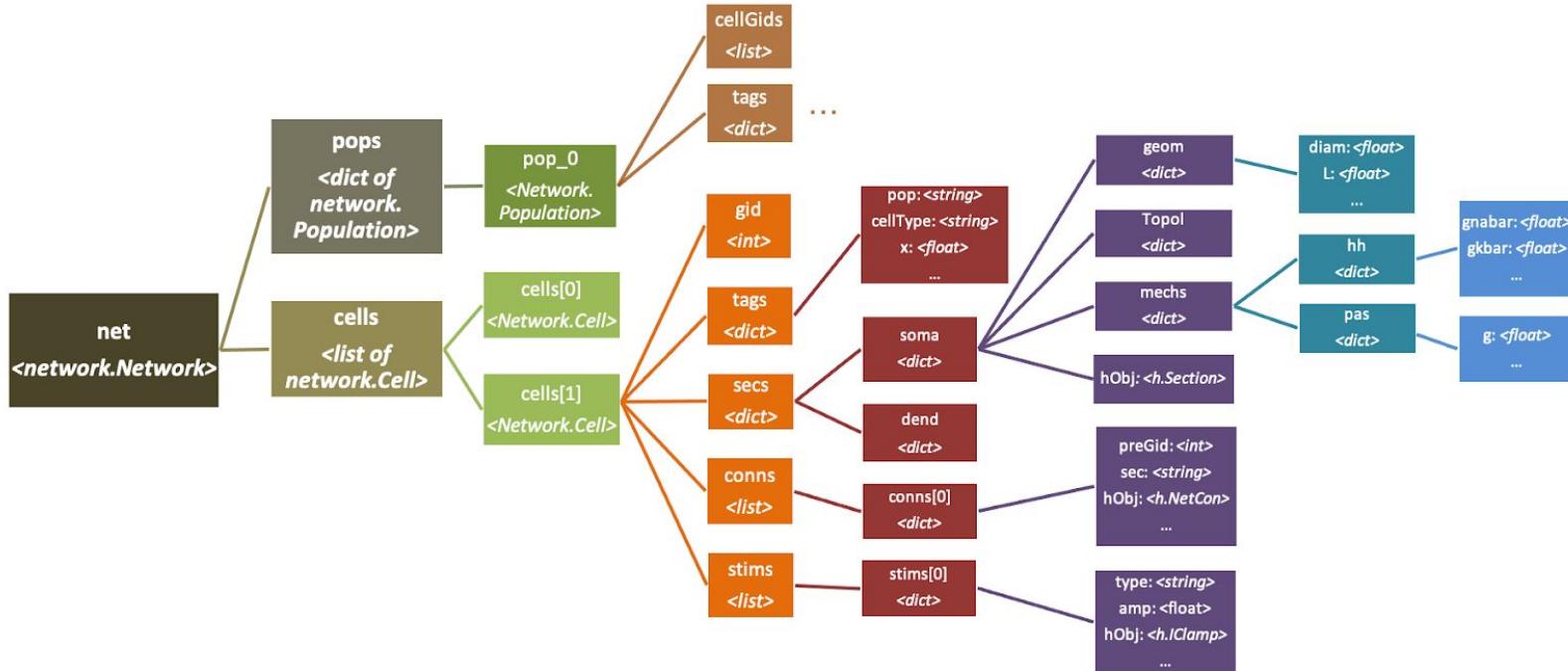
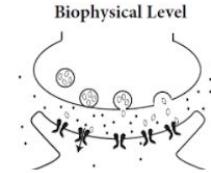
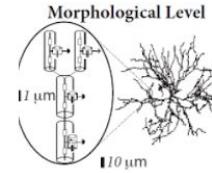
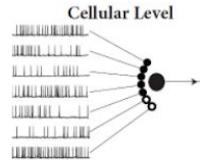
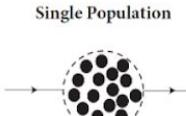
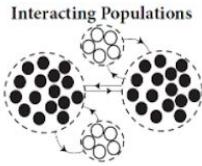
## Molecular reaction-diffusion (RxD)

- Intra- and extracellular **diffusion** of ions, proteins (eg, calcium, potassium, IP<sub>3</sub>, ...)
- Cell internal structures/**organelles** (eg, endoplasmic reticulum, mitochondria,...)
- Molecular **processes** (eg, phosphorylation, buffering, 2nd messenger cascades,...)
- **Interaction** with cell and network scales (eg, firing, plasticity, ...)



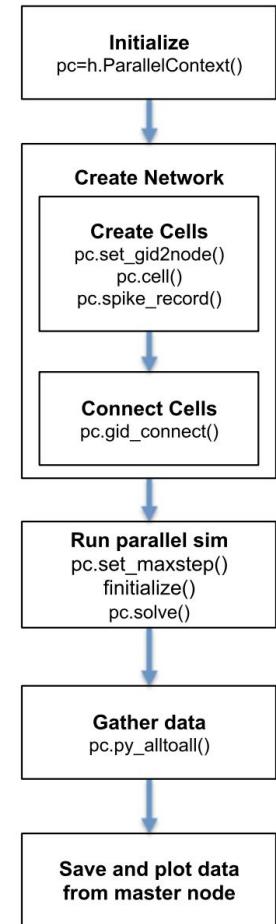
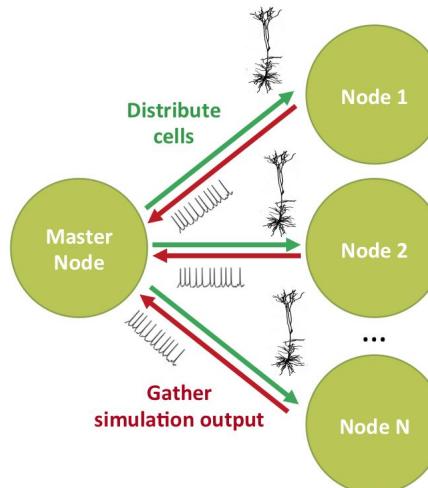
# Network instantiation

(sim.net)



# Parallel Simulation

- Set up for MPI **parallel simulation** across multiple nodes (via NEURON simulator).
- Takes care of balanced **distribution** of cells and **gathering** of simulation output from nodes.

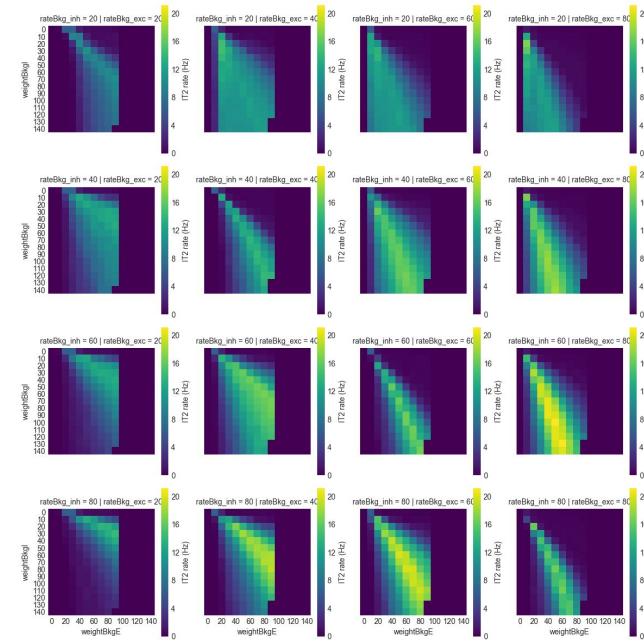


# Parallelized parameter exploration

- Easy specification of parameters values to explore in batch simulations (param sweep and evolutionary)
- Pre-defined, configurable setups to automatically submit jobs in multicore machines (Bulletin board) or supercomputers (SLURM or PBS Torque)



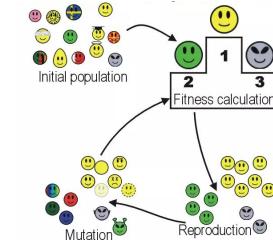
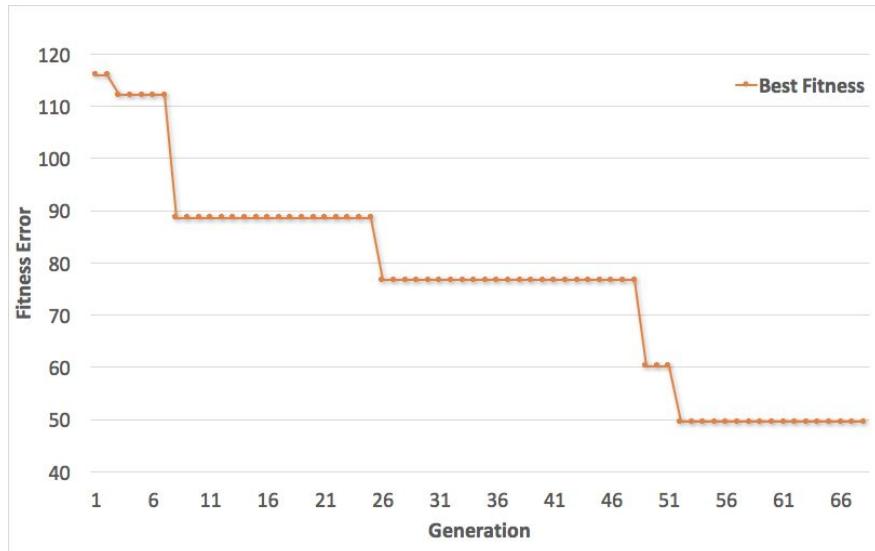
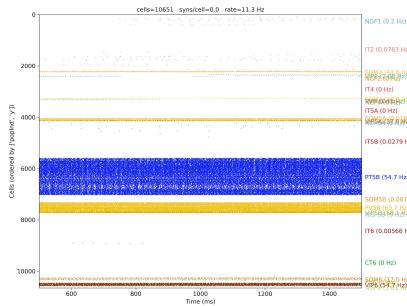
**SDSC** SAN DIEGO  
SUPERCOMPUTER CENTER



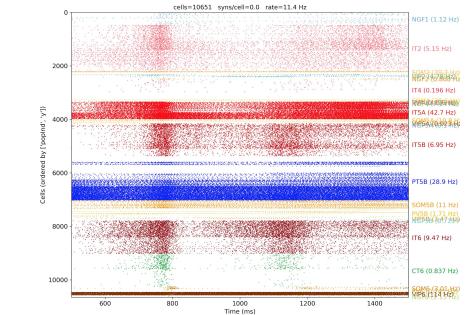
# Parallelized parameter exploration

- Evolutionary algorithm optimization (for networks)

