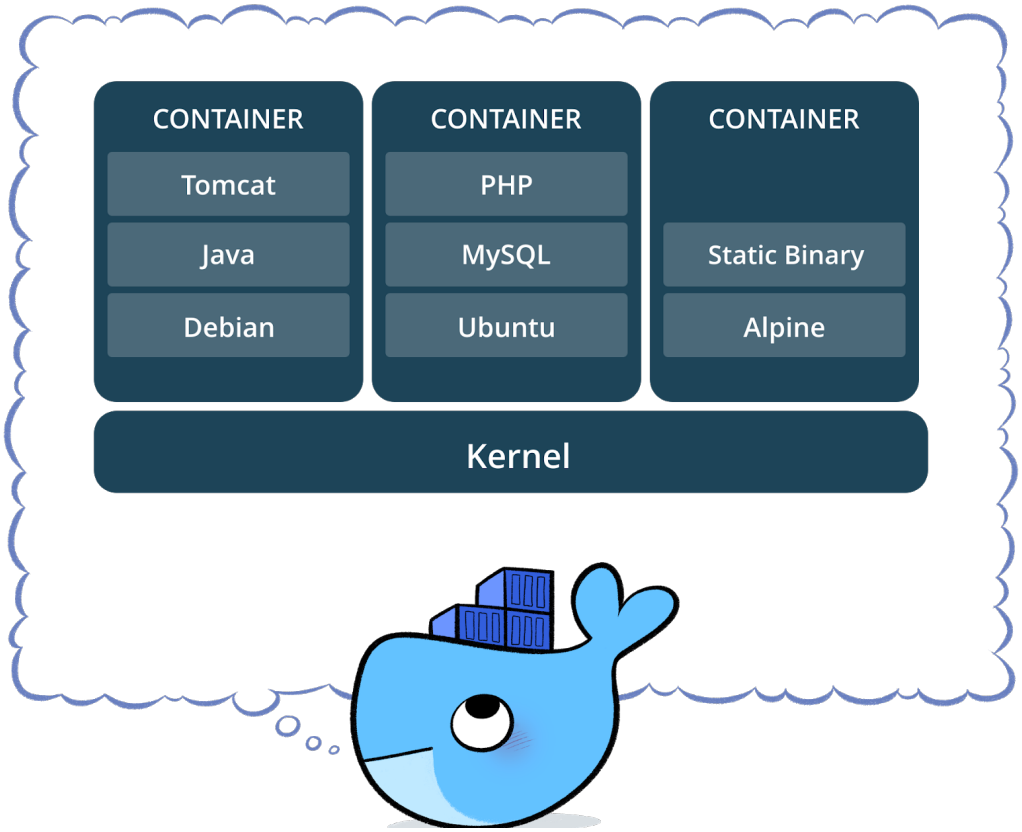
NetPyNE Tutorial

- NEURON vs NEURON GUI vs NEtPyNE vs NetPyNE UI
- Intro to netpyne – motivation + structure
- Simple tutorial
  - 2 populations
  - 1 cell rule (soma+dend)
  - 1 synMech
  - Iclamp -> S
  - 1 conn rule – probability – S->M
  - Basic plots
- Biophys tutorial
  - Build on prev tutorial?
  - Import cell params from ModelDB – Hay 2011 and inhibitory cell
  - 2 layers (L23, L4), each with exc+inhib pop
  - 2 synMechs
  - NetStim -> all
  - Conn rules – yfrac-based weight; dist-dep inh conns
- Import model/tutorial – purkinje?

NetPyNE GUI Tutorial: Installation

Instructions: https://github.com/MetaCell/NetPyNE-UI/blob/master/README.md

1) Install **Docker:** an application that allows you to run “containers” with everything you need already pre-installed.



NetPyNE GUI Tutorial: Installation

Instructions: https://github.com/MetaCell/NetPyNE-UI/blob/master/README.md

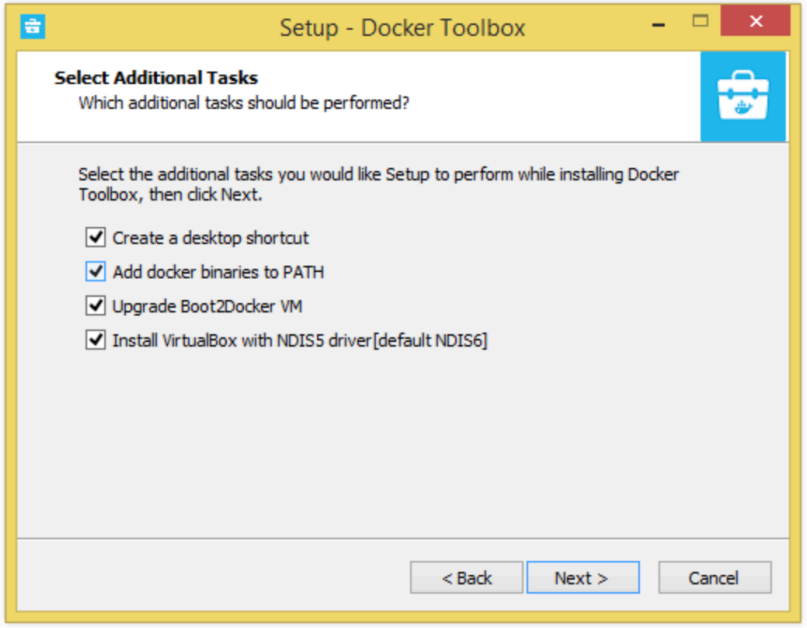
1) Install **Docker** (Windows)

For Windows users

To install docker on Windows download docker-toolbox from the link below and follow the onscreen instructions. Windows

Few notes regarding the windows installation:

- During the step "Select Additional Tasks", select all the checkboxes as the image here.



NetPyNE GUI Tutorial: Installation

Instructions: https://github.com/MetaCell/NetPyNE-UI/blob/master/README.md

1) Install **Docker** (Linux and Mac)

For Linux users

To install docker on Linux follow the link below with the instructions. Linux

Few notes about the linux installation:

- Once the installation has been completed, to run docker from your user you need to add it to the docker group. To do this you can run the command below

```
sudo usermod -a -G docker $USERNAME
```

Once the command has been executed you need to open a new terminal or login in your environment again in order to make the change effective.

For MacOS users

To install docker on Mac OS follow the link below with the instructions. MacOS

NetPyNE GUI Tutorial: Installation

Instructions: https://github.com/MetaCell/NetPyNE-UI/blob/master/README.md

2) Get the NetPyNE GUI container:

Install NEURON-UI using Docker from command line

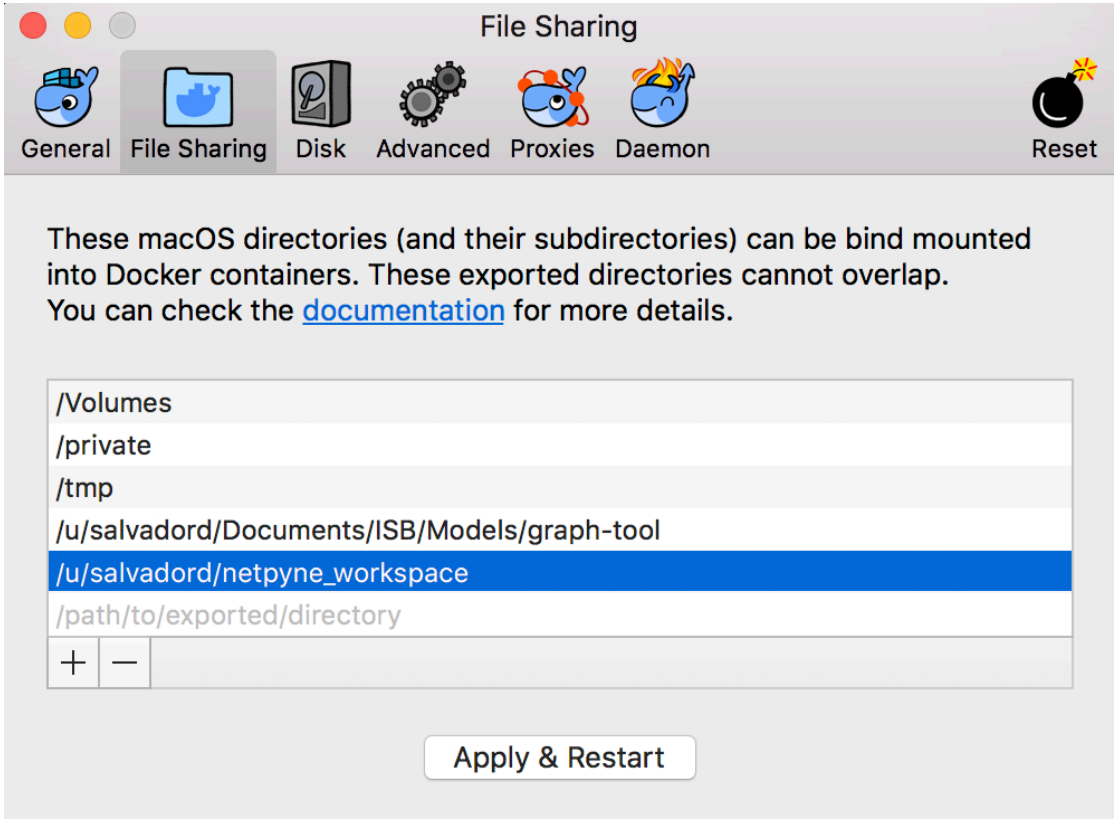
To pull the docker container:

```
docker pull metacell/netpyne-ui
```

NetPyNE GUI Tutorial: Installation

Instructions: https://github.com/MetaCell/NetPyNE-UI/blob/master/README.md

3) Add a shared folder via Docker Settings -> Preferences -> File Sharing



NetPyNE GUI Tutorial: Installation

Instructions: https://github.com/MetaCell/NetPyNE-UI/blob/master/README.md

4) Run the NetPyNE GUI container (Linux and Mac):

To run the docker container:

```
docker run -it -p 8888:8888 metacell/netpyne-ui
```

Or alternatively, if you want a local folder outside of the Docker container to host the NetPyNE-UI workspace (where you can import models from, export models to, inspect the log and the jupyter notebook) you can execute the following command:

```
docker run -it -v ~/folder_in_your_computer:/home/jovyan/netpyne_workspace -p 8888:8888 metacell/netpyne-ui
```

This will mount your local folder `folder_in_your_computer` as a volume inside the container and will be used to host the NetPyNE-UI workspace (`/home/jovyan/netpyne_workspace` inside the container).

Once you run your container you can open your browser and connect to http://localhost:8888/geppetto.

NetPyNE GUI Tutorial: Installation

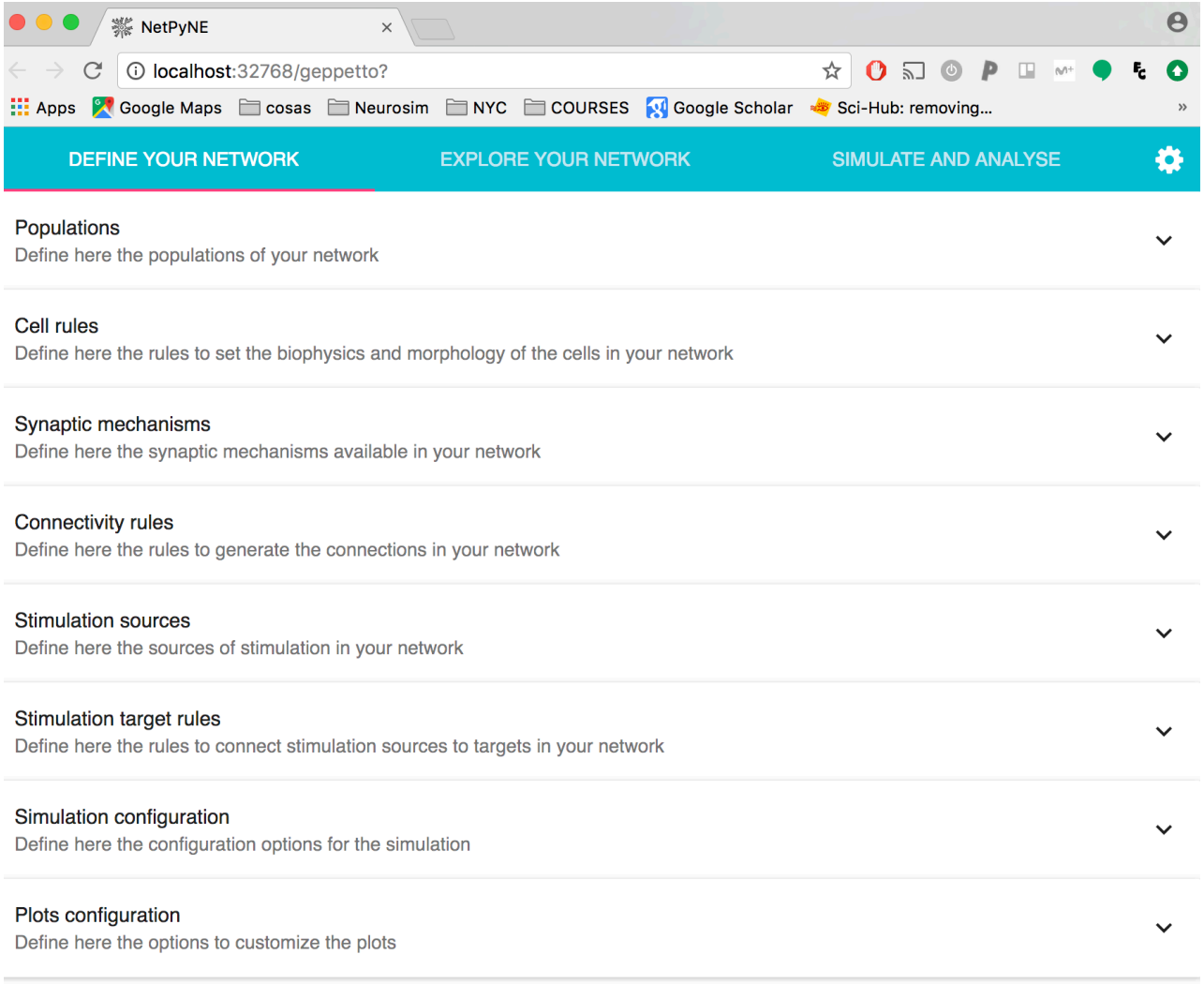
Instructions: https://github.com/MetaCell/NetPyNE-UI/blob/master/README.md

4) Run the NetPyNE GUI container (Windows):

One note for Windows users - once the docker image has been pulled and the NetPyNE-UI container created, you can reach your instance of NetPyNE-UI by default at the address 192.168.99.100:8888, if something has been changed in the settings this will be visible when the docker terminal will be open, as per image below.