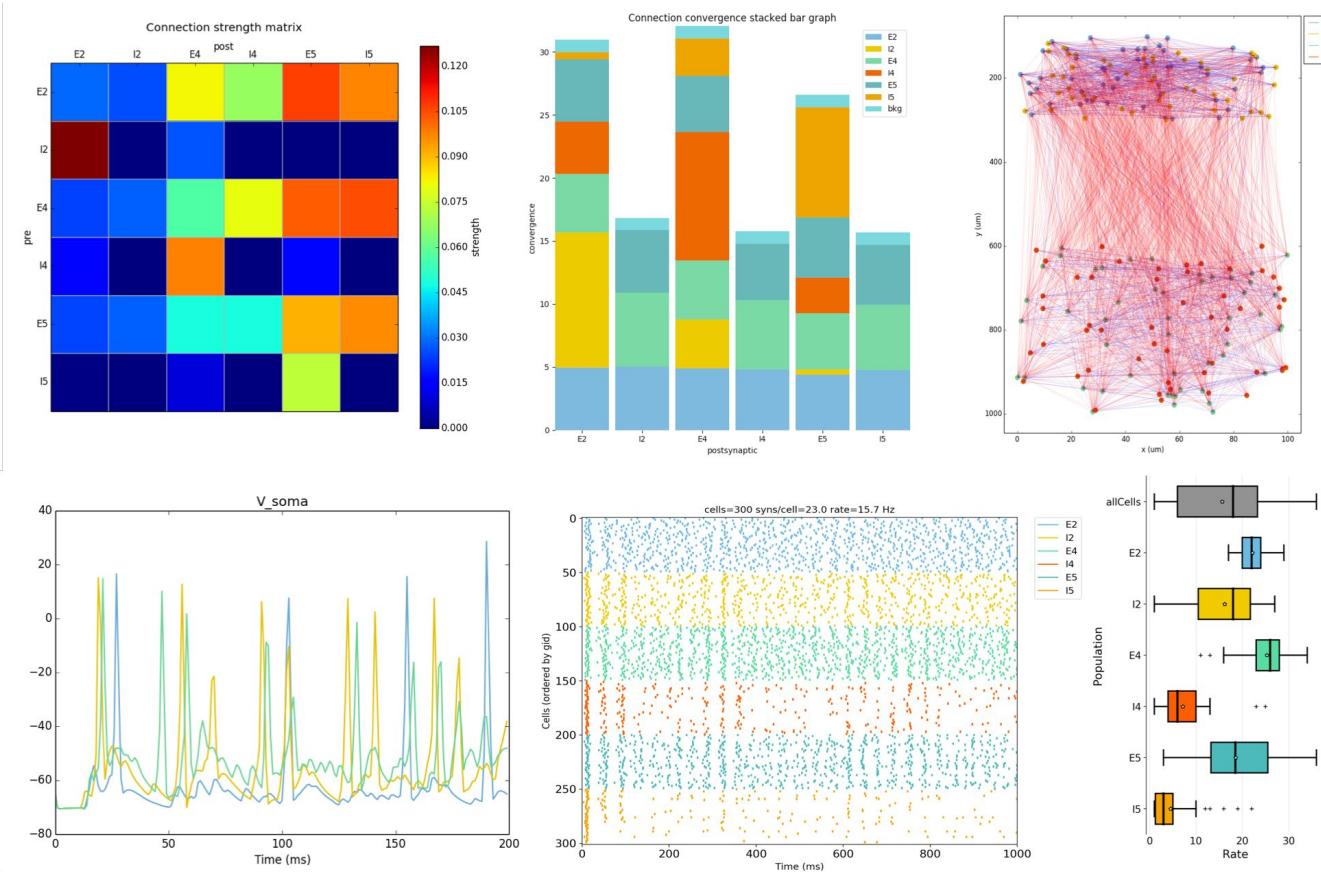
## Initial candidate



## Final candidate

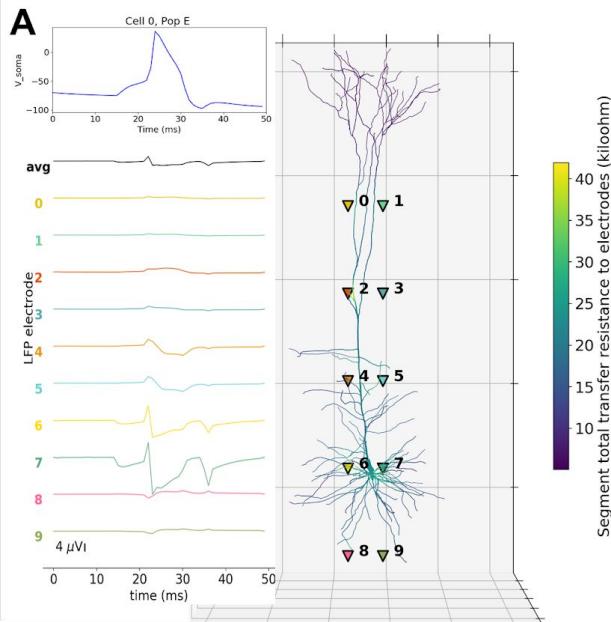


# Simulation Analysis

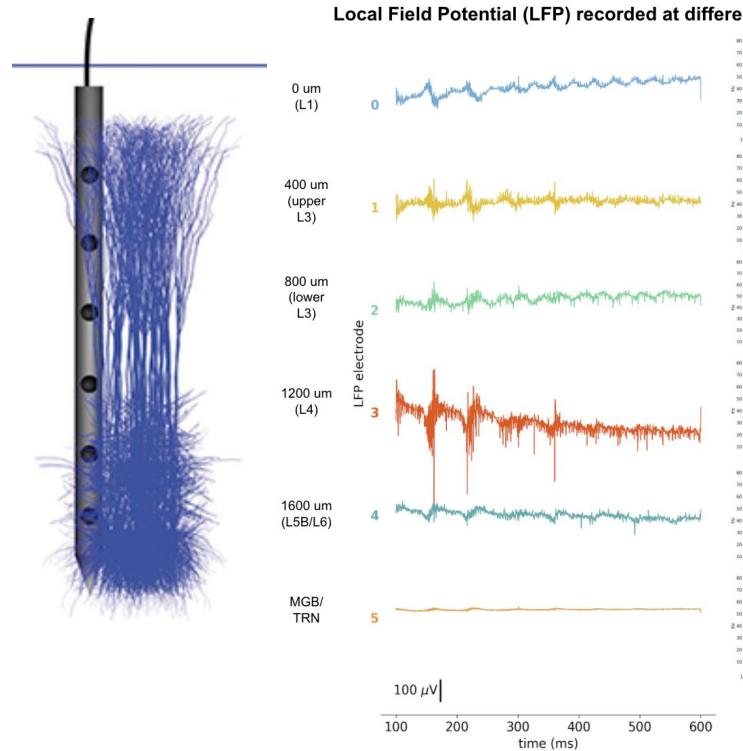


# Local Field Potentials

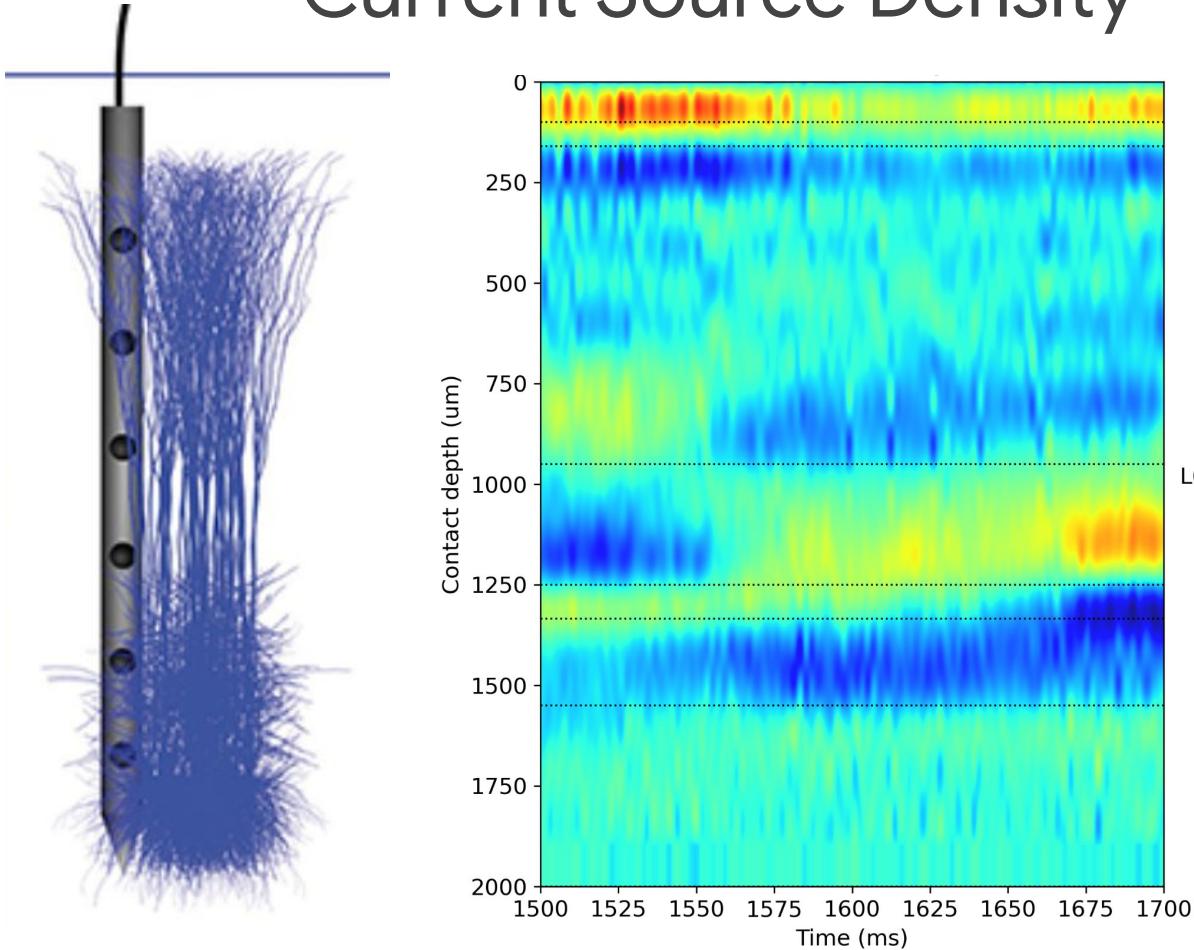
## Single cell



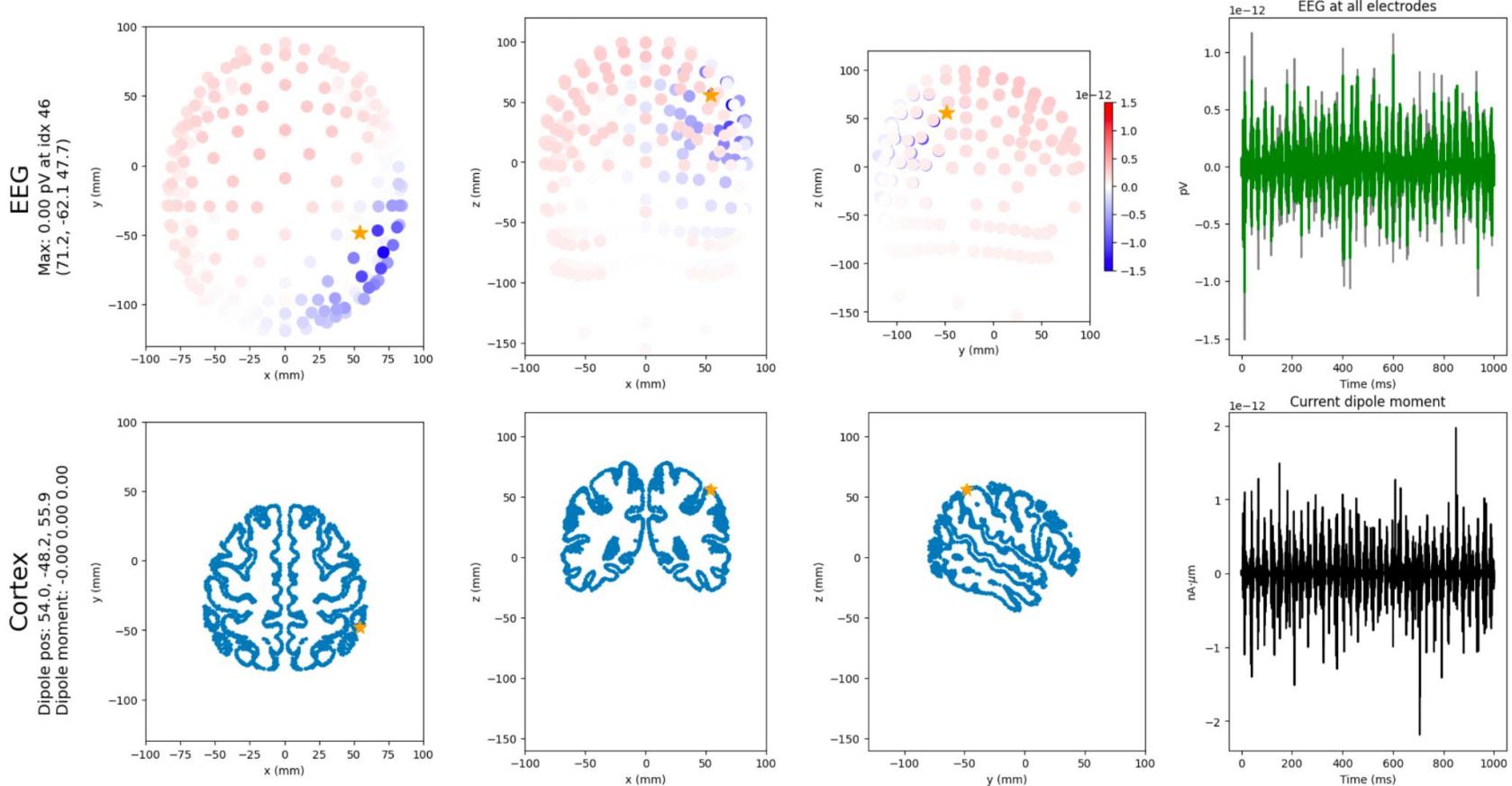
## Network



# Current Source Density



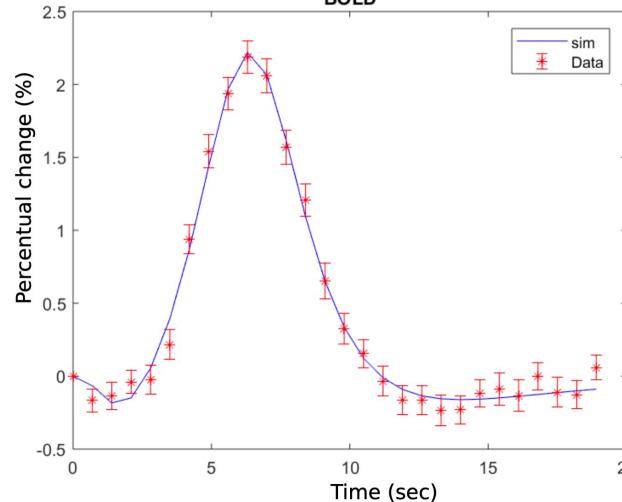
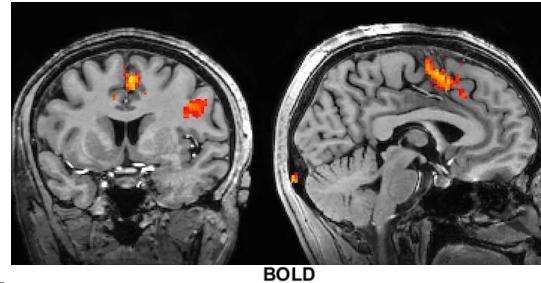
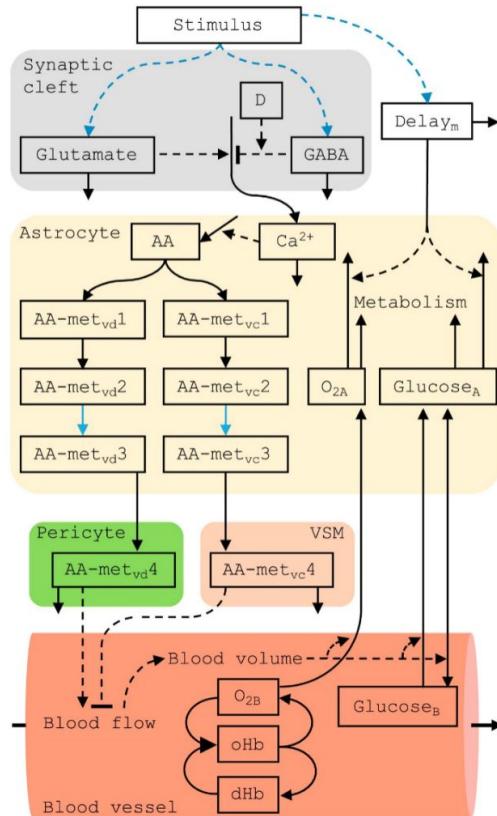
# Current dipoles and EEG



# Neurovascular coupling model (BOLD / fMRI)

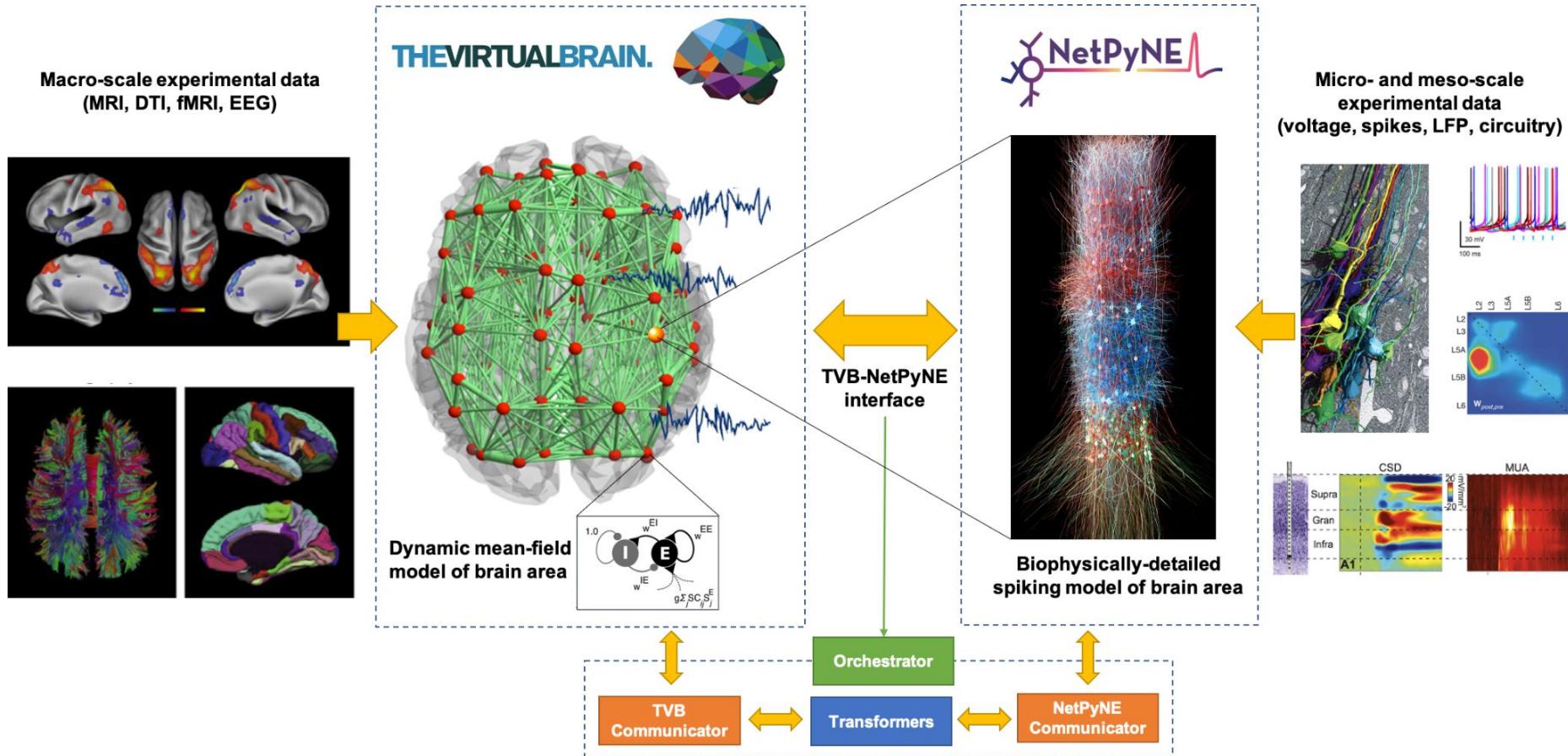
A

## Model representation



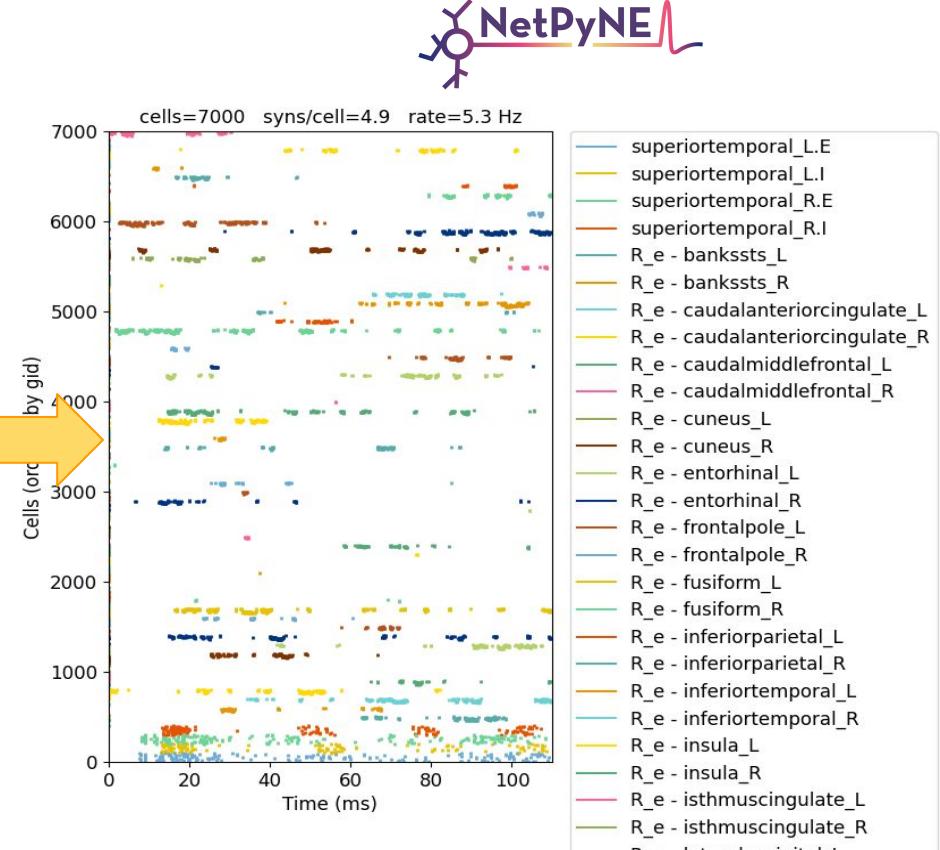
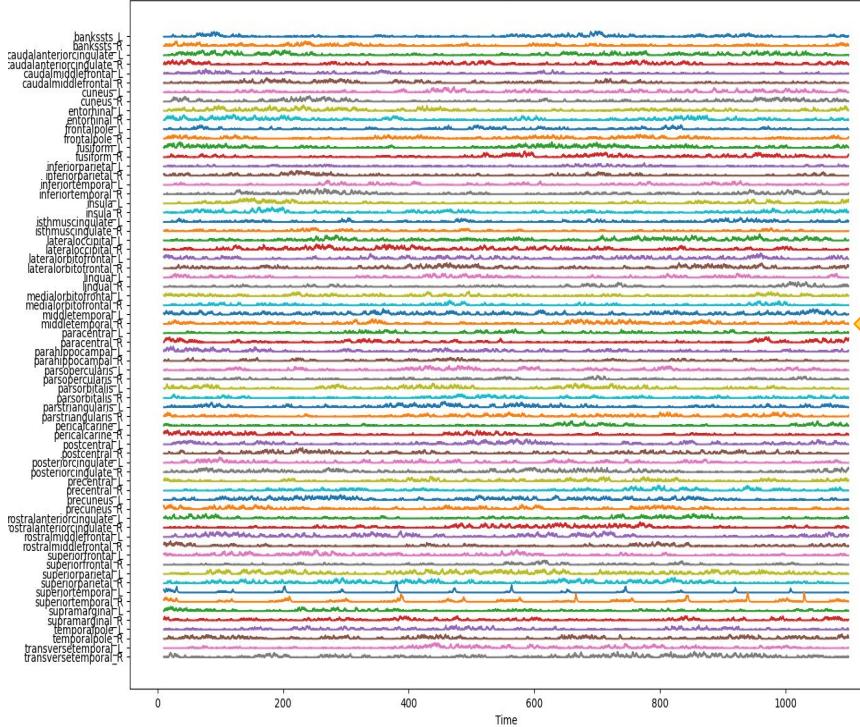
Sten S, et.al. Neural inhibition can explain negative BOLD responses: A mechanistic modelling and fMRI study. Neuroimage. 2017 Sep 1;158:219–31.