NetPyNE GUI Tutorial: Installation

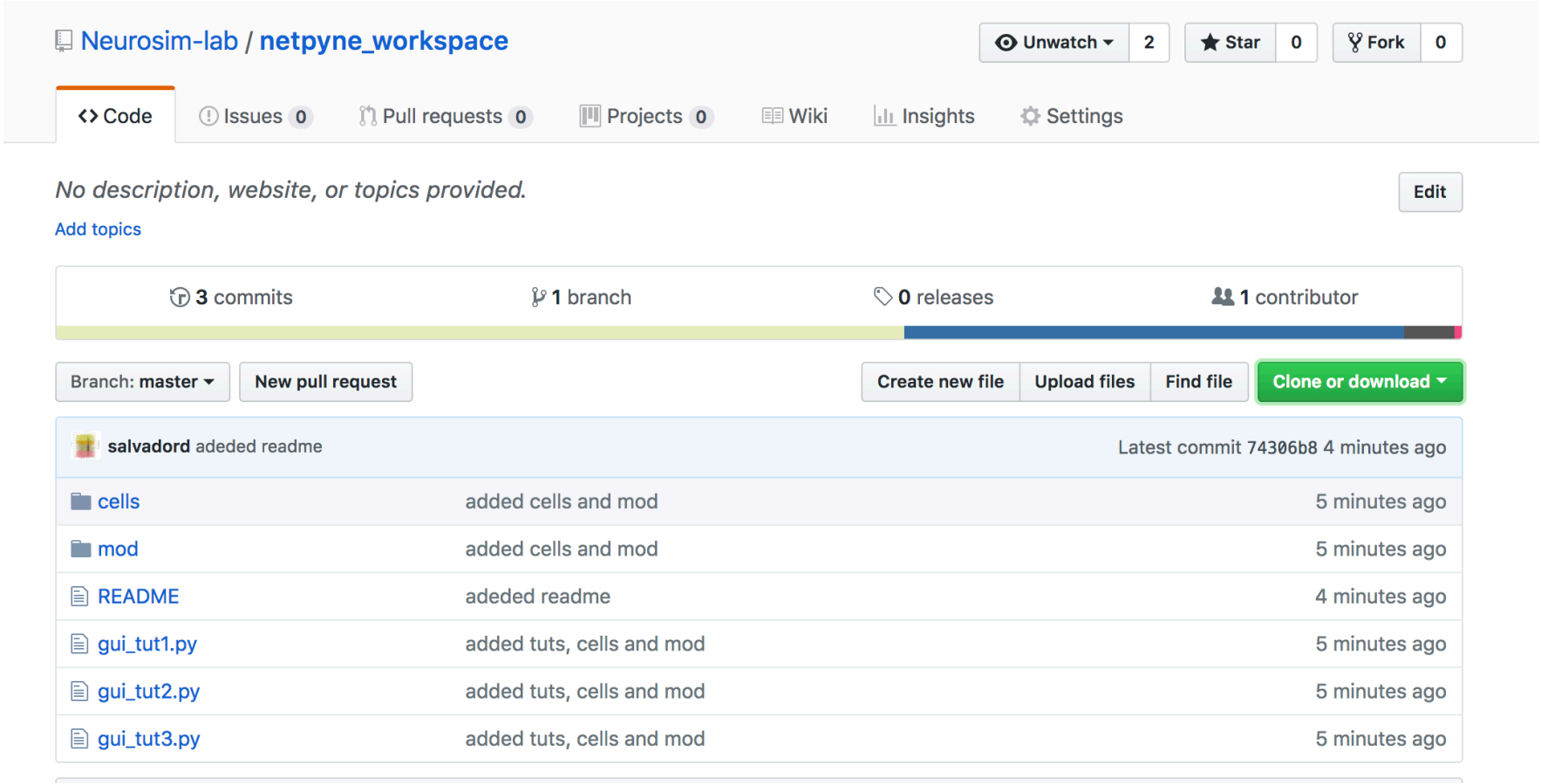
Instructions: https://github.com/MetaCell/NetPyNE-UI/blob/master/README.md



NetPyNE GUI Tutorial: Installation

Clone this Github repo with the workspace files:

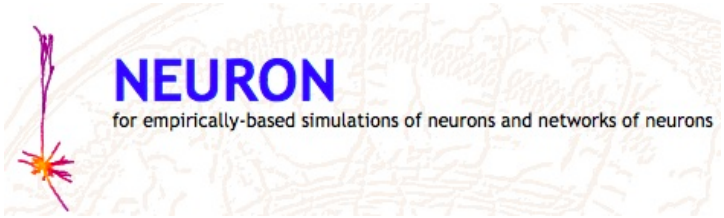
https://github.com/Neurosim-lab/netpyne_workspace.git

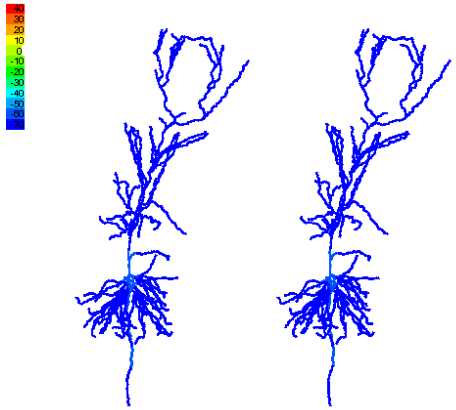


What is what…

NEURON

What is what…





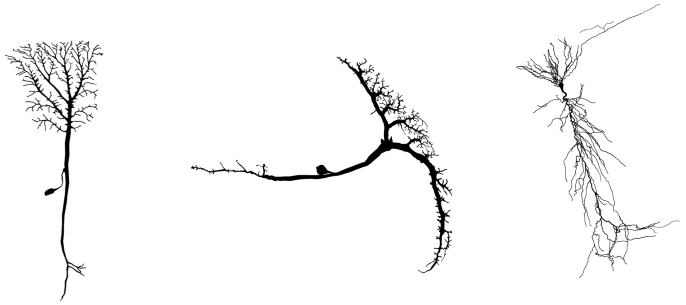
NEURON

Biophysically-detailed simulation at multiple scales:

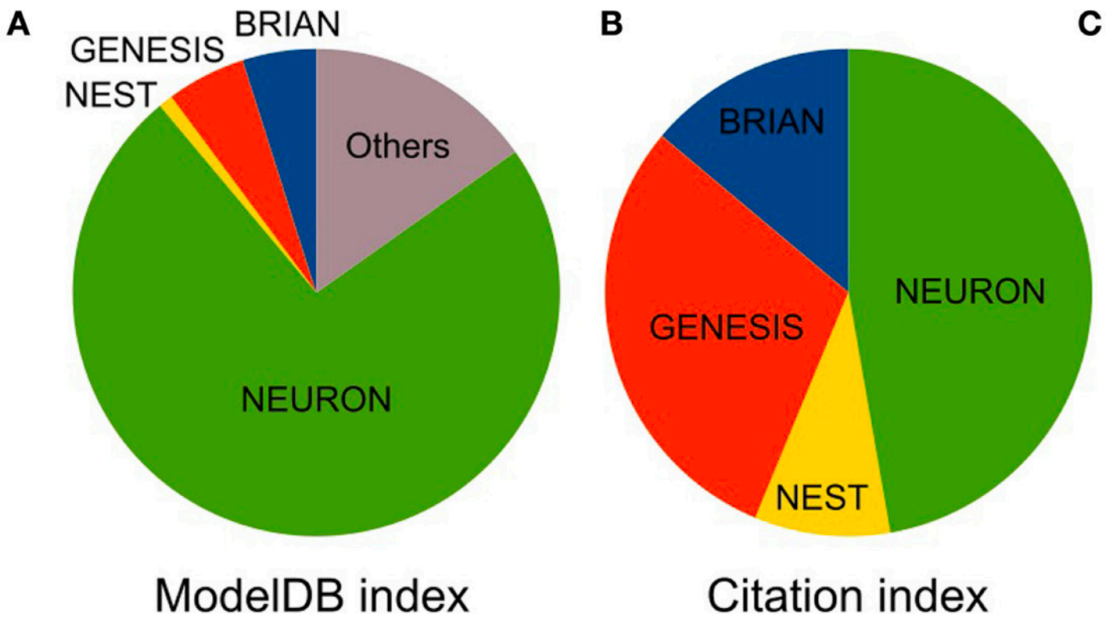
- molecular reaction and diffusion
- ionic channels and synapses
- detailed cell morphologies
- large-scale networks

What is what…

- Widely used (new papers each year)
- Many reusable components: channels, cells, …
- Comprehensive documentation and tutorials

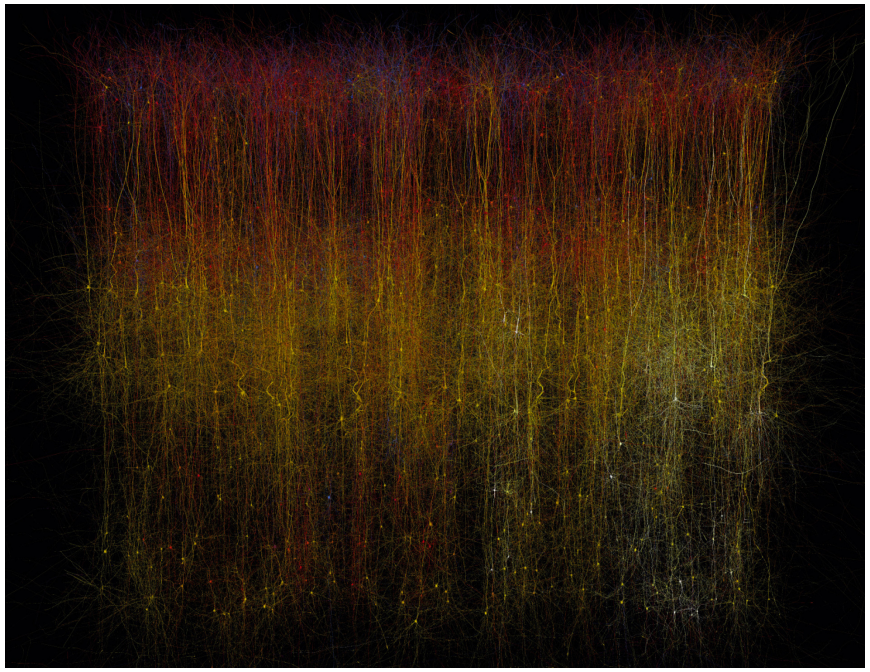


NEURON



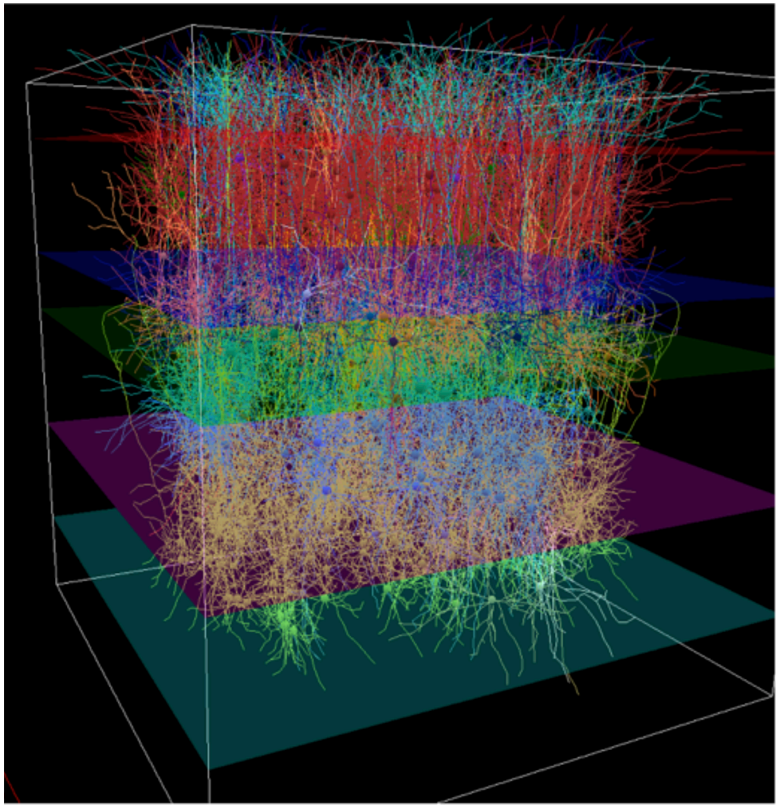
What is what…

- Used in major simulation projects



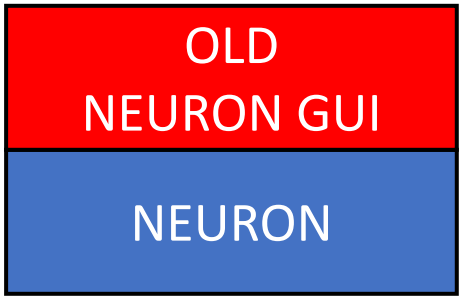
Human Brain Project
(Rat S1)

NEURON

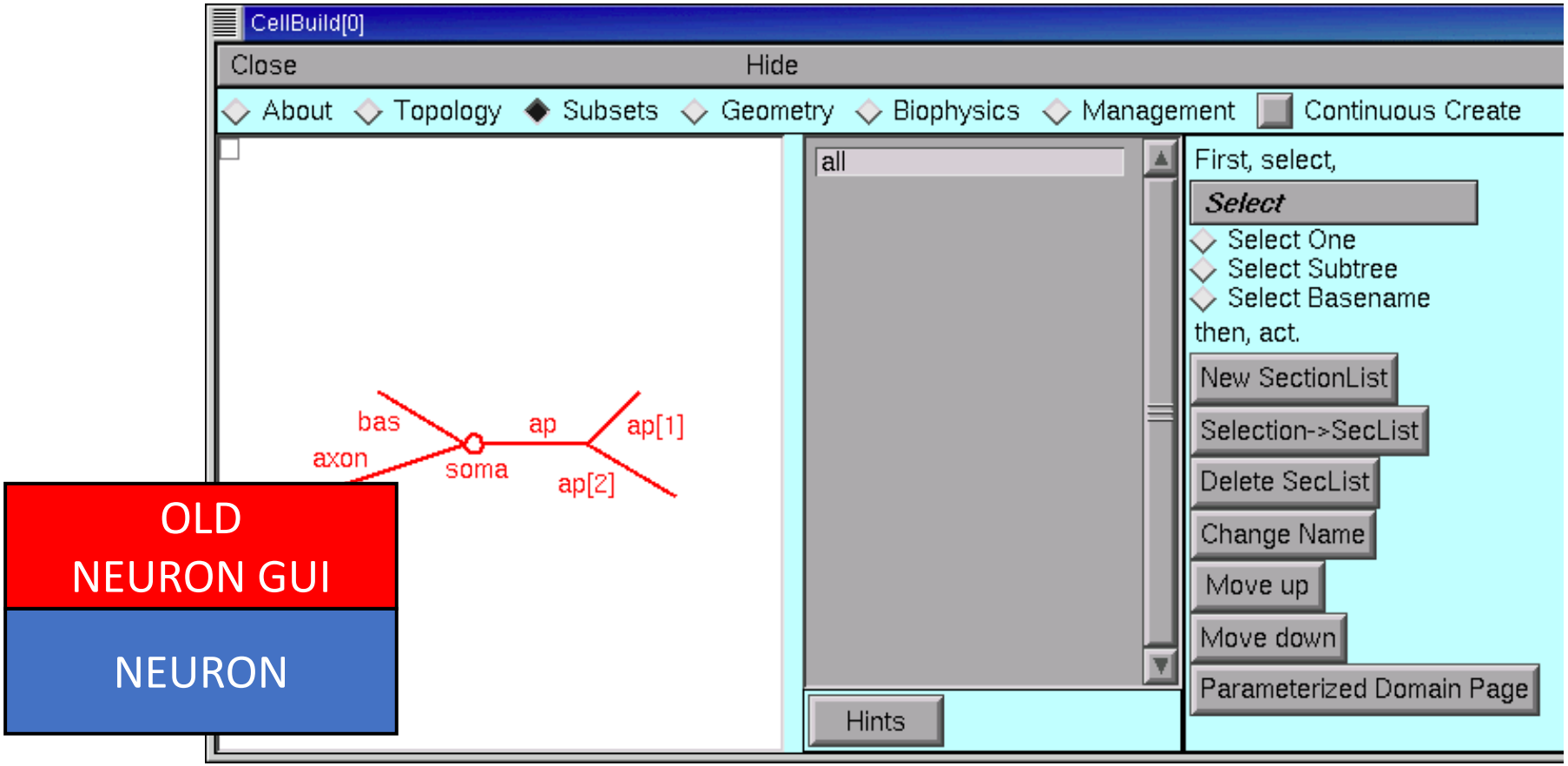


Allen Brain Institute
(Mouse V1)

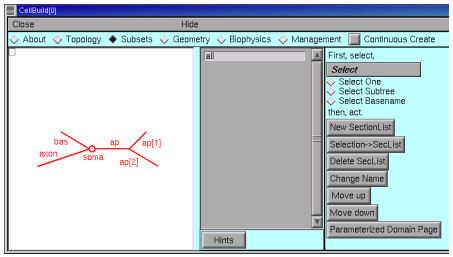
What is what…



What is what…


CellBuild[0]
Close
Hide
About
Topology
Subsets
Geometry
Biophysics
Management
Continuous Create
all
bas
axon
soma
ap
ap[1]
ap[2]
Hints
First, select,
Select
Select One
Select Subtree
Select Basename
then, act.
New SectionList
Selection->SecList
Delete SecList
Change Name
Move up
Move down
Parameterized Domain Page
OLD
NEURON GUI
NEURON