# TVB-NetPyNE co-simulation

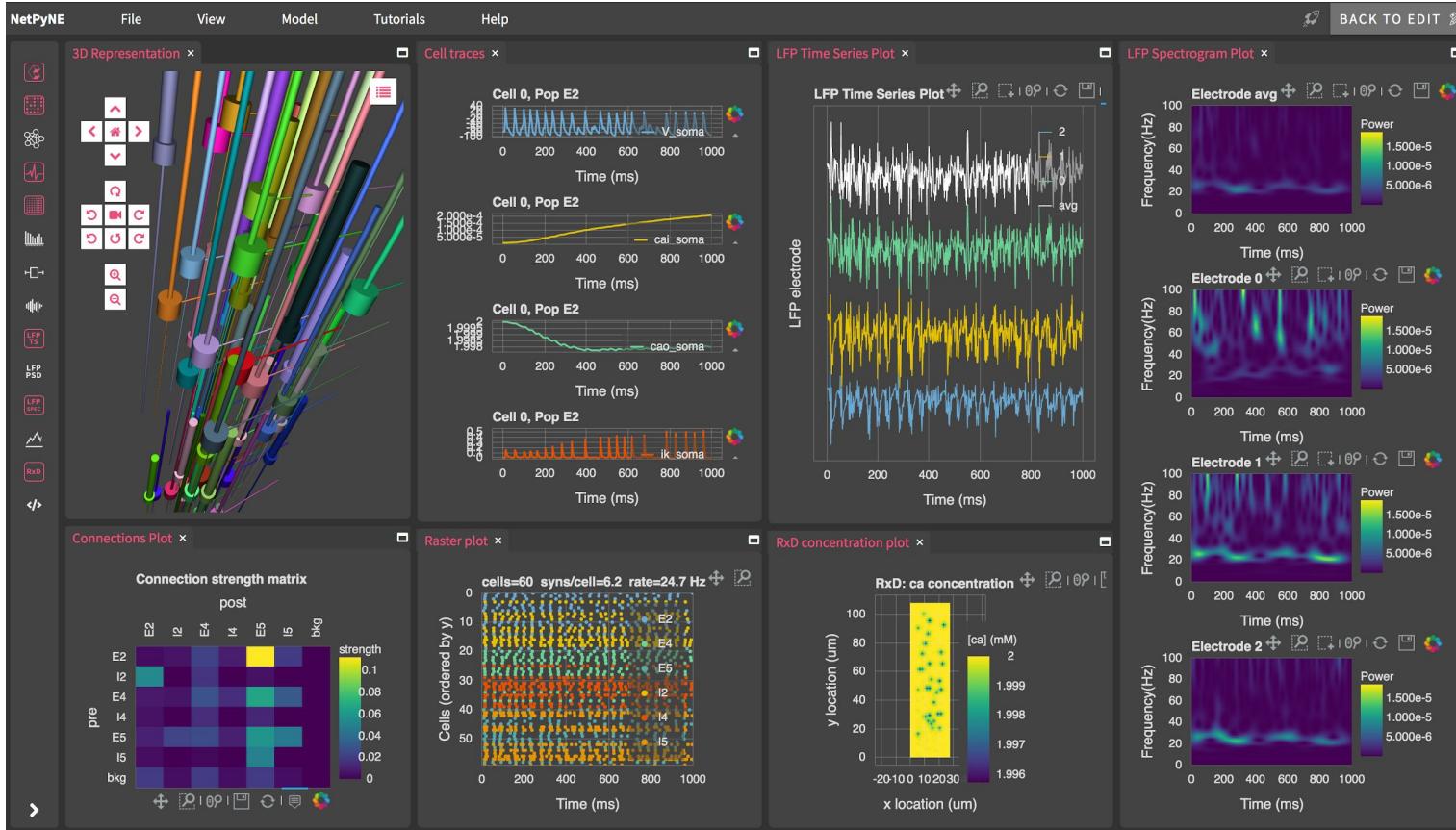


# TVB-NetPyNE Example

THE VIRTUAL BRAIN.



# GUI for Development, Simulation, Analysis



**Useful for**

- Students/beginners
- Model prototyping
- Exploring/Visualizing (interactive in sync)

# GUI for Development, Simulation, Analysis

NetPyNE    File    View    Model    Tutorials    Help    [GO TO EXPLORER](#)    [CREATE NETWORK](#) ▾

Model Specification

- Cell Types
- Populations
- Synaptic Mechanisms
- Connectivity Rules
- Stim. sources
- Stim. targets
- Plot Settings
- RxD Configuration
- Configuration

Tools

- Experiment Manager
- Python

Cell Types x   Populations x   Synaptic Mechanisms x   Connectivity Rules x   Stim. sources x   Stim. targets x   Plot Settings x   RxD Configuration x   Configuration x

The name of the mechanism  
kBK

gpeak  
0.01529200755489

caPh  
0.002

caPk  
1

caPmax  
1

caPmin  
0

caVhh  
0.002

caVhmax  
155.67

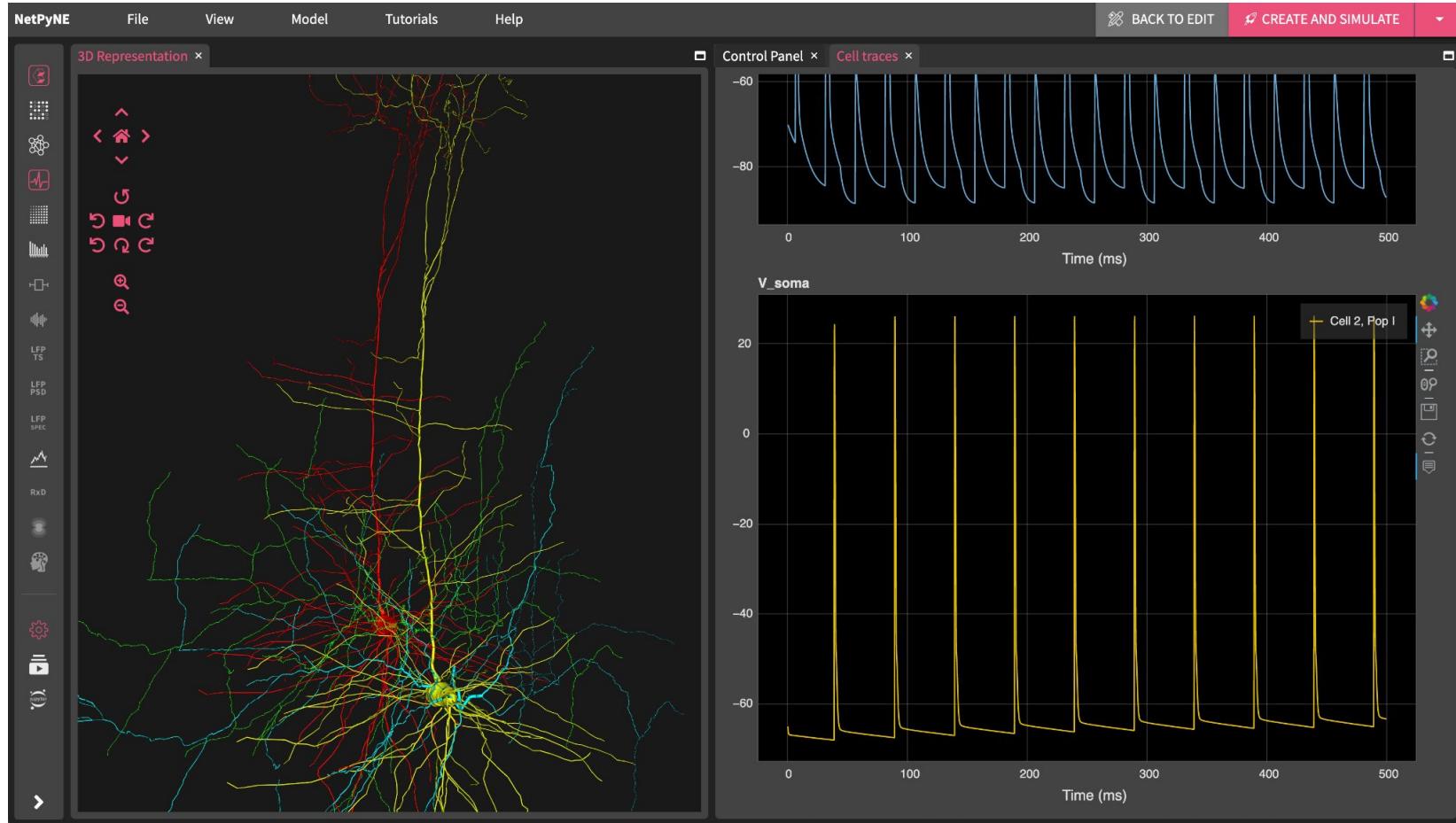
caVhmin  
-2.1899738592999967

k  
17

tau  
1

This screenshot shows the NetPyNE graphical user interface. The main window has a dark theme with a top navigation bar containing links for File, View, Model, Tutorials, Help, GO TO EXPLORER, CREATE NETWORK, and a dropdown menu. On the left, there's a sidebar with sections for Model Specification (Cell Types, Populations, etc.) and Tools (Experiment Manager, Python). The central area is titled 'Cell Types x' and shows a hierarchical tree: Cell > CT > S > Mech. Below this tree is a grid of gears, each representing a synaptic mechanism: kBK (highlighted in pink), pas, cat, ih, kap, can, cal, nax, and kdr. To the right of the grid is a detailed configuration panel for the 'kBK' mechanism, listing various parameters with their current values.

# GUI for Development, Simulation, Analysis



# Documentation, tutorials and examples

Documentation:

[http://doc.netpyne.org/user\\_documentation.html](http://doc.netpyne.org/user_documentation.html)

Package index: [http://netpyne.org/package\\_index.html](http://netpyne.org/package_index.html)

Tutorials: <http://netpyne.org/tutorial.html>

Interactive Tutorials:

<https://github.com/suny-downstate-medical-center/netpyne/tree/development/netpyne/tutorials>

Video tutorials:

<https://www.youtube.com/channel/UCSMXo4L1mfVgw6LgbFAxKTg>

Example models: [www.netpyne.org/models](http://www.netpyne.org/models)

Q&A Forum: [www.netpyne.org/forum](http://www.netpyne.org/forum)

GUI: <https://netpyne.v2.opensourcebrain.org>

## Connect the cells

Now we will specify the connectivity in our model by adding an entry to the `connParams` dictionary. We will call our connectivity rule `E->E` as it will define connectivity from our `E` population to our `E` population.

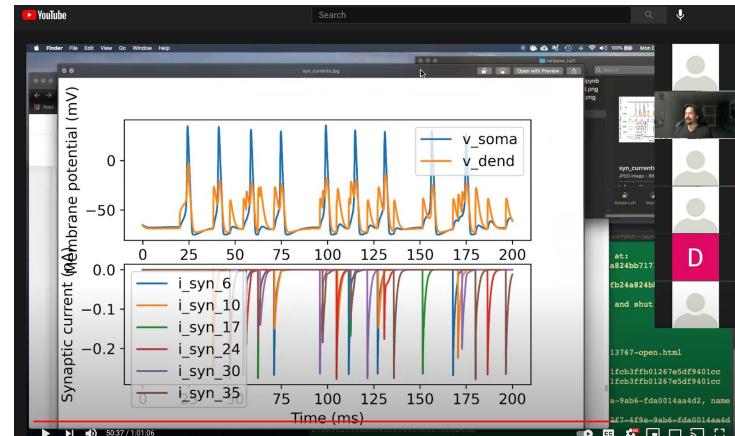
We will use the `synMech exc`, which we defined above. For this synaptic mechanism, a weight of about 0.005 is appropriate. These cells will have a 10% probability of getting connected, and will be activated five milliseconds after an action potential occurs in the presynaptic cell. Synapses will occur on the `dend` section at its very end (`location 1.0`)

```
In [ ]: netParams.connParams['E->E'] = {  
    "preConds": {"pop": "E"},  
    "postConds": {"pop": "E"},  
    "weight": 0.005,  
    "probability": 0.1,  
    "delay": 5.0,  
    "synMech": "exc",  
    "sec": "dend",  
    "loc": 1.0,  
}
```

## Set up the simulation configuration

```
In [ ]: simConfig.filename = "netpyne_tut"  
simConfig.duration = 200.0  
simConfig.dt = 0.1
```

We will record from the first cell (0) and we will record the voltage in the middle of the soma and the end of the dendrite.



# NetPyNE tool: Publication in eLife

The image shows the header of the eLife website. It features the eLife logo on the left, followed by a horizontal navigation bar with links for "ABOUT", "COMMUNITY", "SUBMIT MY RESEARCH", and a user icon. Below this is a secondary navigation bar with "HOME", "MAGAZINE", and "INNOVATION" links, along with a search icon.

Accepted manuscript, PDF only. Full online edition to follow.



COMPUTATIONAL AND SYSTEMS BIOLOGY, NEUROSCIENCE



## NetPyNE, a tool for data-driven multiscale modeling of brain circuits



Salvador Dura-Bernal , Benjamin A Suter, Padraig Gleeson, Matteo Cantarelli, Adrian Quintana, Facundo Rodriguez, David J Kedziora, George L Chadderdon, Cliff C Kerr, Samuel A Neymotin, Robert A McDougal, Michael Hines, Gordon M G Shepherd, William W Lytton  
[« see less](#)

State University of New York Downstate Medical Center, United States; Northwestern University, United States; University College London, United Kingdom; Metacell LLC, United States; EyeSeeTea Ltd, United Kingdom; University of Sydney, Australia; Yale University, United States

TOOLS AND RESOURCES Apr 26, 2019

CITED 0 VIEWS 272 ANNOTATIONS

CITE AS: eLife 2019;8:e44494 DOI: 10.7554/eLife.44494

# NetPyNE-based papers

<http://netpyne.org/about.html#publications>

nature > schizophrenia

Article | Open Access | Published: 28 April 2022

## The effect of alterations of schizophrenia-associated genes on gamma band oscillations

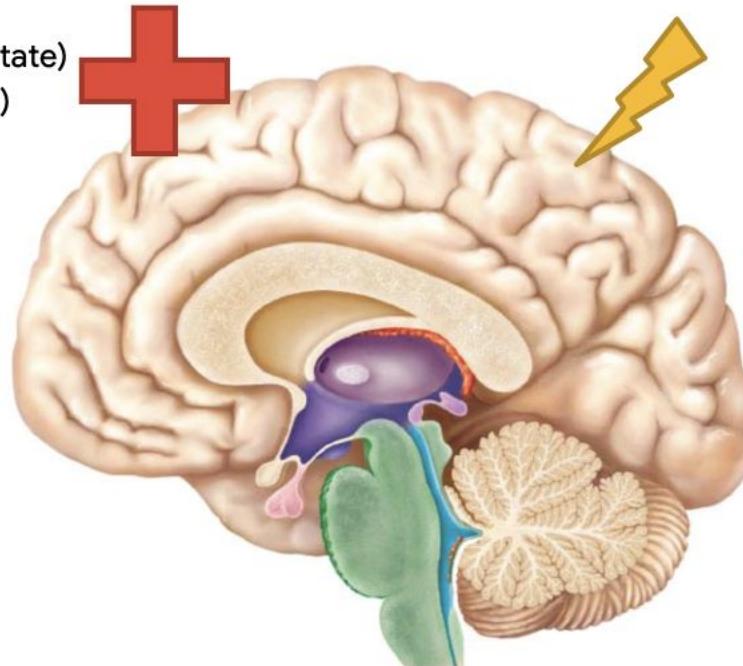
Christoph Metzner , Tuomo Mäki-Marttunen, Gili Karni, Hana McMahon-Cole & Volker Steuber

*Schizophrenia* 8, Article number: 46 (2022) | [Cite this article](#)

- Hasegan D, Deible M, Earl C, D'Onofrio D, Hazan H, Anwar H, Neymotin SA. Multi-timescale biological learning algorithms train spiking neuronal network motor control bioRxiv 2021.11.20.469405, 2021. doi: <https://doi.org/10.1101/2021.11.20.469405>.
- Bryson A, Berkovic SF, Petrou S, Grayden DB. State transitions through inhibitory interneurons in a cortical network model. PLoS Comput Biol 17(10): e1009521, 2021. doi: <https://doi.org/10.1371/journal.pcbi.1009521>.
- Pimentel JM, Moioli RC, de Araujo MFP, Ranieri CM, Romero RAF, Broz F and Vargas PA. Neuro4PD: An Initial Neurorobotics Model of Parkinson's Disease. Front. Neurorobot. 15:640449, 2021. doi: [10.3389/fnbot.2021.640449](https://doi.org/10.3389/fnbot.2021.640449).
- Volk, V.L., Hamilton, L.D., Hume, D.R. et al. Integration of neural architecture within a finite element framework for improved neuromusculoskeletal modeling. Scientific Reports 11, 22983, 2021. doi: <https://doi.org/10.1038/s41598-021-02298-9>.
- Ranieri CM, Moioli RC, Vargas P, Romero RAF, A Neurorobotics Approach to Behaviour Selection based on Human Activity Recognition, arXiv:2107.12540 [cs.RO], 2021. <https://arxiv.org/abs/2107.12536>.
- Ranieri CM, Pimentel JM, Romano MR, Elias LA, Romero RAF, Lones MA, Araujo MFP, Vargas PA, Moioli RC, A Data-Driven Biophysical Computational Model of Parkinson's Disease Based on Marmoset Monkeys, IEEE Access, vol. 9, pp. 122548-122567, 2021. doi: <https://doi.org/10.1109/ACCESS.2021.3108682>.
- Metzner C, Mäki-Marttunen T, Karni G, McMahon-Cole H, Steuber V. The Effect of Alterations of Schizophrenia-Associated Genes on Gamma Band Oscillations, bioRxiv 2020.09.28.316737, 2020. doi: <https://doi.org/10.1101/2020.09.28.316737>.
- Gao P, Graham J, Zhou W, Jang J, Angulo S, Dura-Bernal S, Hines ML, Lytton WW, and Antic S Local Glutamate-Mediated Dendritic Plateau Potentials Change the State of the Cortical Pyramidal Neuron, Journal of Neurophysiology (In Press). 2020.
- Sivagnanam S, Gorman W, Doherty D, Neymotin S, Hovhannisanian H, Fang S, Lytton W, Dura-Bernal S Simulating large-scale models of brain neuronal circuits using Google Cloud Platform, Practice and Experience in Advanced Research Computing, PEARC2020, 2020.
- Neymotin, S.A., Daniels, D.S., Caldwell, B., McDougal, R.A., Carnevale, N.T., Jas, M., Moore, C.I., Hines, M.L., Hamalainen, M. and Jones, S.R., Human Neocortical Neurosolver (HNN), a new software tool for interpreting the cellular and network origin of human MEG/EEG data, eLife, 9, p.e51214, 2020. doi: <https://elifesciences.org/articles/51214> .
- Lytton WW, Sherif MA, Dura-Bernal S, Neymotin S. Multiscale modeling of a "mental" disease, Virtual Physiological Human Conference, VPH2020, 2020.
- Dura-Bernal S, Neymotin SA, Suter BA, Shepherd GMG, Lytton WW. Multiscale dynamics and information flow in a data-driven model of the primary motor cortex microcircuit, bioRxiv 201707, 2019. doi: <https://www.biorxiv.org/content/10.1101/201707v3>.
- Gerkin, R.C., Birgiolas, J., Jarvis, R.J., Omar, C. and Crook, S.M.. NeuronUnit: A package for data-driven validation of neuron models using SciUnit, bioRxiv, p.665331, 2019.
- Gleeson P, Cantarelli M, Quintana A, Earnsah M, Piasini E, Birgiolas J, Cannon RC, Cayco- Gajic A, Crook S, Davison AP, Dura-Bernal S, et al. Open Source Brain: a collaborative resource for visualizing, analyzing, simulating and developing standardized models of neurons and circuits, Neuron, 10.1016/j.neuron.2019.05.019, 2019. doi: [https://www.cell.com/neuron/fulltext/S0896-6273\(19\)30444-1](https://www.cell.com/neuron/fulltext/S0896-6273(19)30444-1).
- Dai K, Hernando J, Billeh JN, Gratay SL, Planas J, Davison AP, Dura-Bernal S, Gleeson P, Devresse A, Gevaert M, King JG, Van Geit WAH, Povolotsky AV, Muller E, Courcol J-D, Arkhipov A. The SONATA Data Format for Efficient Description of Large-Scale Network Models, bioRxiv, 625491 [Preprint] doi: <https://www.biorxiv.org/content/10.1101/625491v2>. Under review in PLoS Computational Biology. 2019
- Dura-Bernal S, Neymotin SA, Suter BA, Shepherd GMG, Lytton WW Long-range inputs and H-current regulate different modes of operation in a multiscale model of mouse M1 microcircuits, bioRxiv 201707, 2018. doi: <https://www.biorxiv.org/content/10.1101/201707v3>.

# Growing community (>80 models, >40 Labs)

- Schizophrenia (TU Berlin, Brown)
- Ischemic stroke (Yale, Downstate)
- Epilepsy (Cincinnati, Brown, Downstate)
- Chronic Pain (Okinawa, Downstate)
- Depression (Brown)
- Parkinson's (Edinburgh)



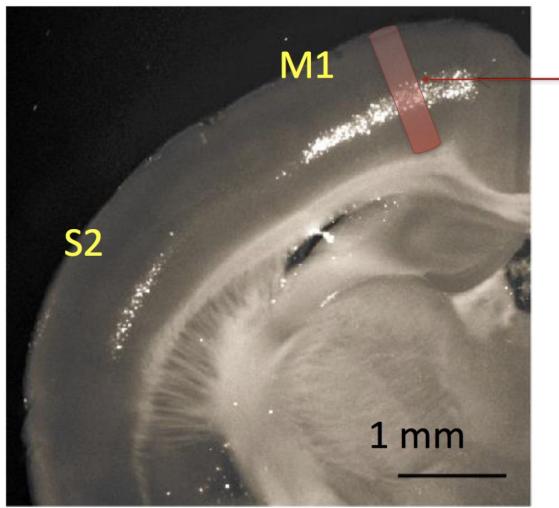
- PFC (Sao Paulo)
- Thalamus (UCL, Missouri)
- Olfactory Bulb (Palermo)
- Striatum (Dublin)
- Amygdala (Princeton)
- Hippocampus (Sao Paulo)
- Cardiac circuits (Jefferson, Downstate, Pavia)



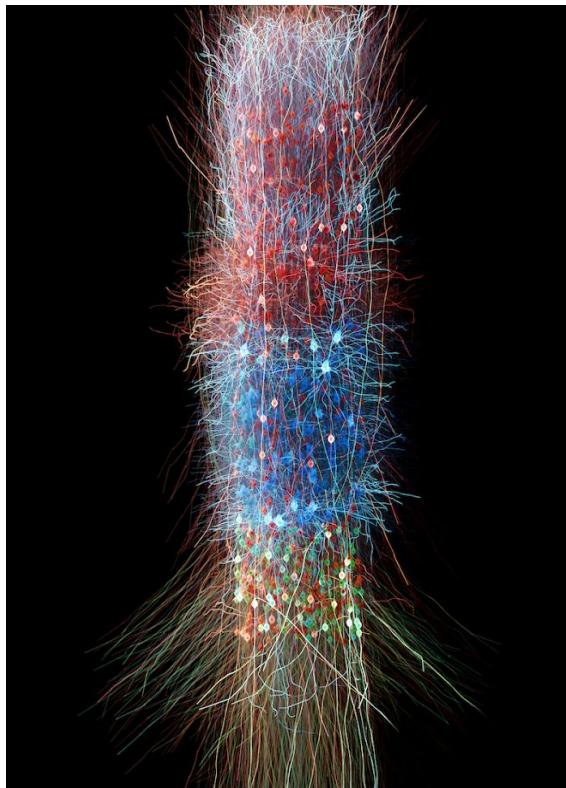
- TMS / tDCS / tACS (Duke, Toronto)
- Optogenetics (Sydney)
- Electrical stimulation (Downstate)
- Ketamine (Brown)
- EEG/MEG (Puerto Rico, Brown, Rice)
- fMRI (Linkopig)
- M1 and S1 (Downstate)
- A1 (NKI)
- V1 (Queensland, Sao Paulo)
- Claustrum (Singapore)
- Cerebellum (Sao Paulo)
- Spinal Cord (Northeastern)
- Enteric / gastrointestinal circuits (Melbourne)