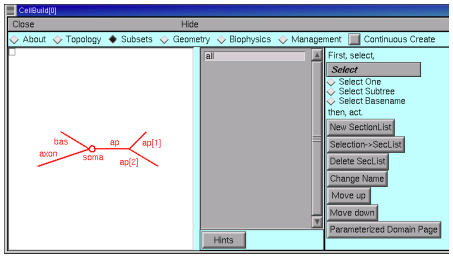
What is what…



OLD
NEURON GUI

NEURON

What is what…

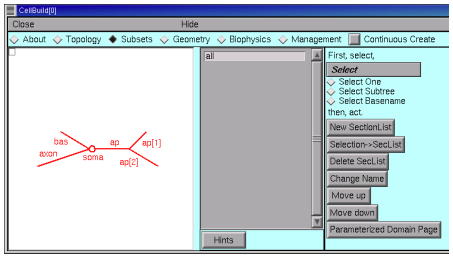


OLD
NEURON GUI

NEURON

NEURON

What is what…



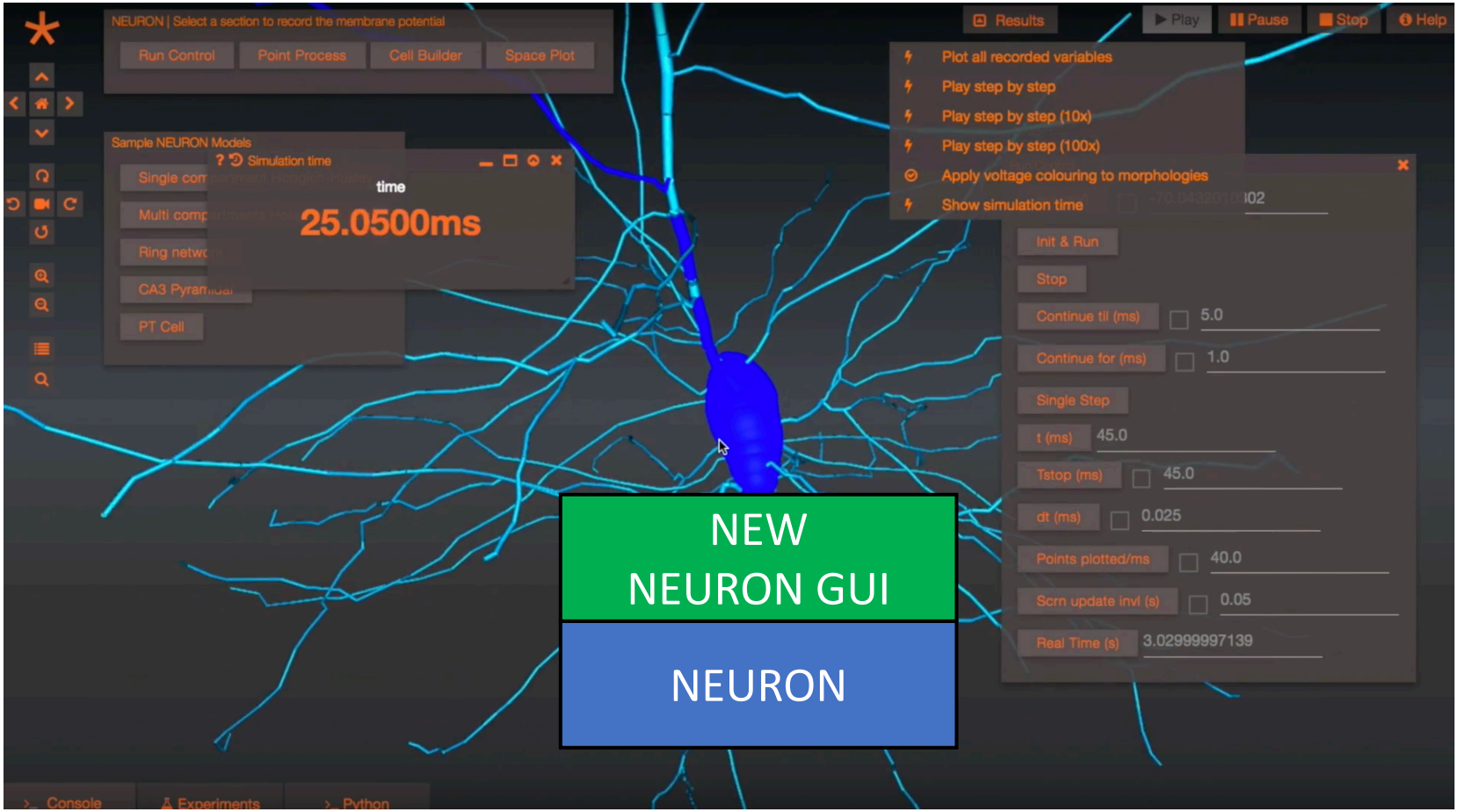
OLD
NEURON GUI

NEURON

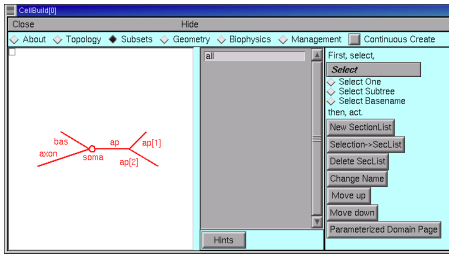
NEW
NEURON GUI

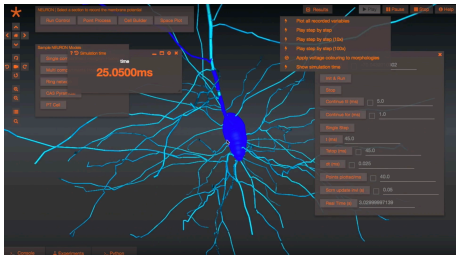
NEURON

What is what…


NEURON | Select a section to record the membrane potential
Run Control
Point Process
Cell Builder
Space Plot
Results
Play
Pause
Stop
Help
Plot all recorded variables
Play step by step
Play step by step (10x)
Play step by step (100x)
Apply voltage colouring to morphologies
Show simulation time
Sample NEURON Models
Simulation time
time
25.0500ms
PT Cell
Init & Run
Stop
Continue til (ms)
5.0
Continue for (ms)
1.0
Single Step
t (ms)
45.0
Tstop (ms)
45.0
dt (ms)
0.025
Points plotted/ms
40.0
Scrn update invl (s)
0.05
Real Time (s)
3.02999997139
Console
Experiments
Python
NEW
NEURON GUI
NEURON

What is what…





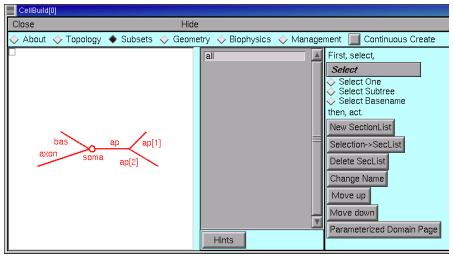
OLD
NEURON GUI

NEURON

NEW
NEURON GUI

NEURON

What is what…



OLD
NEURON GUI

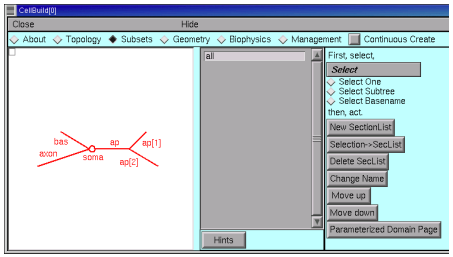
NEURON

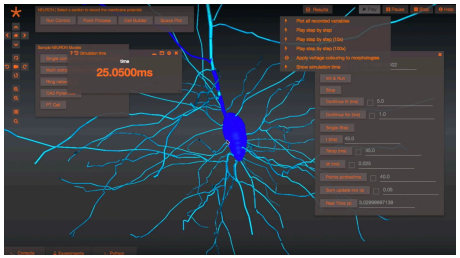
NEW
NEURON GUI

NEURON

NEURON

What is what…





OLD
NEURON GUI

NEURON

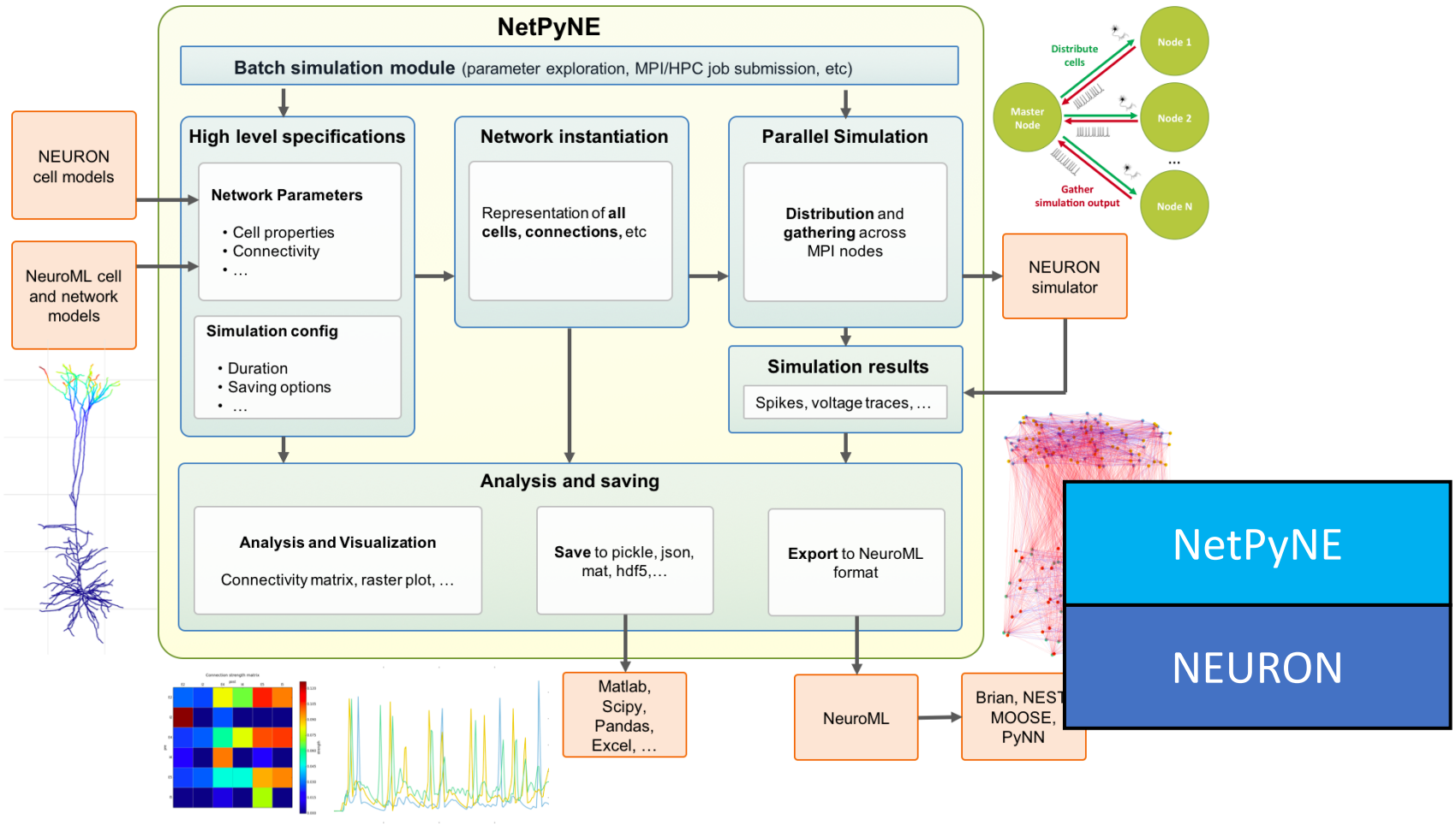
NEW
NEURON GUI

NEURON

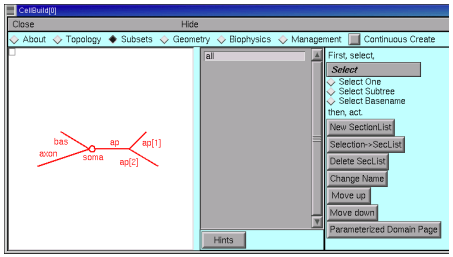
NetPyNE

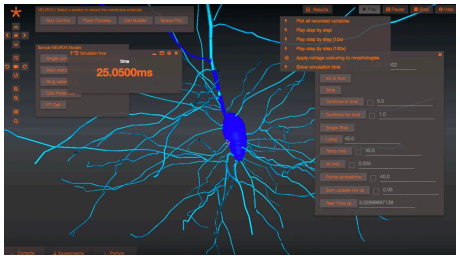
NEURON

What is what…


NetPyNE
Batch simulation module (parameter exploration, MPI/HPC job submission, etc)
NEURON cell models
NeuroML cell and network models
High level specifications
Network Parameters
• Cell properties
• Connectivity
• …
Simulation config
• Duration
• Saving options
• …
Network instantiation
Representation of all cells, connections, etc
Parallel Simulation
Distribution and gathering across MPI nodes
Distribute cells
Master Node
Node 1
Node 2
Node N
Gather simulation output
NEURON simulator
Simulation results
Spikes, voltage traces, …
Analysis and saving
Analysis and Visualization
Connectivity matrix, raster plot, …
Save to pickle, json, mat, hdf5,…
Export to NeuroML format
Matlab, Scipy, Pandas, Excel, …
NeuroML
Brian, NEST, MOOSE, PyNN
NetPyNE
NEURON