# Data-driven model of motor cortex (M1) circuits

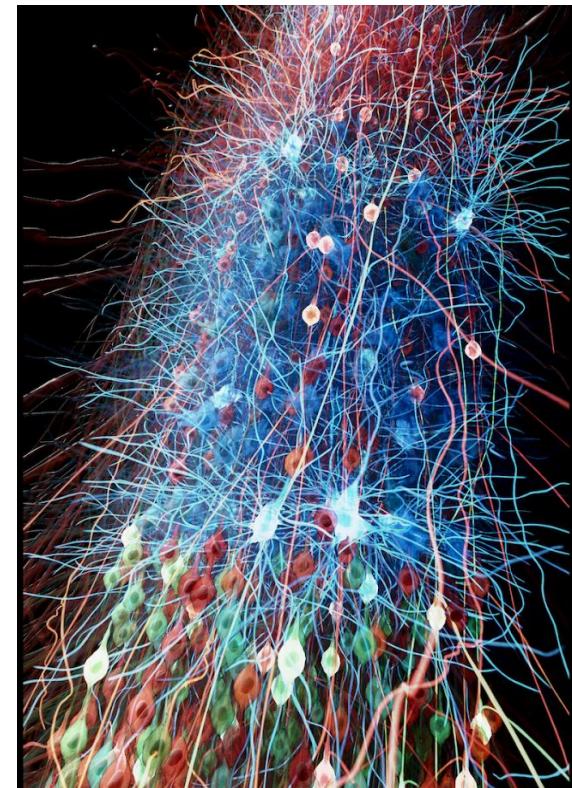
300 um diameter column



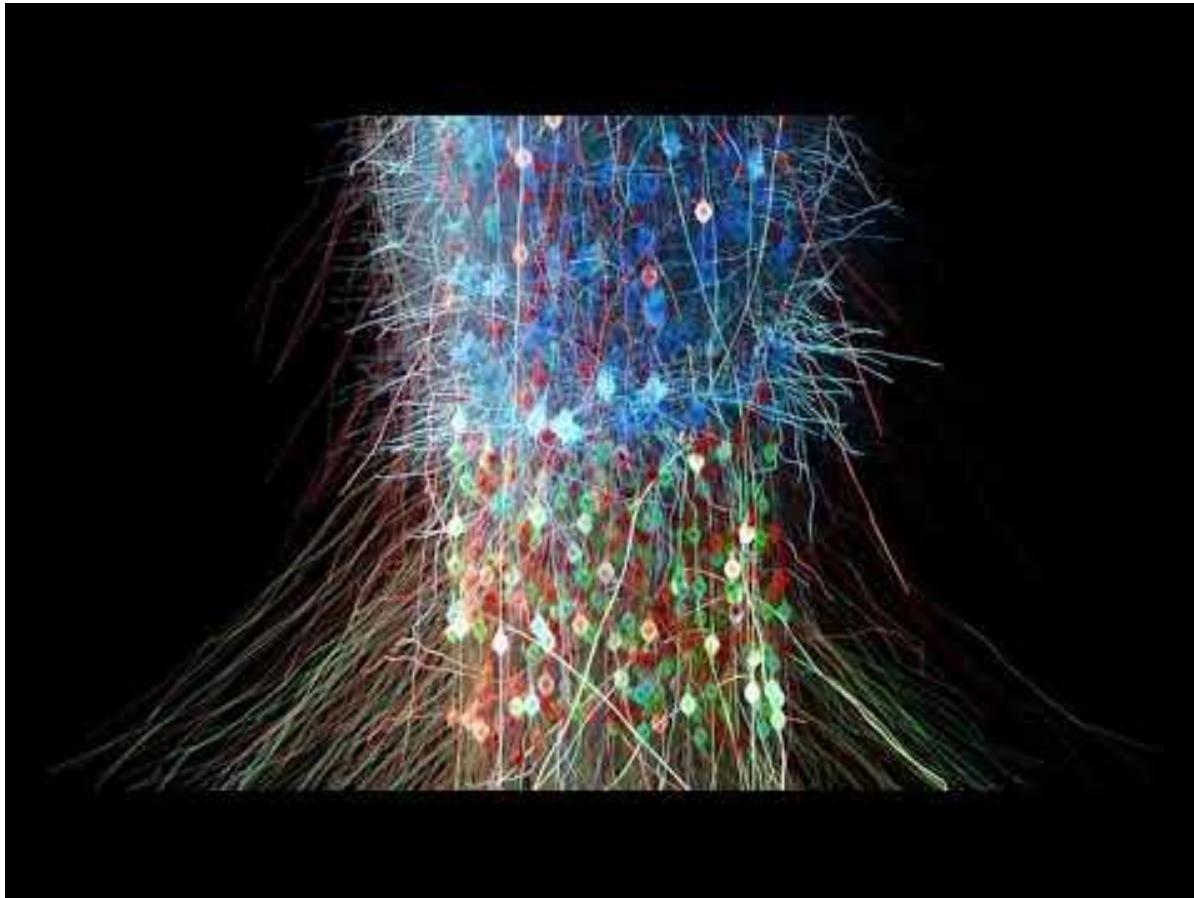
10,000 neurons



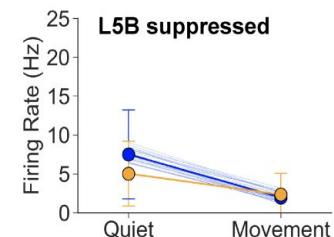
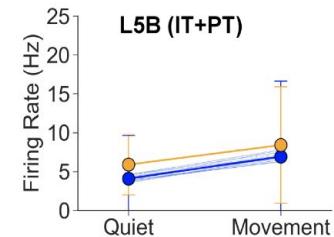
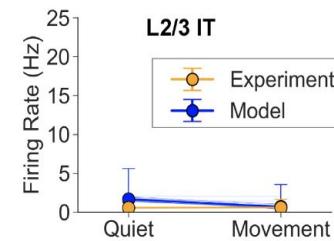
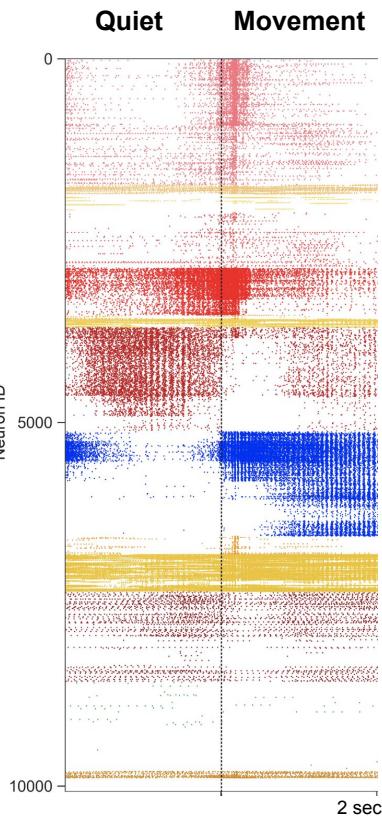
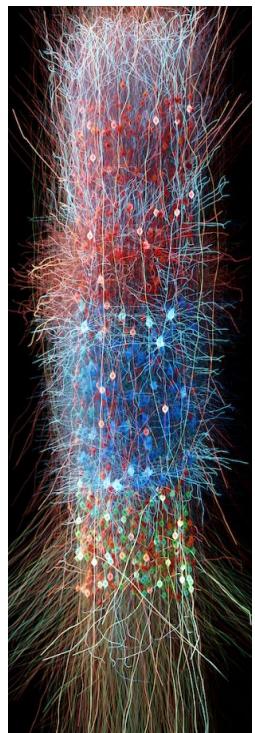
30 million connections



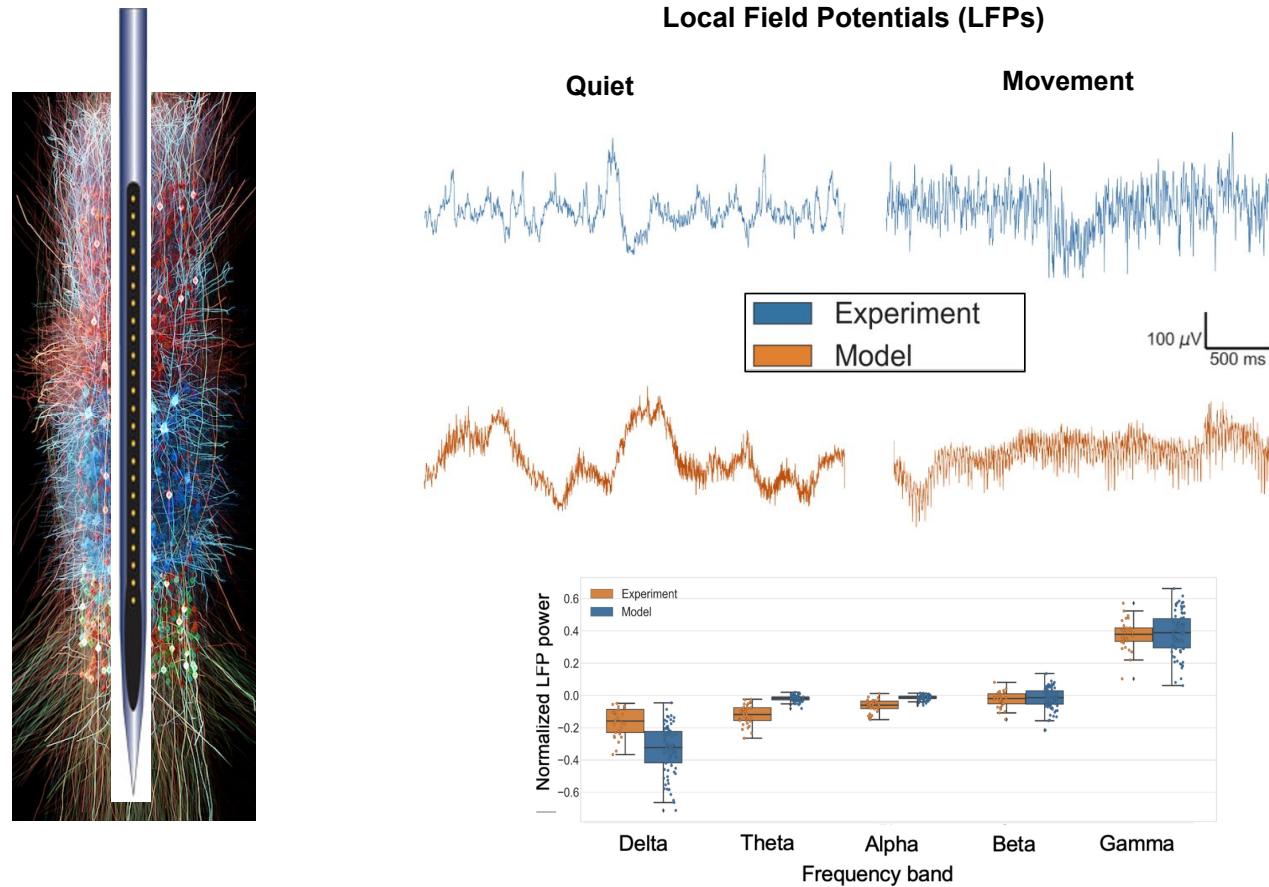
# Data-driven model of motor cortex (M1) circuits



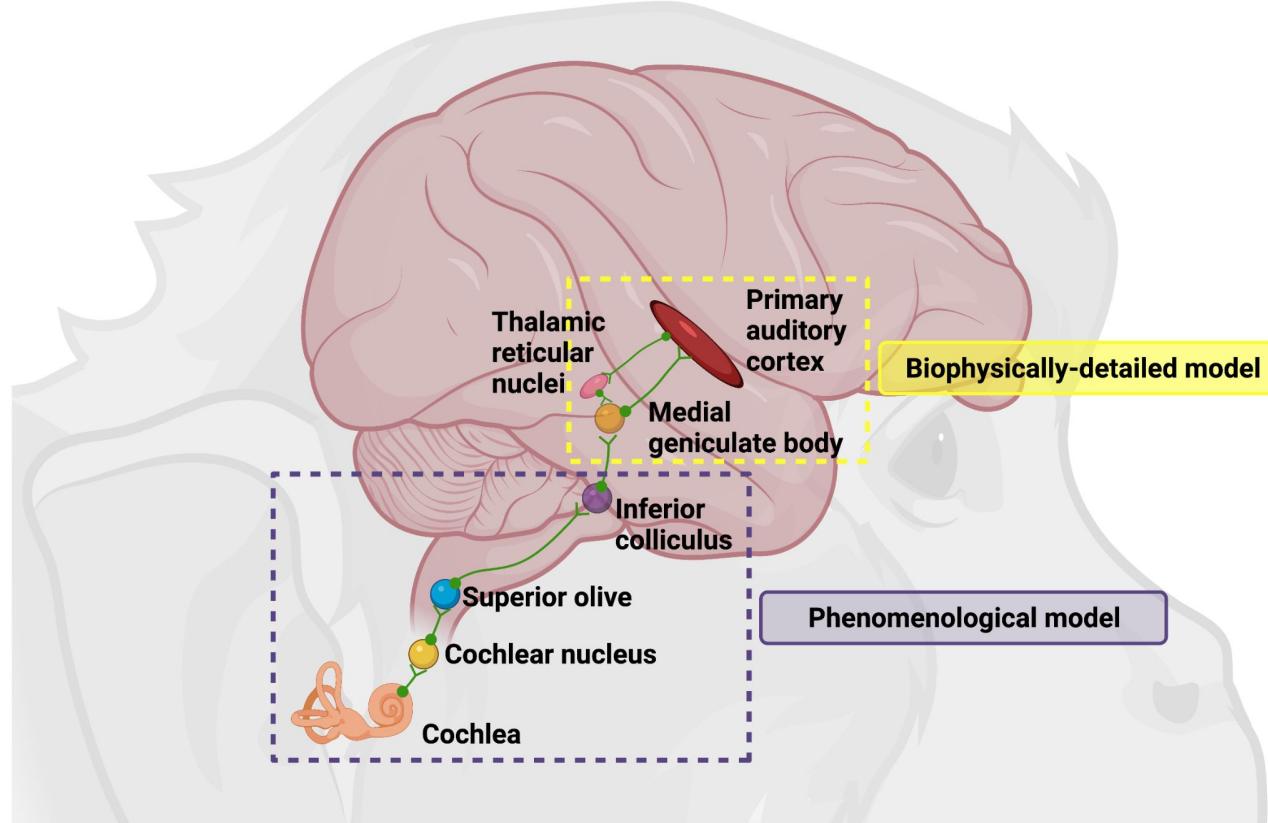
# M1 model validated against in vivo data



# M1 model validated against in vivo data

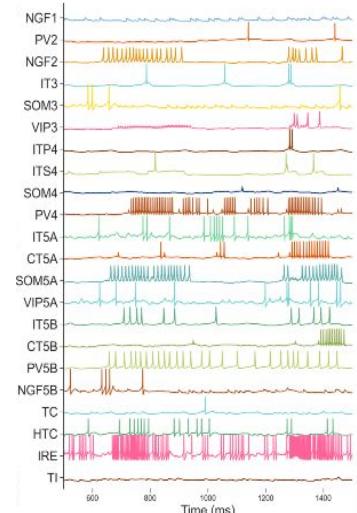


# Macaque auditory thalamocortical model

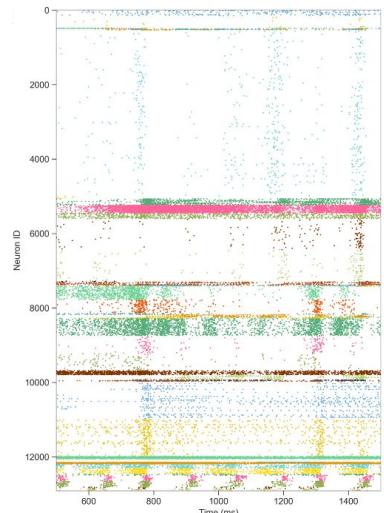


# A1 model recordings at multiple scales/modalities

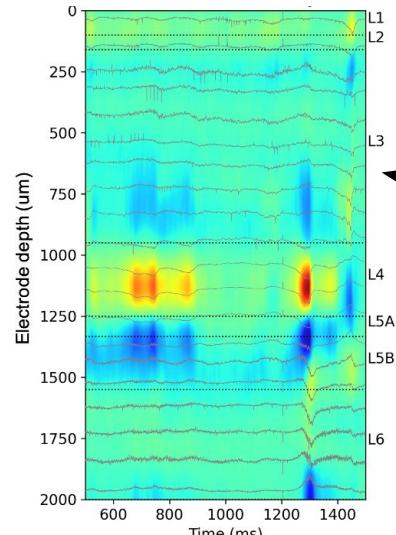
Membrane Voltage



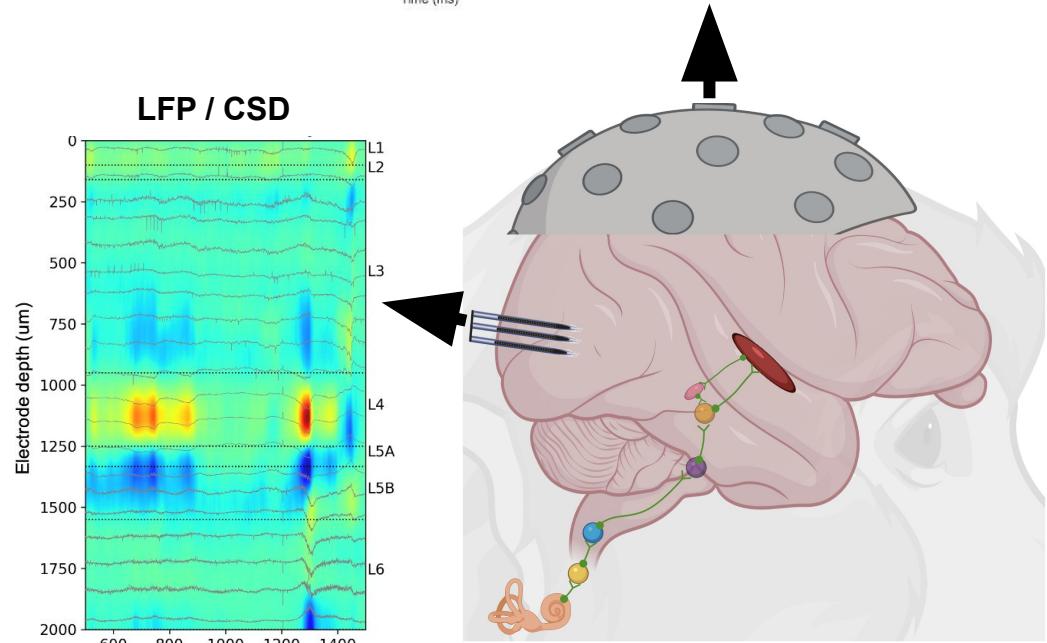
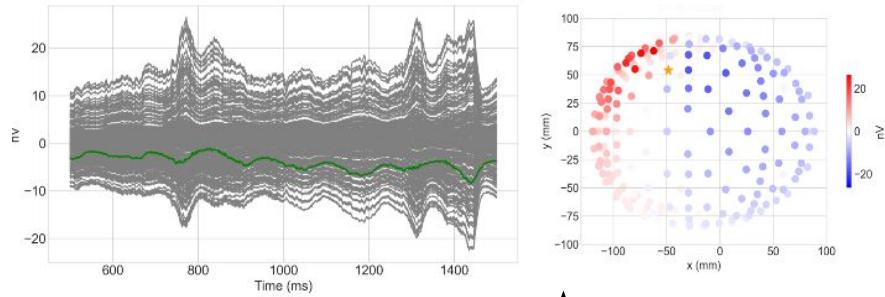
Spikes



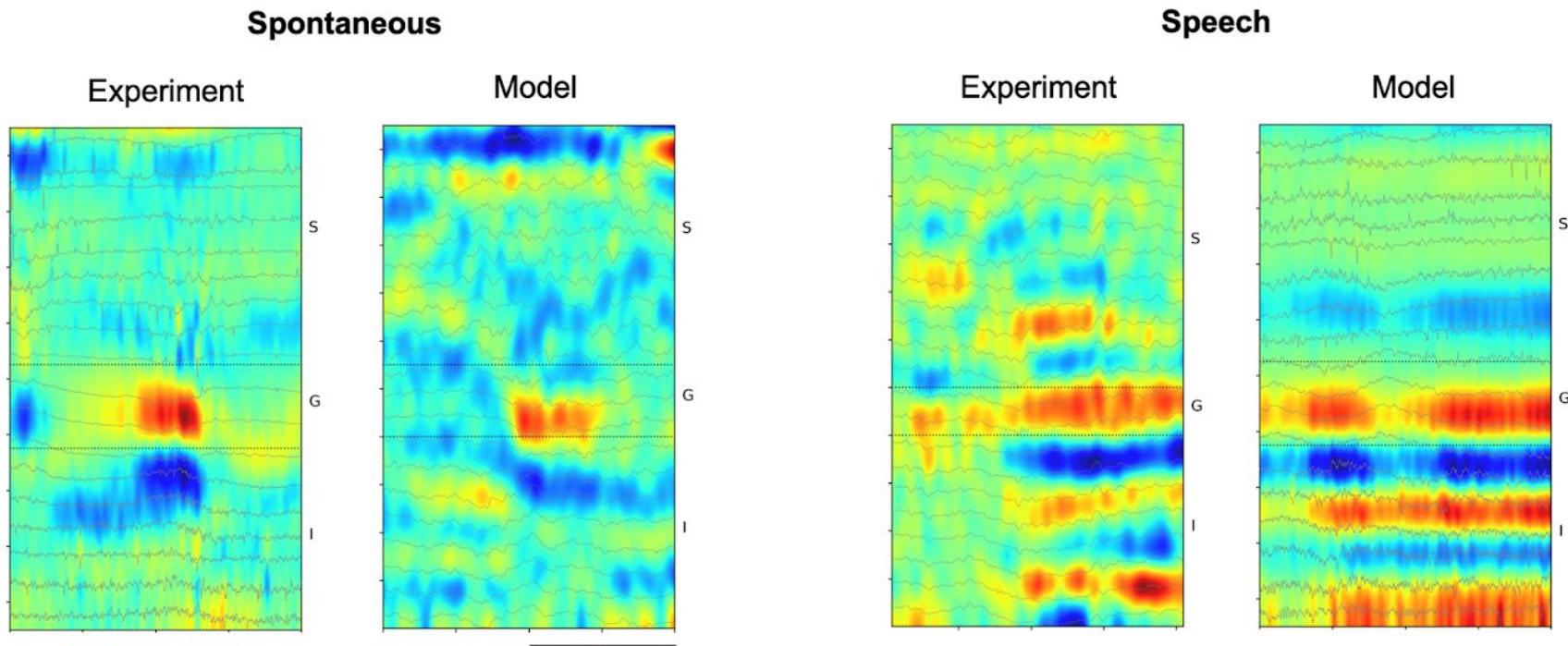
LFP / CSD



EEG

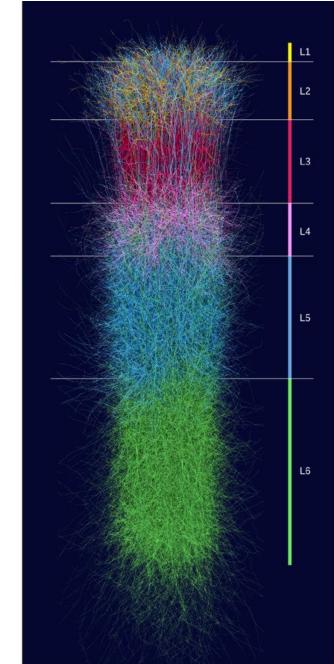
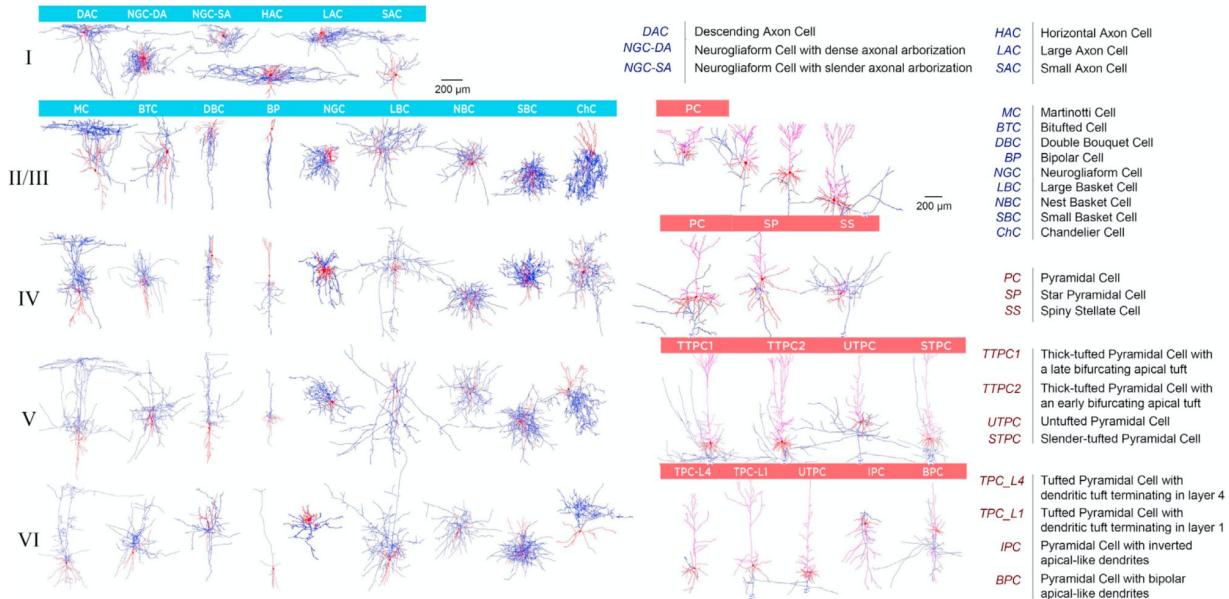