What is what…





OLD
NEURON GUI

NEURON

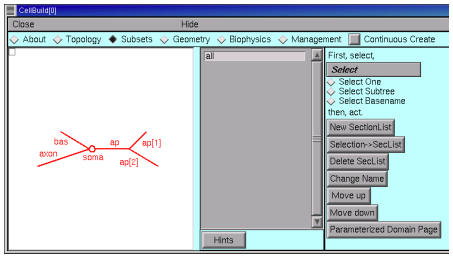
NEW
NEURON GUI

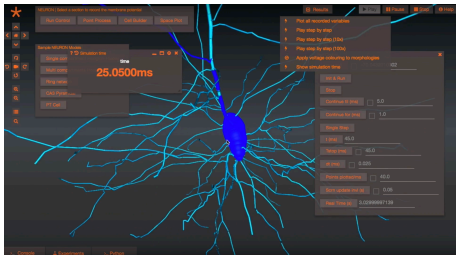
NEURON

NetPyNE

NEURON

What is what…





OLD
NEURON GUI

NEURON

NEW
NEURON GUI

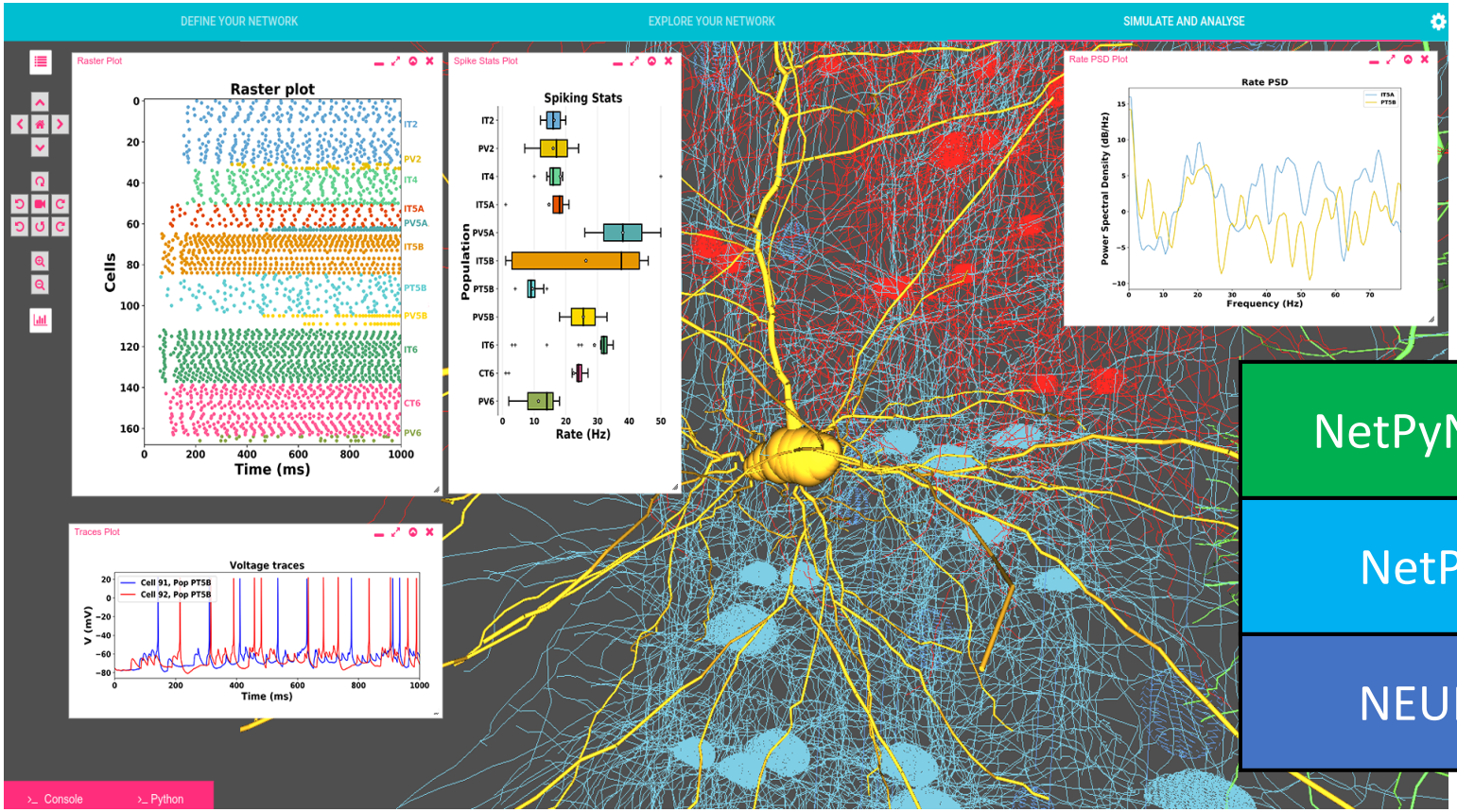
NEURON

NetPyNE GUI

NetPyNE

NEURON

What is what…

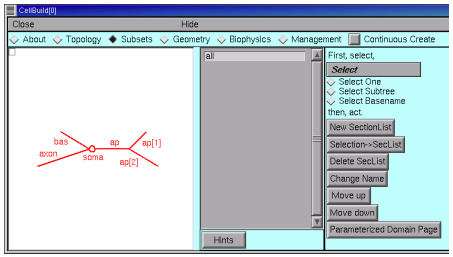


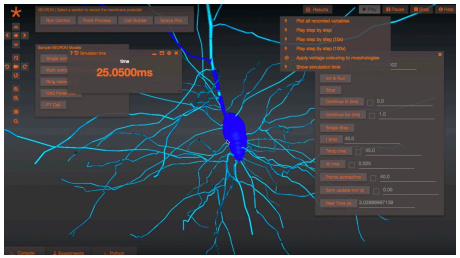
NetPyNE GUI

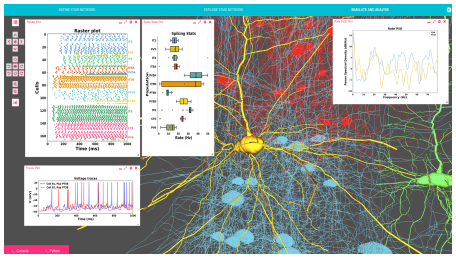
NetPyNE

NEURON

What is what…







OLD
NEURON GUI

NEURON

NEW
NEURON GUI

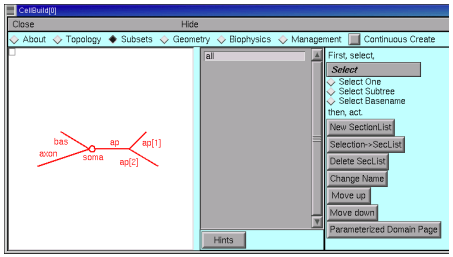
NEURON

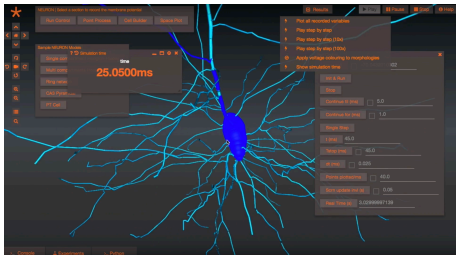
NetPyNE GUI

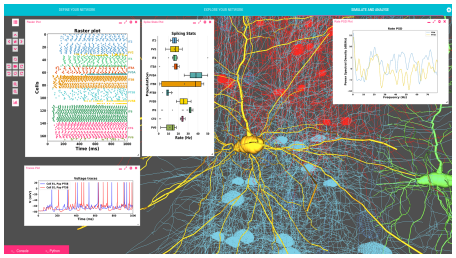
NetPyNE

NEURON

New NEURON GUI







OLD
NEURON GUI

NEURON

NEW
NEURON GUI

NEURON

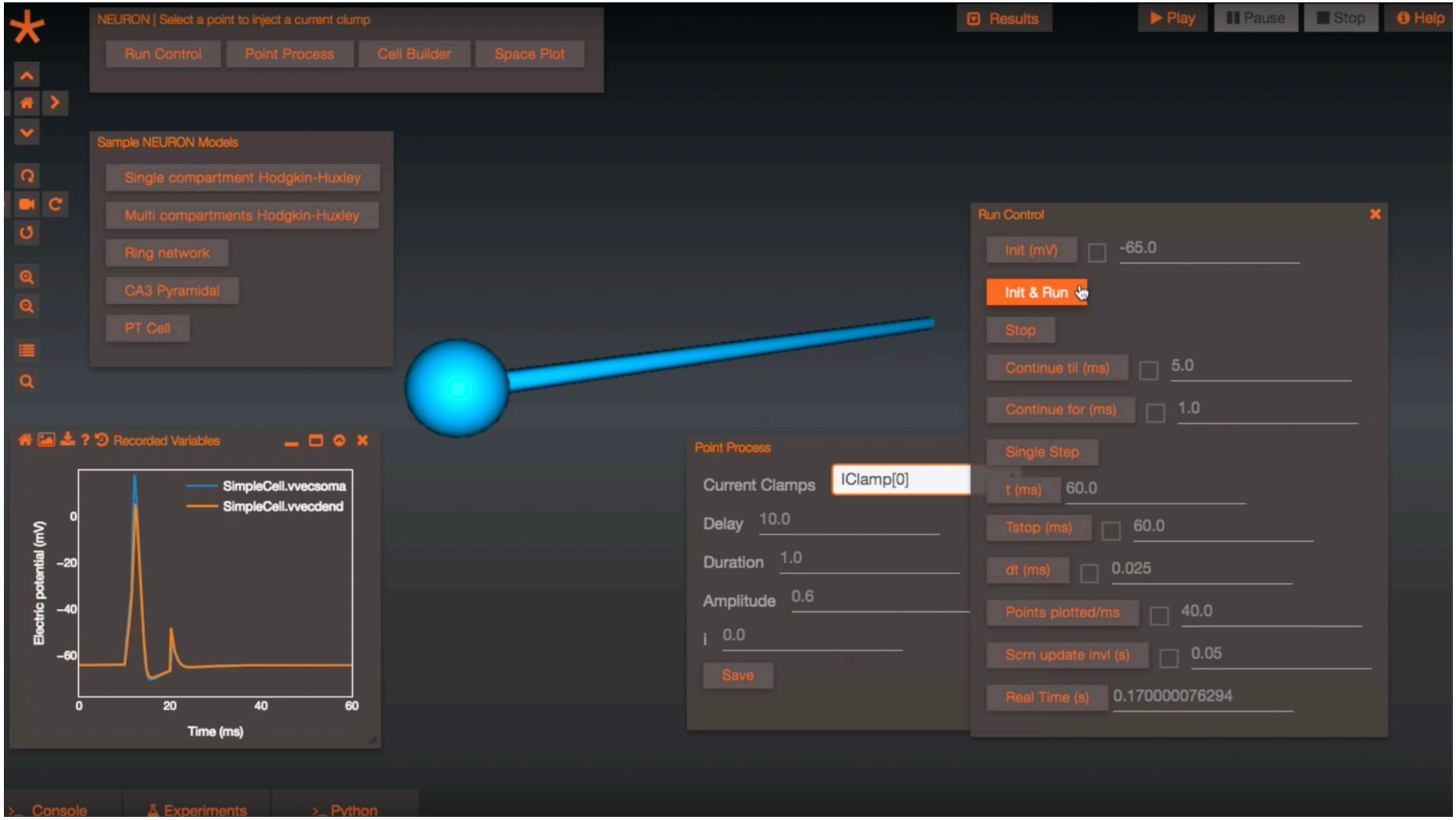
NetPyNE GUI

NetPyNE

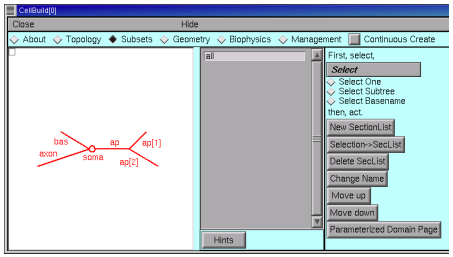
NEURON

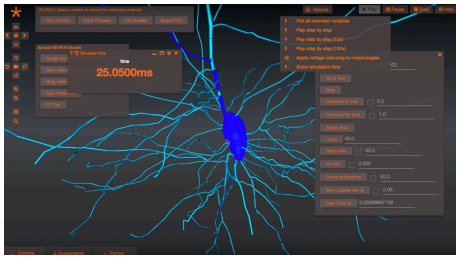
New NEURON GUI

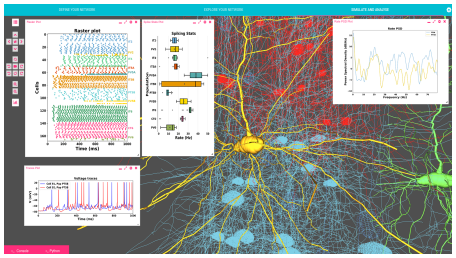
https://www.youtube.com/watch?v=CjoA3lTa25I



NetPyNE







OLD
NEURON GUI

NEURON

NEW
NEURON GUI

NEURON

NetPyNE GUI

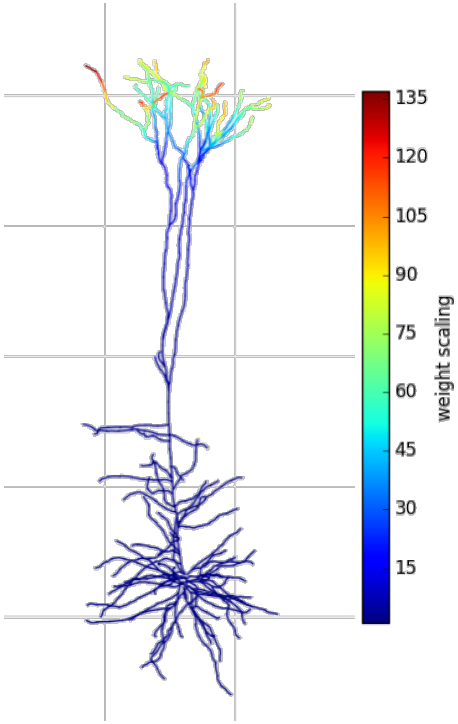
NetPyNE

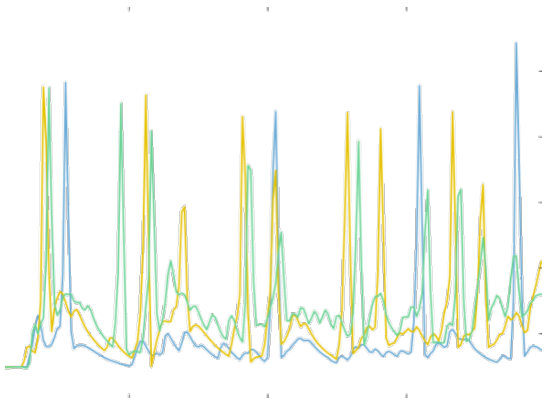
NEURON

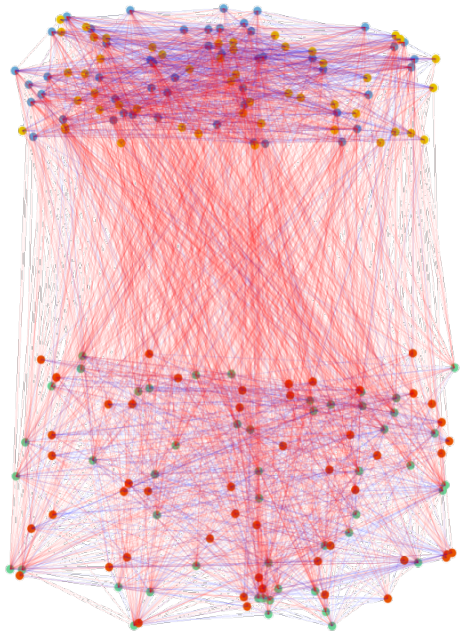
netPyne

A Python package to facilitate the development, simulation and analysis of biological neuronal networks in NEURON

www.netpyne.org

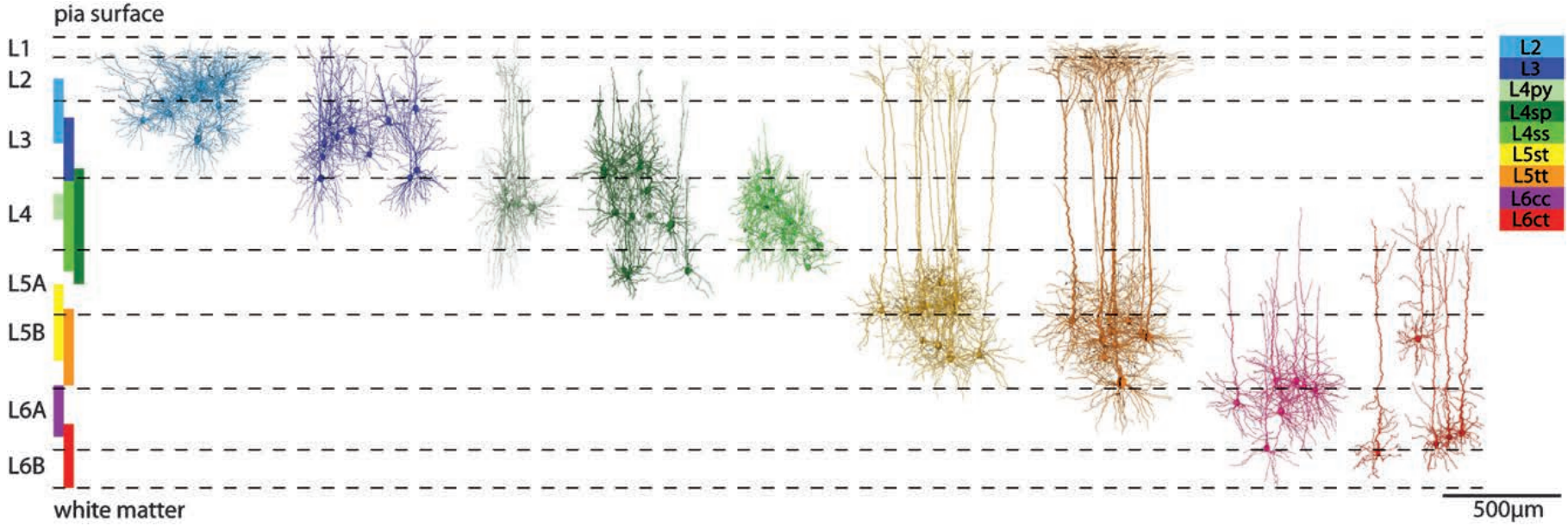






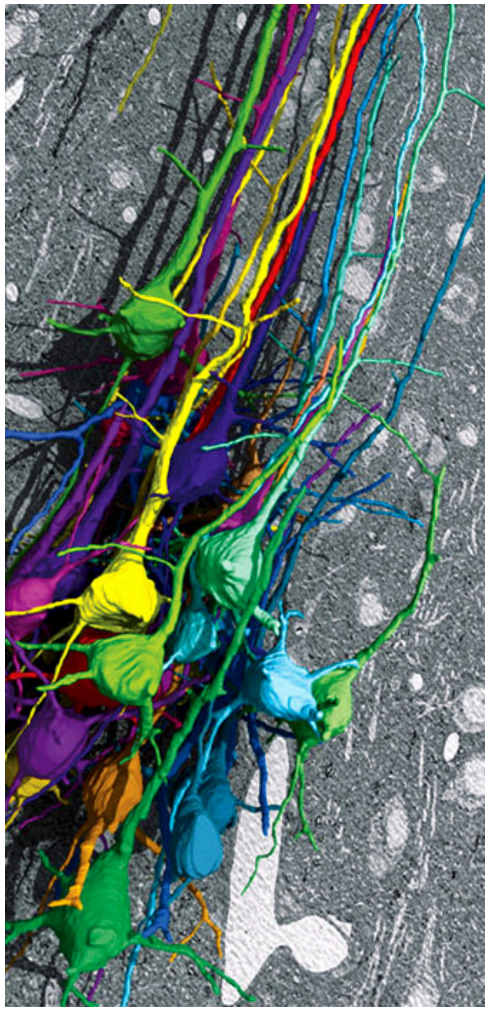
NetPyNE: Motivation

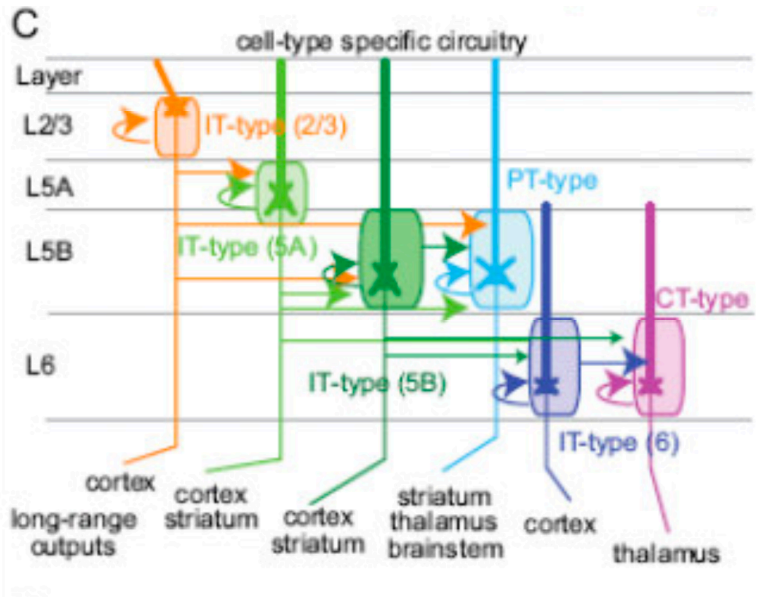
- Facilitate incorporation of experimental data at multiple scales



NetPyNE: Motivation

- Facilitate incorporation of experimental data at multiple scales

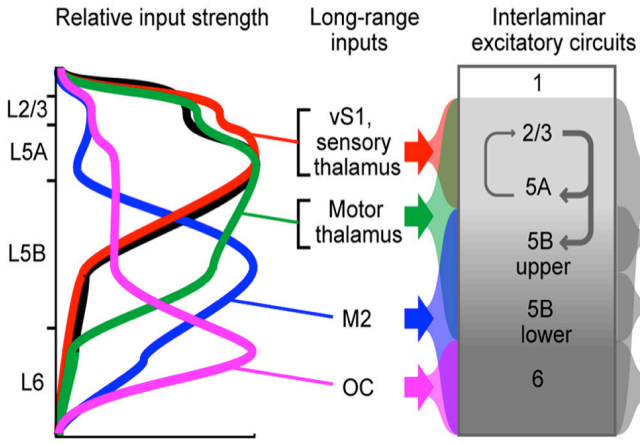




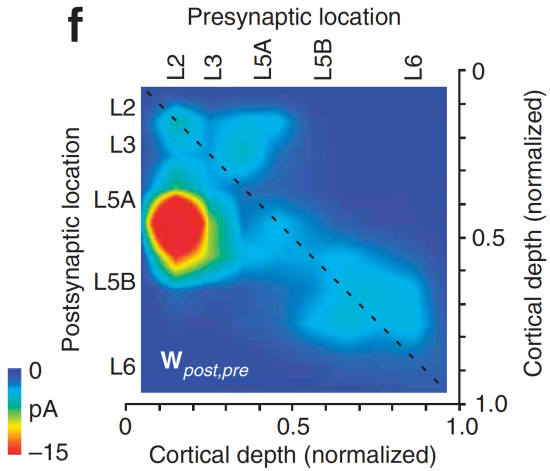
NetPyNE: Motivation

- Facilitate incorporation of experimental data at multiple scales

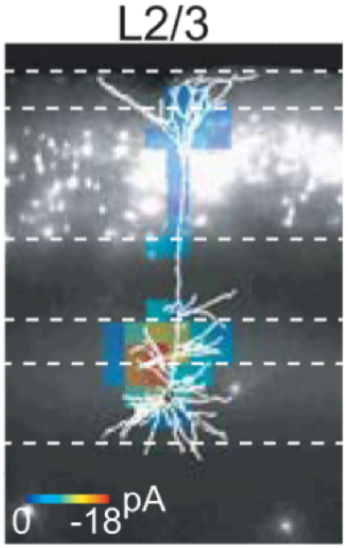
Long-range inputs



Local microcircuits



Dendritic inputs

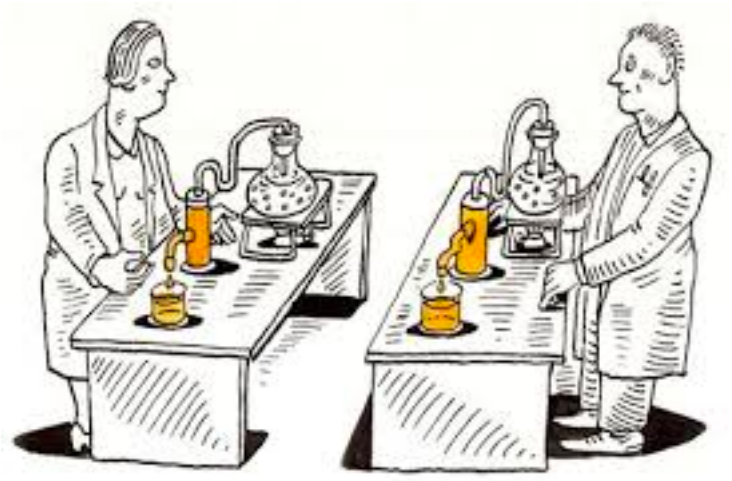


NetPyNE: Motivation

- Separate model parameters from implementation
- Standardize format – easy to read, interpret, edit, share etc

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

```
for cellParams in range(pop['numCells']):
            cell = sim.Cell(cellParams)
            cell.tags['y'] = numpy.random(100,400)
            cell.tags['cellType'] = 'PYR'
```