# A1 model validated against in vivo data



# Somatosensory thalamocortical circuits

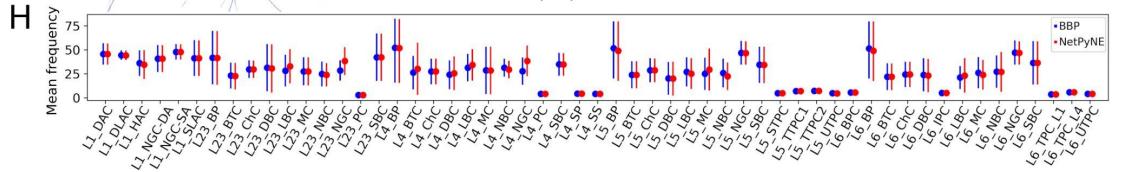
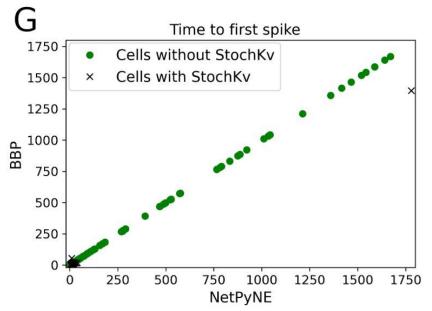
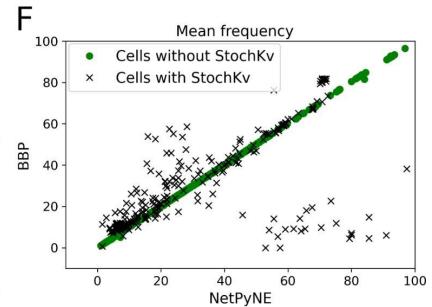
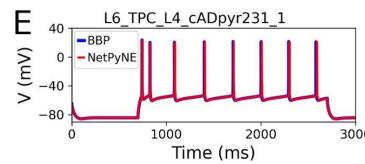
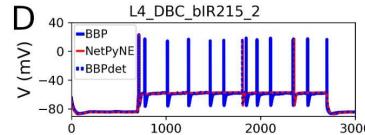
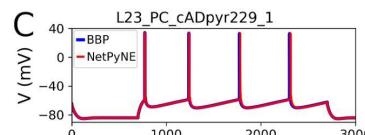
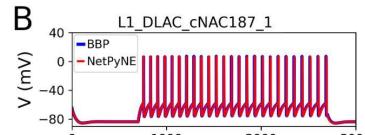
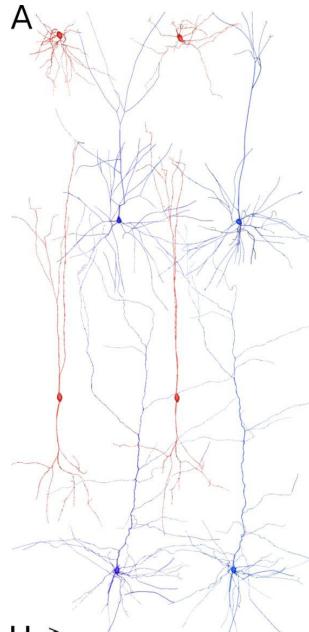
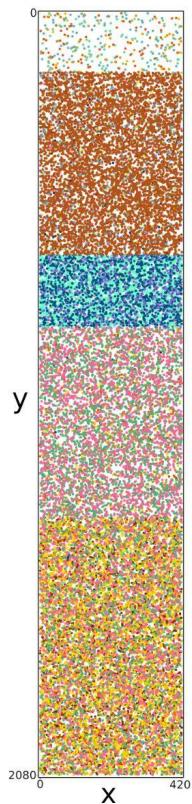
- Highly detailed model: 31k cells, 37M synapses, 207 cell types, 1941 pathways
- Original model by EPFL Blue Brain Project



Markram et al, 2015

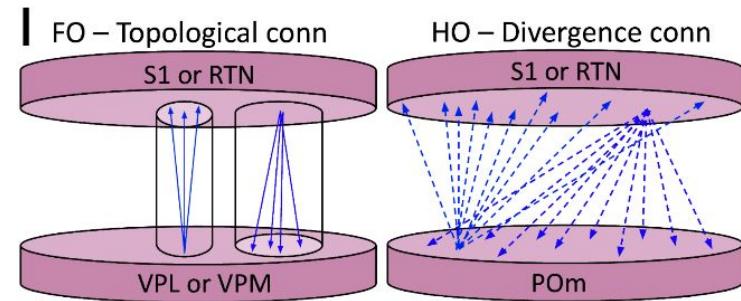
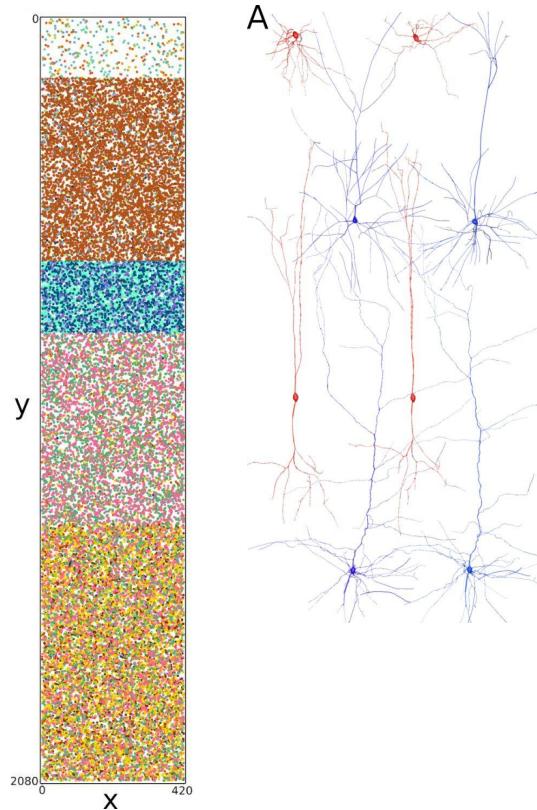
# Somatosensory thalamocortical circuits

- Reimplemented  
in NetPyNE

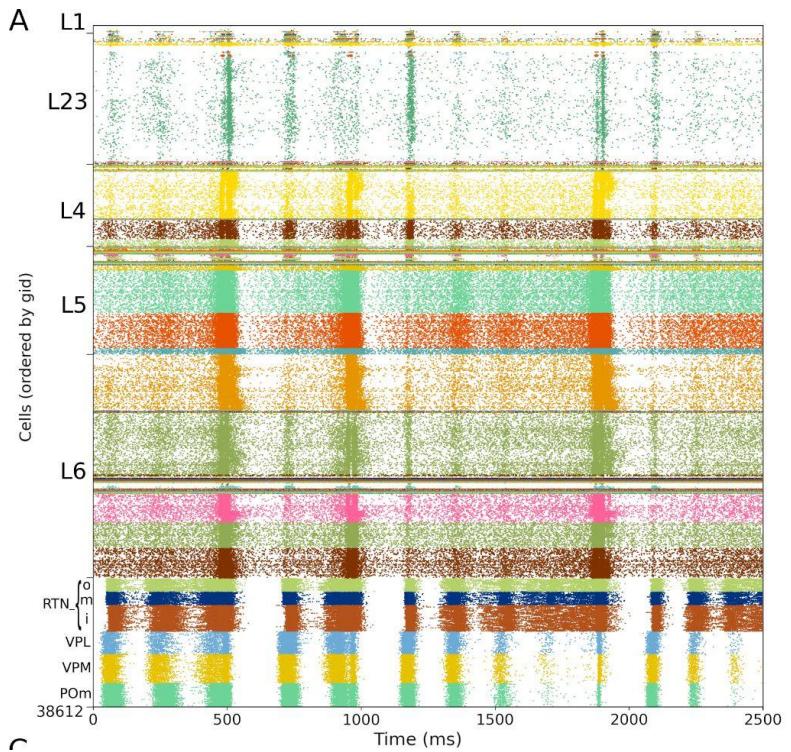
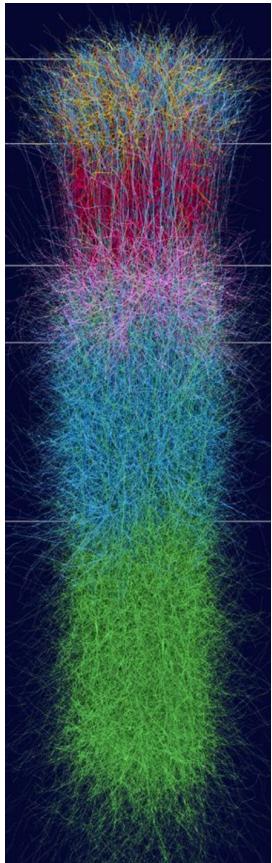


# Somatosensory thalamocortical circuits

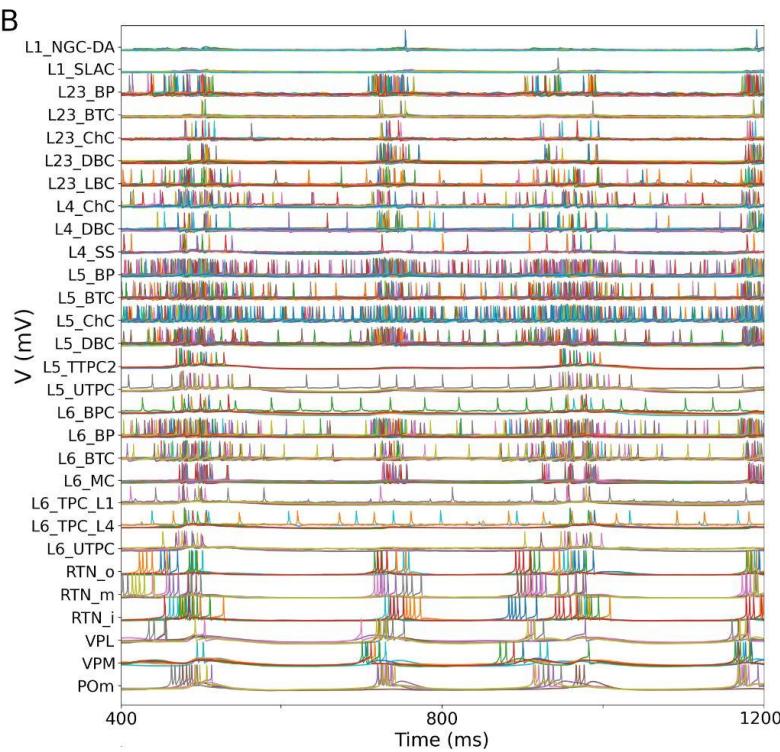
- Reimplemented in NetPyNE
- Replicated cell properties and connectivity
- Extended to include thalamus



# Somatosensory thalamocortical circuits



C



# Acknowledgements

## People involved:

### SUNY Downstate:

Salvador Dura-Bernal  
Bill Lytton  
Joe Graham  
Eugenio Urdapilletta  
Valery Bragin  
James Chen  
Don Doherty  
Erica Griffith  
Craig Kelley  
Siddartha Mitra  
Mohamed Sheriff

### Nathan Kline Institute:

Sam Neymotin  
Peter Lakatos

### Northwestern:

Gordon MG Shepherd  
Ben Suter

### Yale:

Michael Hines

### MetaCell Ltd:

Matteo Cantarelli  
Dario del Piano  
Filippo Ledda  
Facundo Rodriguez

### UCSD:

Amit Majumdar  
Subha Sivagnanam

### UCL:

Padraig Gleeson  
Angus Silver

### Brown:

Stephanie Jones  
Wilson Truccolo

### University of Sydney

Cliff Kerr  
David Kedziora

## Funding:

NSF E-CAS 1904444-1022C, 2019-2021 (M1)

NIH U24EB028998, 2019-2024 (NetPyNE)

NIH U01EB017695, 2014-2021 (M1)

NIH R01DC012947, 2012-2023 (A1)

NIH R01EB022889, 2016-2020 (HNN/NetPyNE)

NIH R01MH086638, 2010-2021 (NEURON RxD)

NYS DOH01-C32250GG, 2017-2022 (M1)

NSF XSEDE TG-IBN160014, 2017-2021 (M1)



DOWNSTATE  
SUNY  
Medical Center