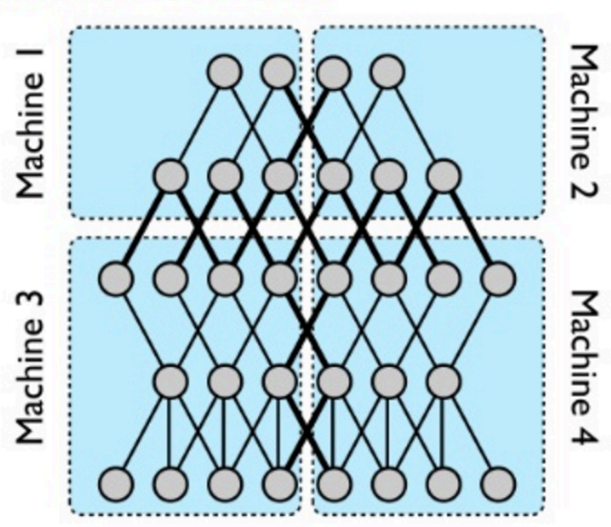


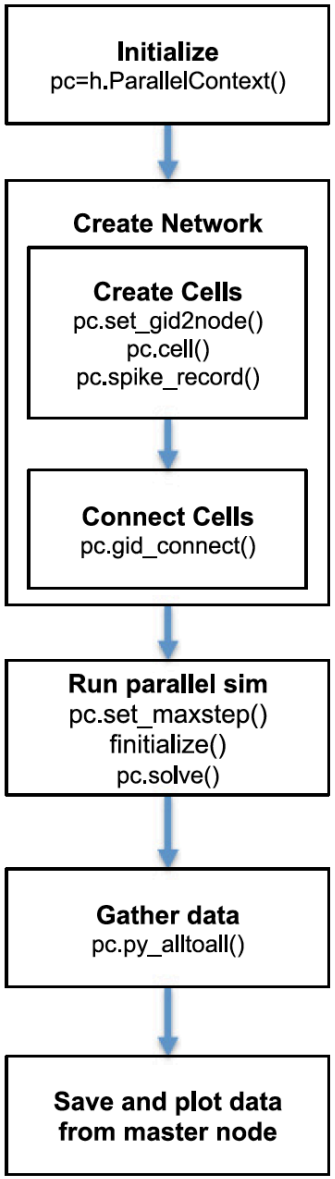
Replicate: get same thing to run again

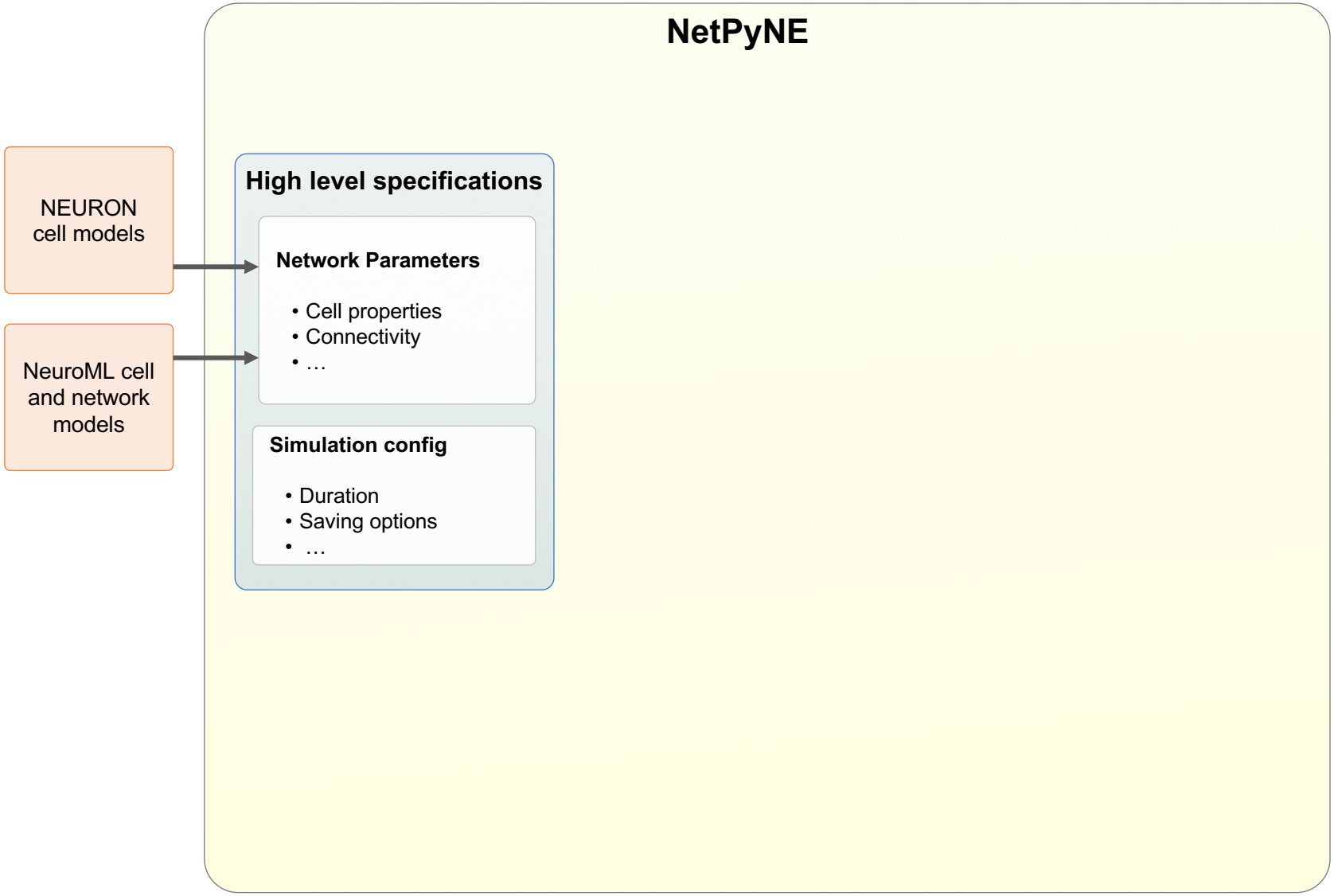
Reproduce: make it youself

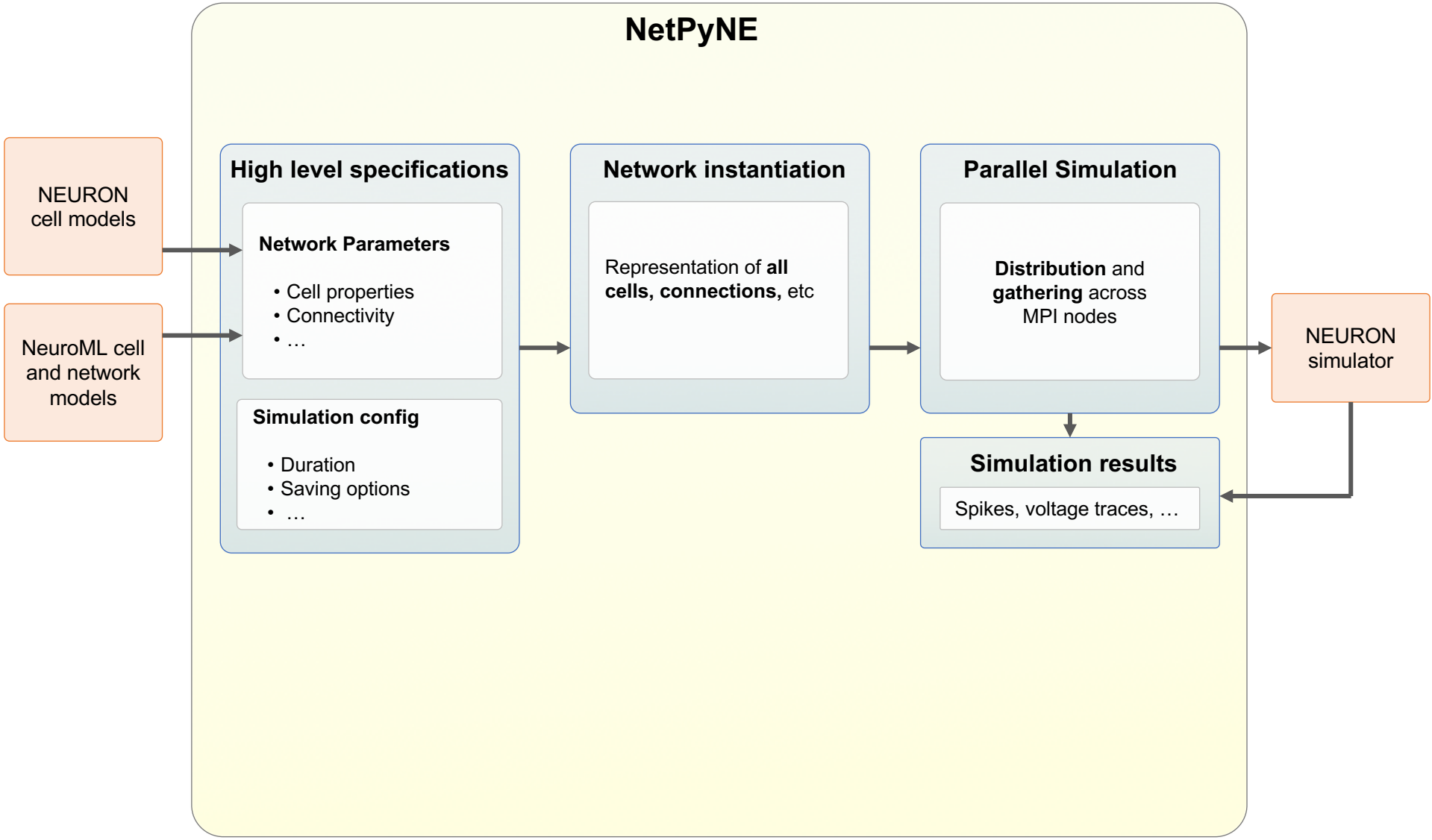
NetPyNE: Motivation

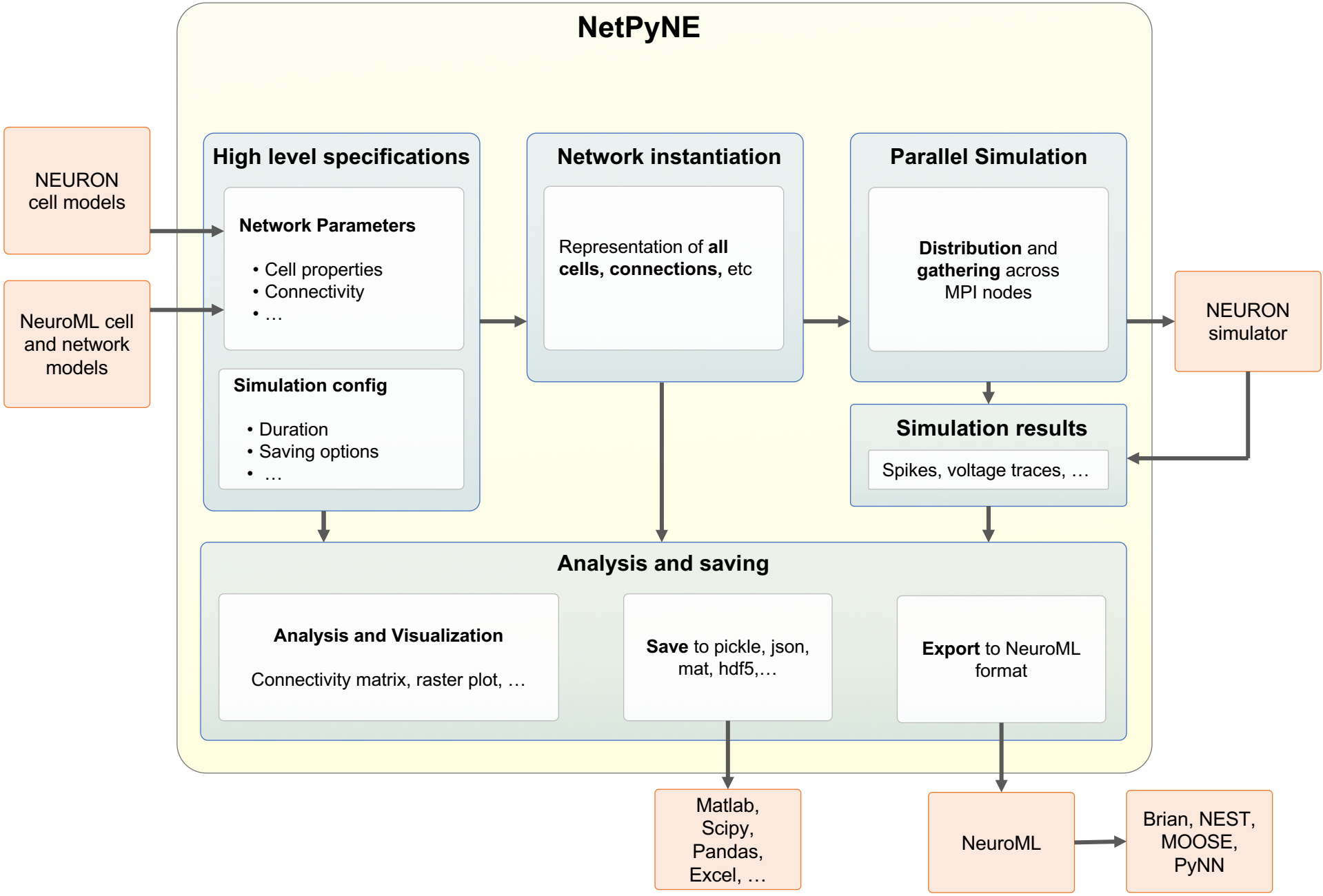
- Facilitate model parallelization (HPCs)
- Batch parameter exploration/optimization

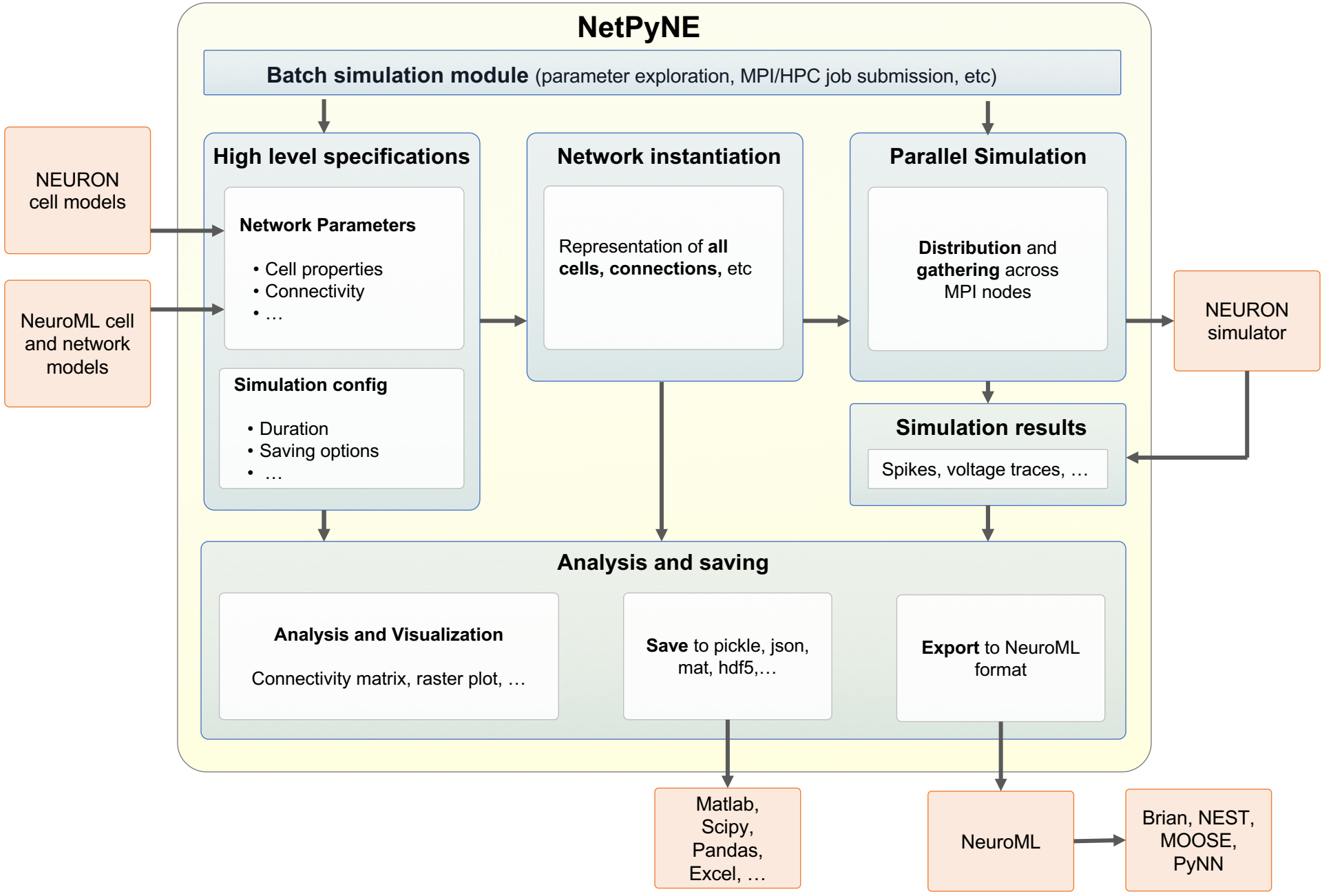





NetPyNE
NEURON cell models
NeuroML cell and network models
High level specifications
Network Parameters
• Cell properties
• Connectivity
• …
Simulation config
• Duration
• Saving options
• …

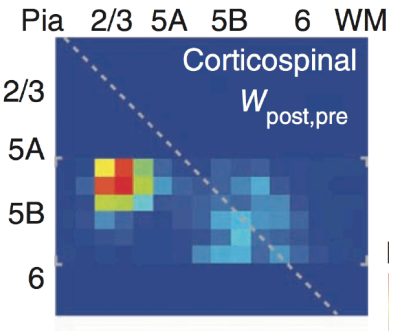

NetPyNE
NEURON cell models
NeuroML cell and network models
High level specifications
Network Parameters
• Cell properties
• Connectivity
• …
Simulation config
• Duration
• Saving options
• …
Network instantiation
Representation of all cells, connections, etc
Parallel Simulation
Distribution and gathering across MPI nodes
Simulation results
Spikes, voltage traces, …
NEURON simulator


NetPyNE
NEURON cell models
NeuroML cell and network models
High level specifications
Network Parameters
• Cell properties
• Connectivity
• …
Simulation config
• Duration
• Saving options
• …
Network instantiation
Representation of all cells, connections, etc
Parallel Simulation
Distribution and gathering across MPI nodes
NEURON simulator
Simulation results
Spikes, voltage traces, …
Analysis and saving
Analysis and Visualization
Connectivity matrix, raster plot, …
Save to pickle, json, mat, hdf5,…
Export to NeuroML format
Matlab, Scipy, Pandas, Excel, …
NeuroML
Brian, NEST, MOOSE, PyNN


NetPyNE
Batch simulation module (parameter exploration, MPI/HPC job submission, etc)
NEURON cell models
NeuroML cell and network models
High level specifications
Network Parameters
• Cell properties
• Connectivity
• …
Simulation config
• Duration
• Saving options
• …
Network instantiation
Representation of all cells, connections, etc
Parallel Simulation
Distribution and gathering across MPI nodes
NEURON simulator
Simulation results
Spikes, voltage traces, …
Analysis and saving
Analysis and Visualization
Connectivity matrix, raster plot, …
Save to pickle, json, mat, hdf5,…
Export to NeuroML format
Matlab, Scipy, Pandas, Excel, …
NeuroML
Brian, NEST, MOOSE, PyNN

NetPyNE: High level specifications

- Specifications are provided in **a standardized, declarative** Python format (JSON-like, lists and dicts).
- Clear **separation** of parameters from implementation code.
- Error **checking** and **suggestions** to facilitate model definition.

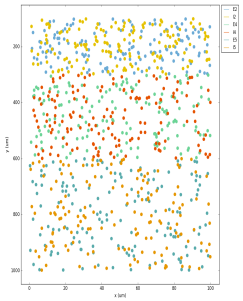



```
connParams['bin3->CSP'] = {
 'preConds':     {'y': [100, 150]},
 'postConds':    {'pop': 'CSP'},
 'probability': 0.15,
 'weight':       0.4,
 'delay':        5,
 'synMech':      'AMPA'}
```

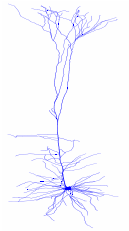
NetPyNE facilitates building models based on experimental data

NetPyNE: High level specifications

- User can define:
  - ***Populations**:* cell type, number of neurons or density, spatial extent, ...



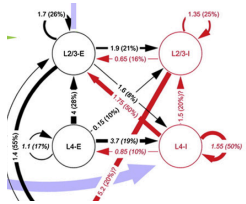
  - ***Cell properties:*** Morphology, biophysics, implementation, …



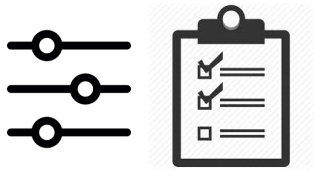
  - ***Synaptic mechanisms:*** Time constants, reversal potential, implementation, …

  - ***Stimulation:*** Spike generators, current clamps, spatiotemporal properties, …

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

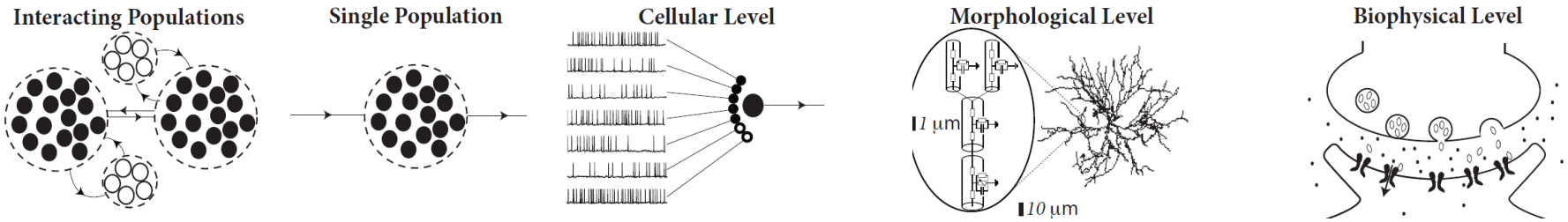


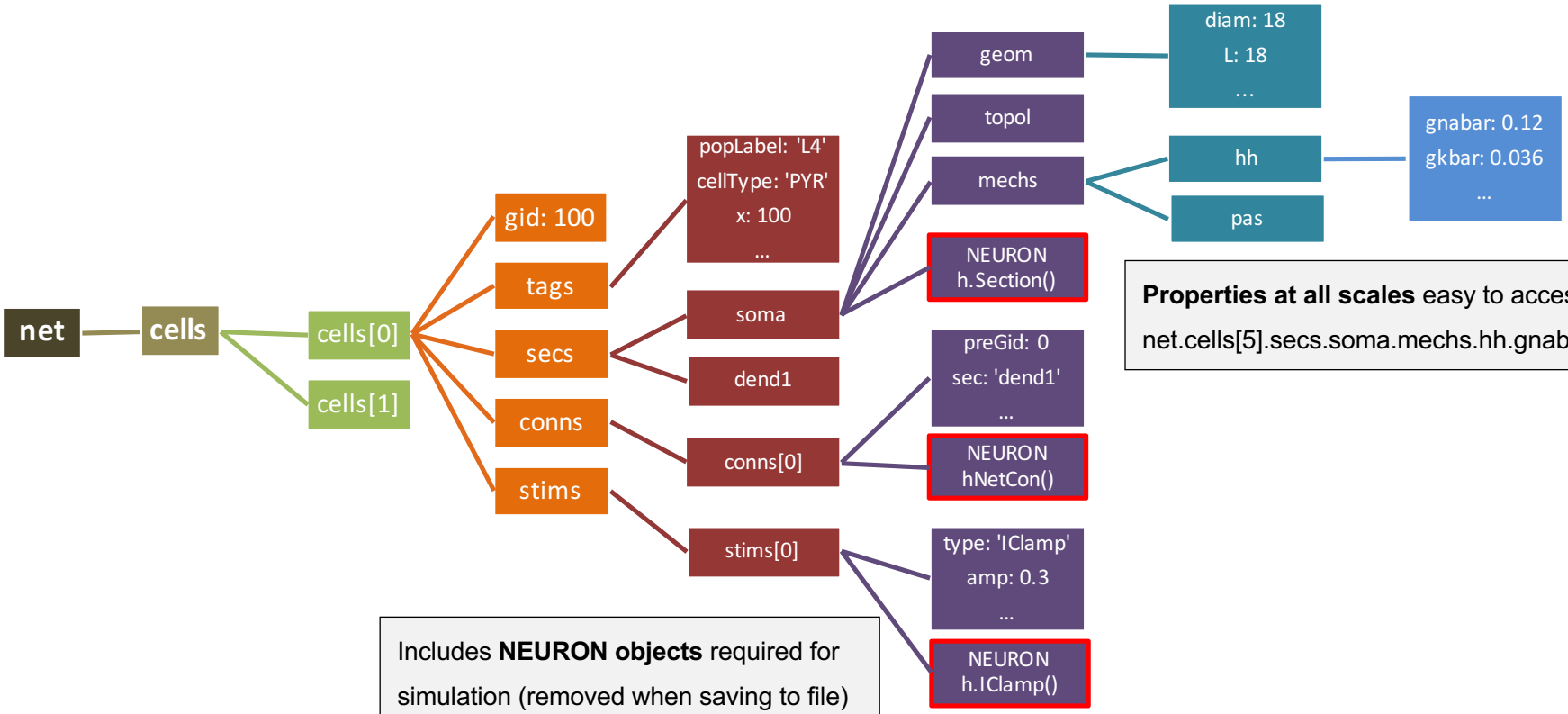
  - ***Simulation configuration:*** duration, saving and analysis, graphical output, ...



NetPyNE: Network Instantiation

- Network is created as Python-based **standardized hierarchical data structure.**

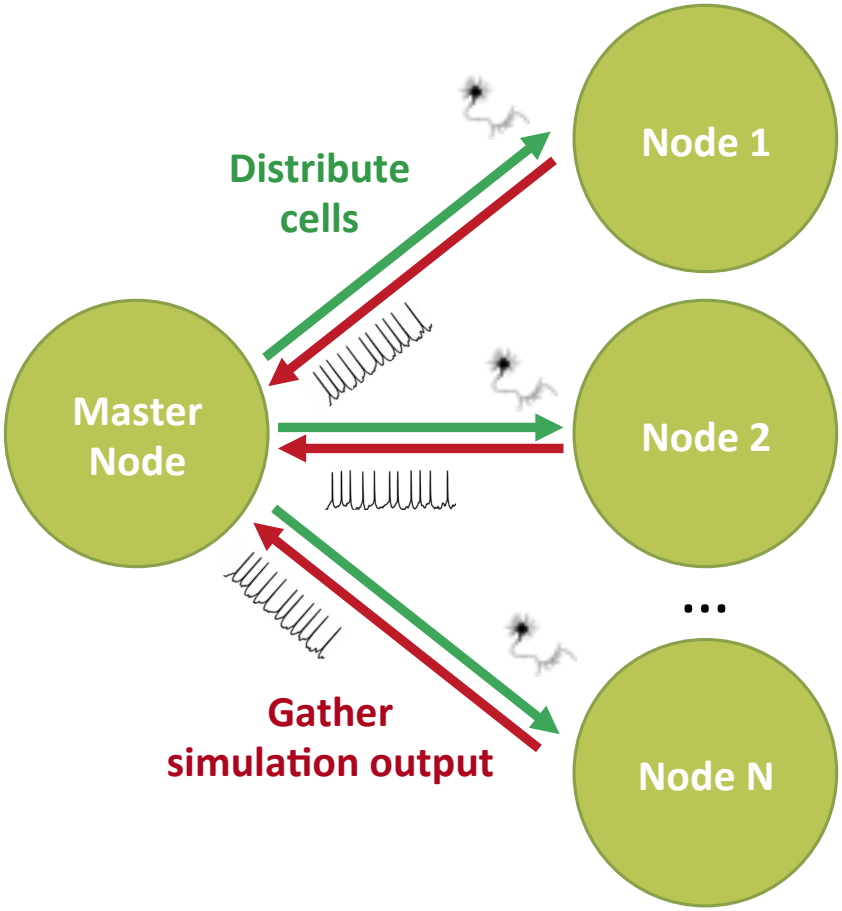




NetPyNE: Parallel Simulation

- Set up for MPI **parallel simulation** across multiple nodes (via NEURON simulator).

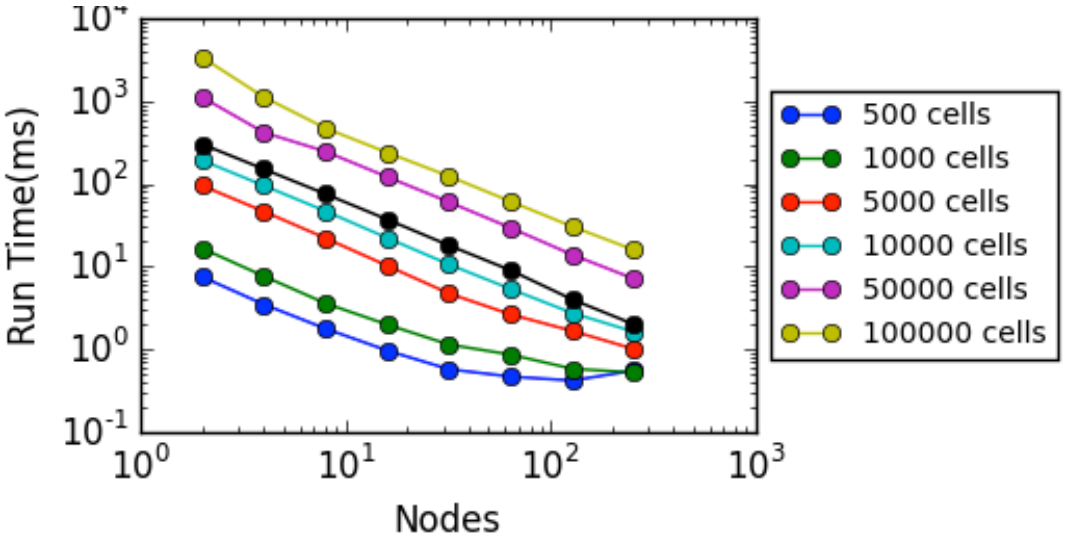
- Takes care of balanced **distribution** of cells and **gathering** of simulation output from nodes.



NetPyNE: Parallel simulation

- NetPyNE available on **the Neuroscience Gateway (NSG)** supercomputing platform.

NSG NEUROSCIENCE GATEWAY
A Portal for Computational Neuroscience



Simulation **run time** as a function of number of cells and number of nodes *(Neural Comput, 2016)*.

Results obtained using **NetPyNE on NSG**.

NetPyNE: Batch parallel simulations

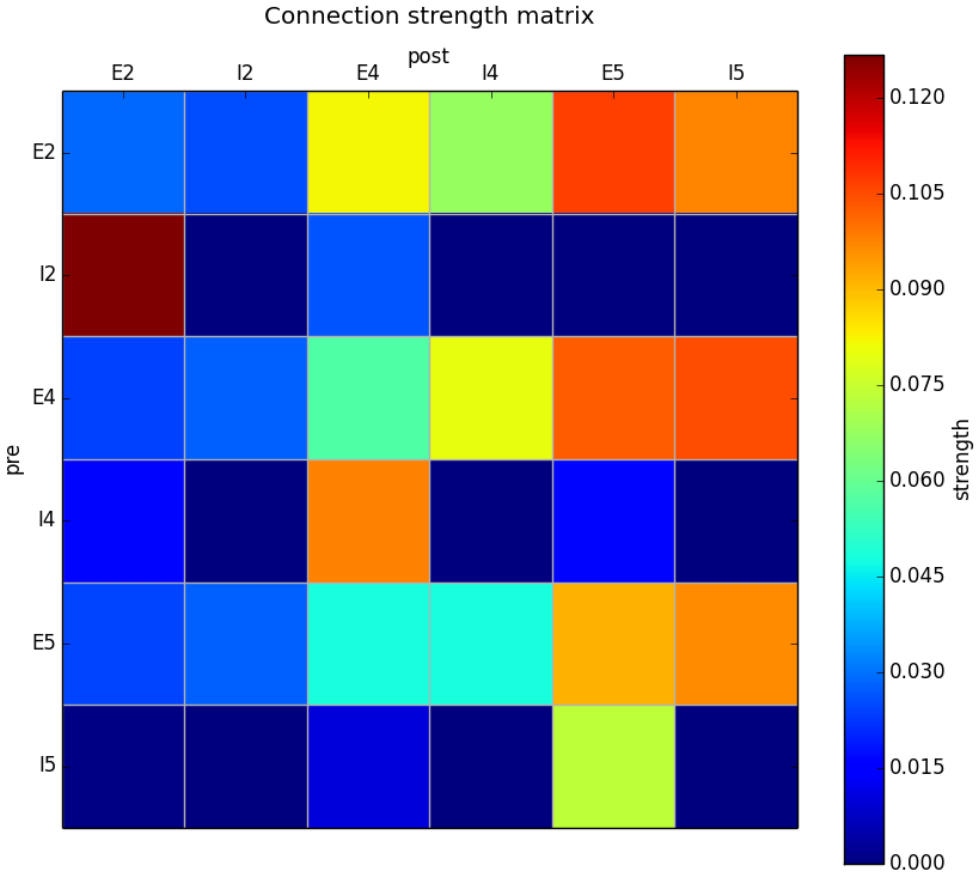
- **Easy specification** of parameters and range of values to explore in batch simulations.

- **Pre-defined, configurable** setups to automatically **submit jobs** in multicore machines (Bulletin board) or supercomputers (SLURM or PBS Torque)

SDSC SAN DIEGO SUPERCOMPUTER CENTER

NetPyNE: Analysis

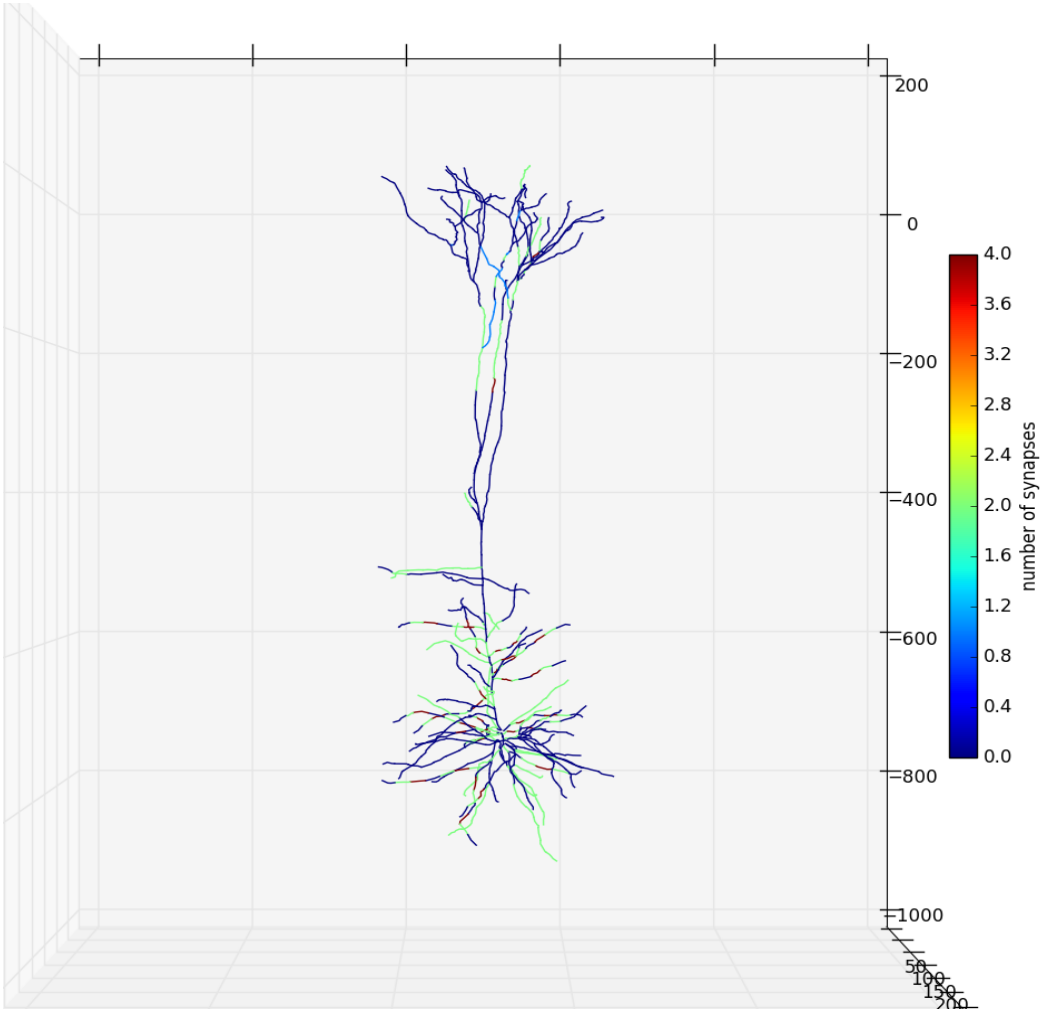
- **Connectivity matrix** at cell or population level (weights, num connections, probability,...)




```
plotConn(include = ['allCells'], feature='strength',
groupBy='pop', figSize=(9,9), showFig=True)
```

NetPyNE: Analysis

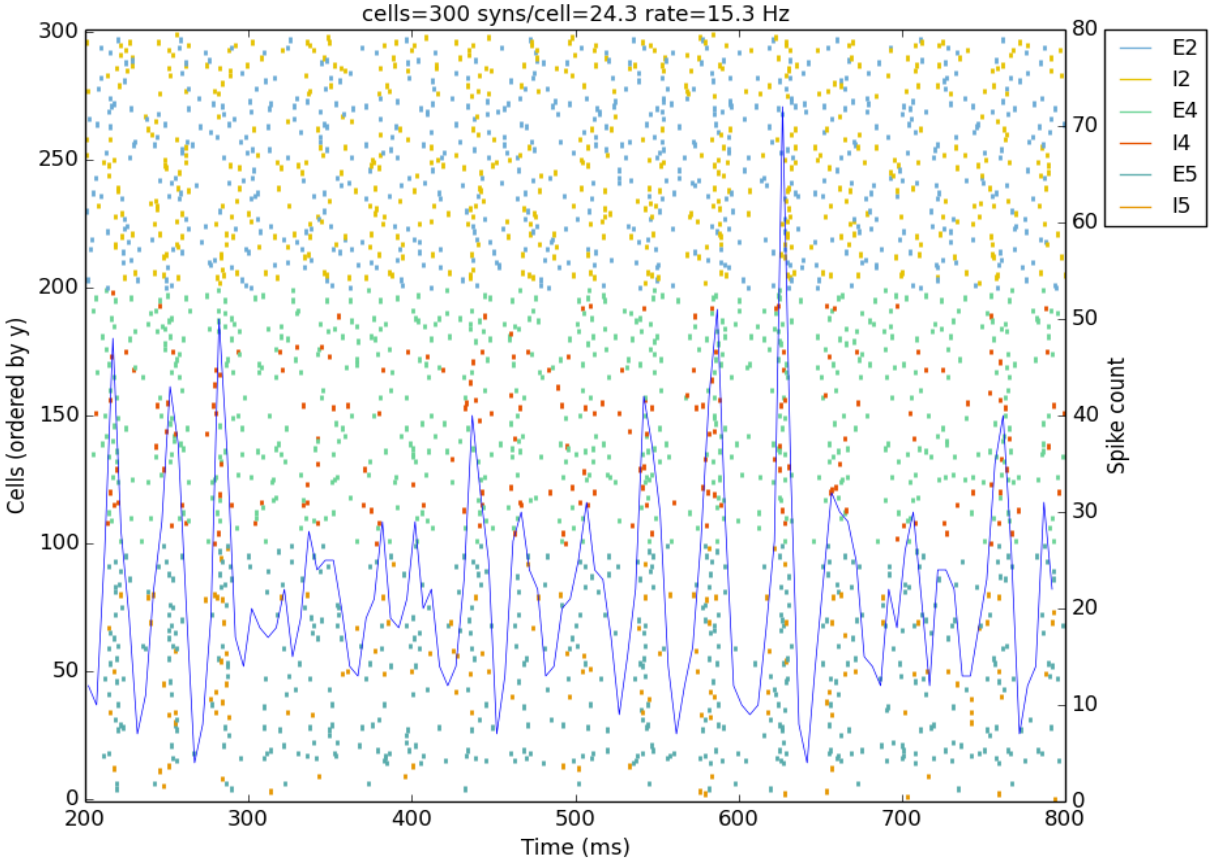
- 3D **cell shape plot**
- Option to include **color-coded variables** (eg, num of synapses)



`plotShape(...)`

NetPyNE: Analysis

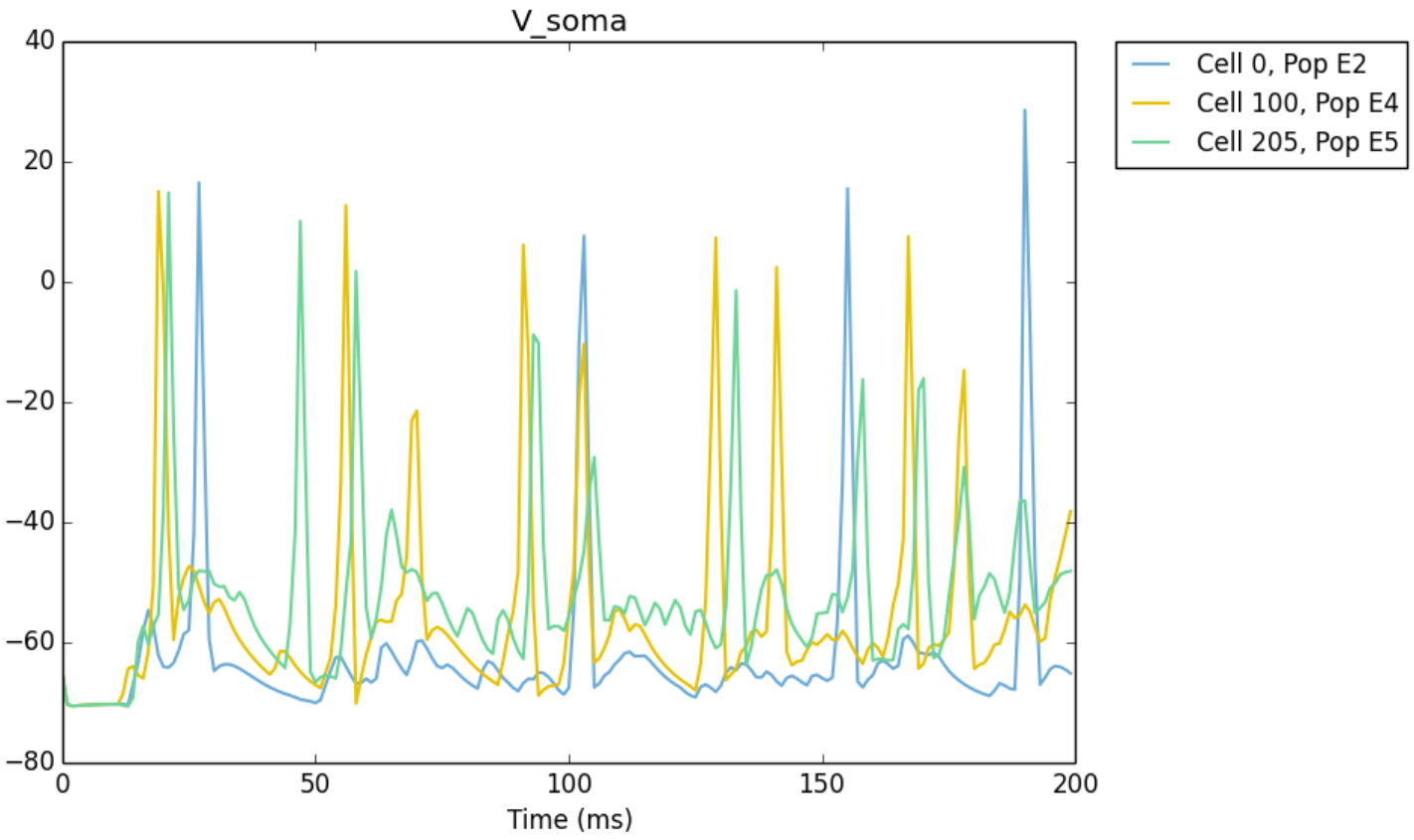
- Easy-to-use functions for **analysis and plotting** of network and simulation output
  - **Raster plot** of any subset of cells
  - **Spike histogram** of populations or subsets of cells




```
plotRaster(include=['allCells'], timeRange=[200,800], orderBy='y',
orderInverse=True, spikeHist='overlay', spikeHistBin=5)
```

NetPyNE: Analysis

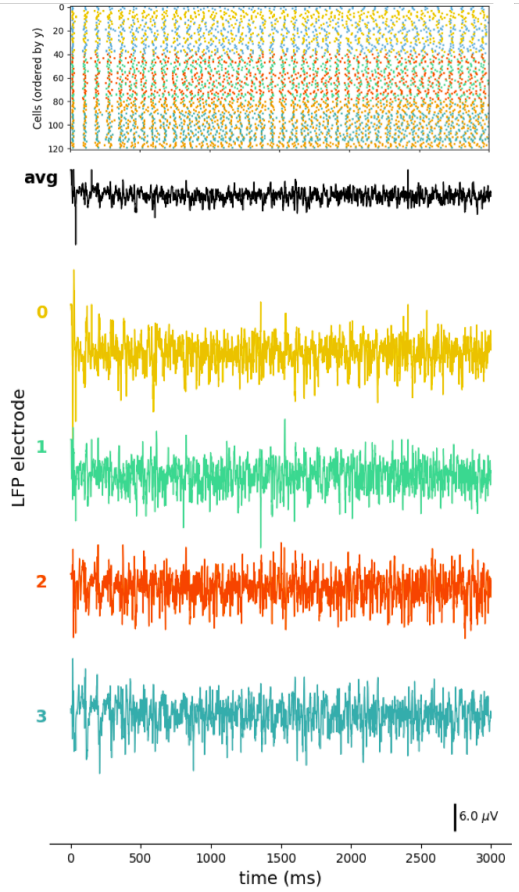
- Intrinsic cell variables (voltages, currents, conductance) **trace plots**

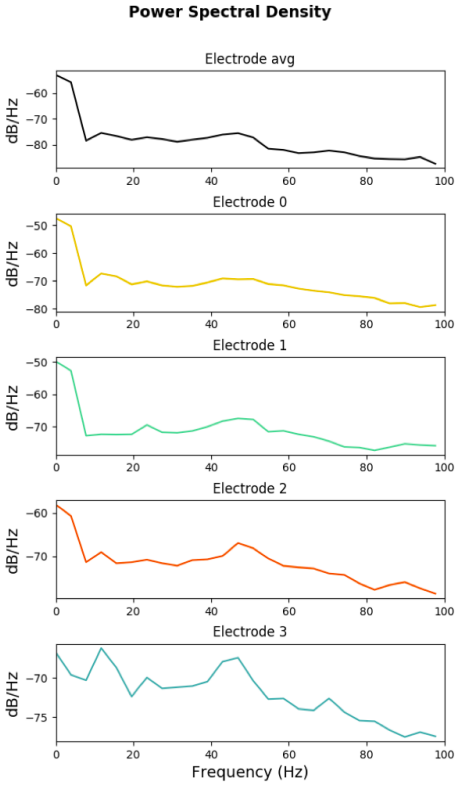


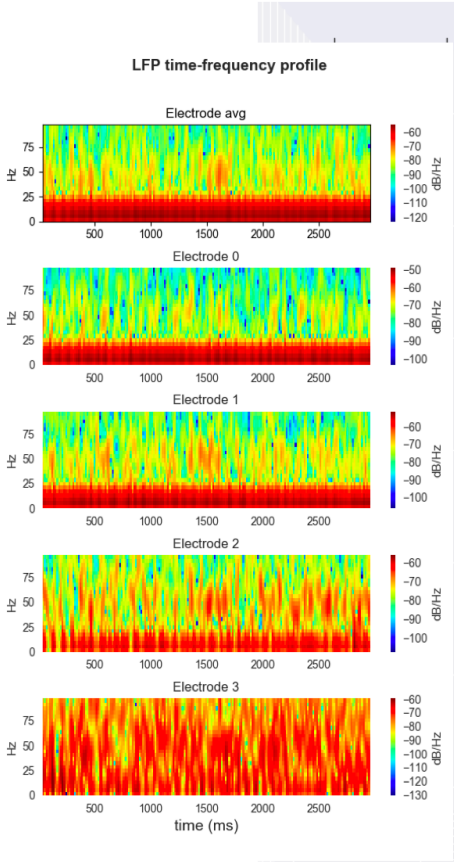

```
plotTraces(include=[('E2',0), ('E4',0), ('E5',5)],
timeRange=[0,200], overlay=True, oneFigPer='trace')
```

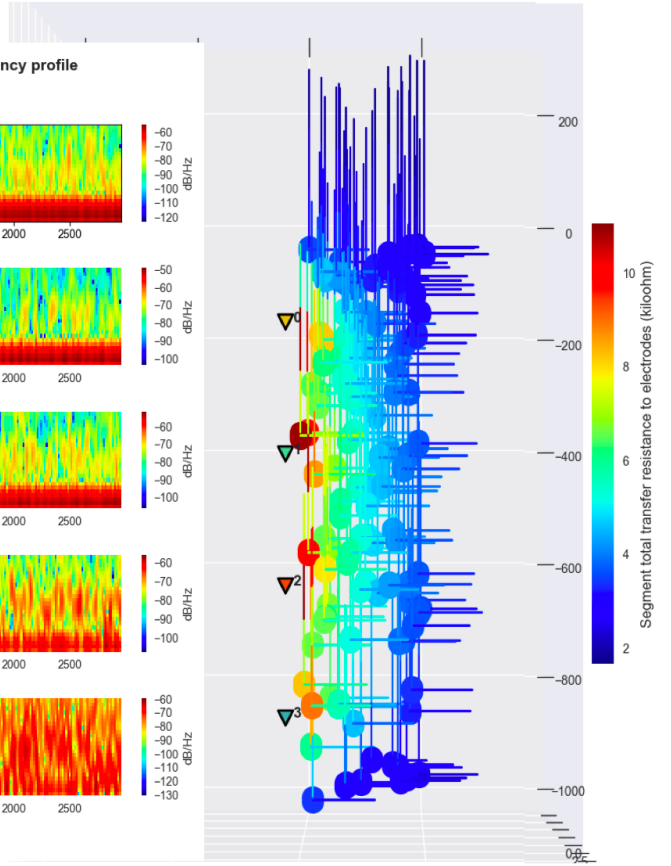
NetPyNE: Analysis

- **LFP** time-series, PSD, spectrogram and electrode locations





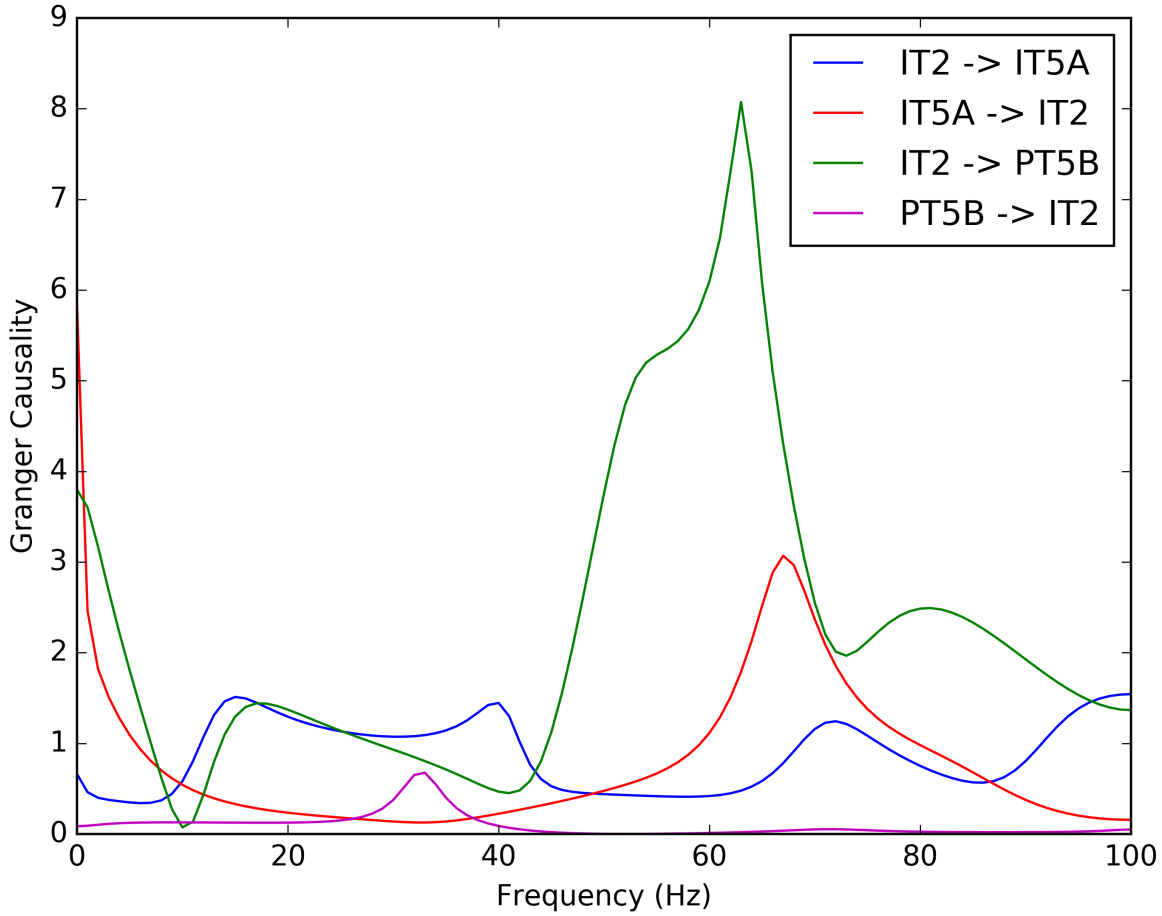




`plotLFP(...)`

NetPyNE: Analysis

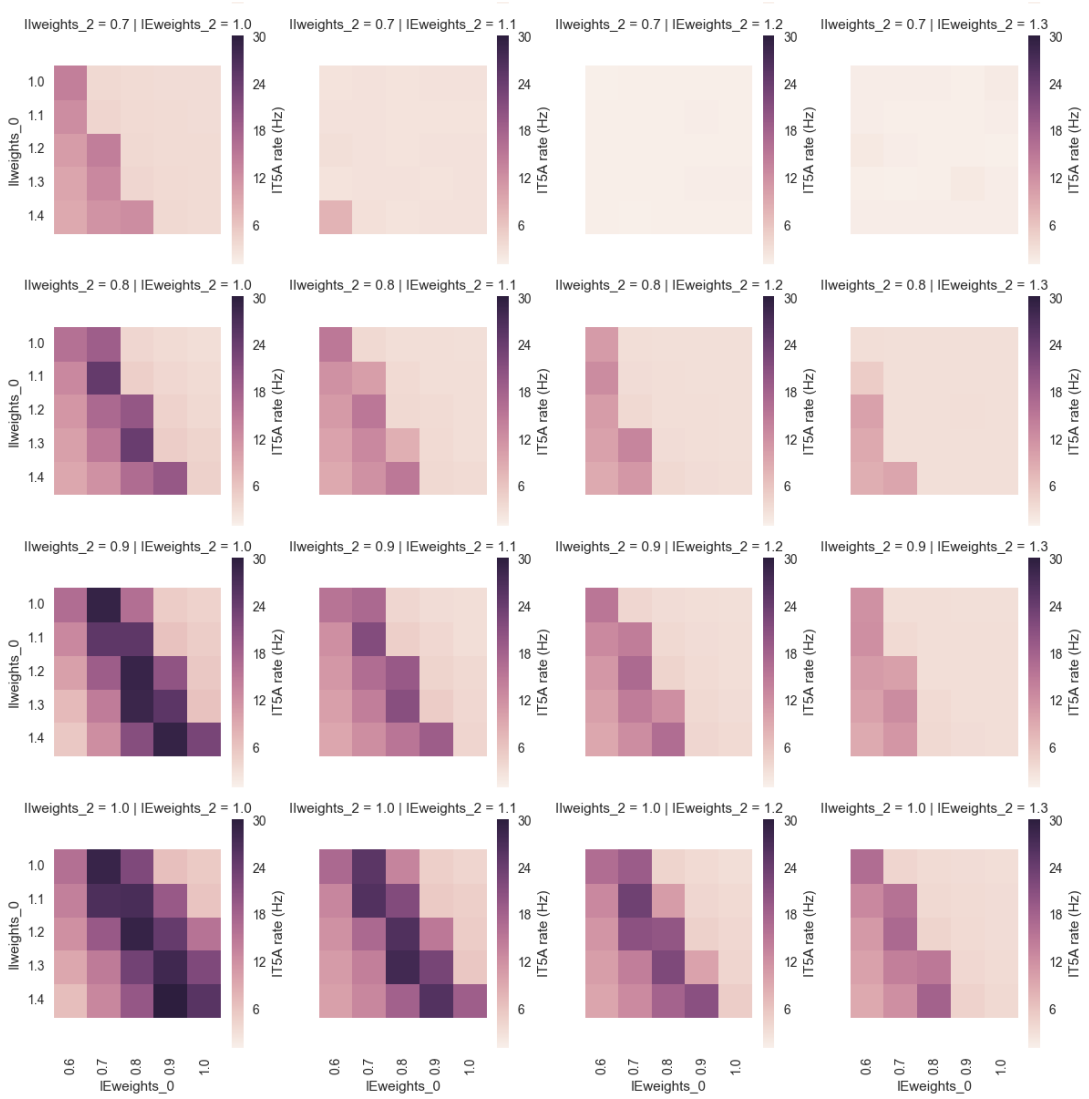
- Spectral **Granger causality**
- Normalized **transfer entropy**



`plotGranger(…)`

NetPyNE: Analysis

- **Analysis and visualization** of multidimensional batch simulation results.



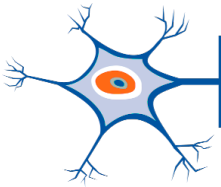
NetPyNE: Data saving and exporting

- **Save and load** high-level specifications, network instance, simulation config and/or simulation results.
- **Multiple formats** supported: pickle, Matlab, JSON, CSV, HDF5
- **Export/import** network instance to/from **NeuroML**, the standard format for neural models.

NetPyNE: Data saving and exporting

netPyne

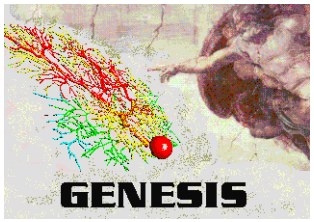
Import/export to standard format



Import/export to other simulators

nest ::

PyNN



NetPyNE: Documentation and Tutorials

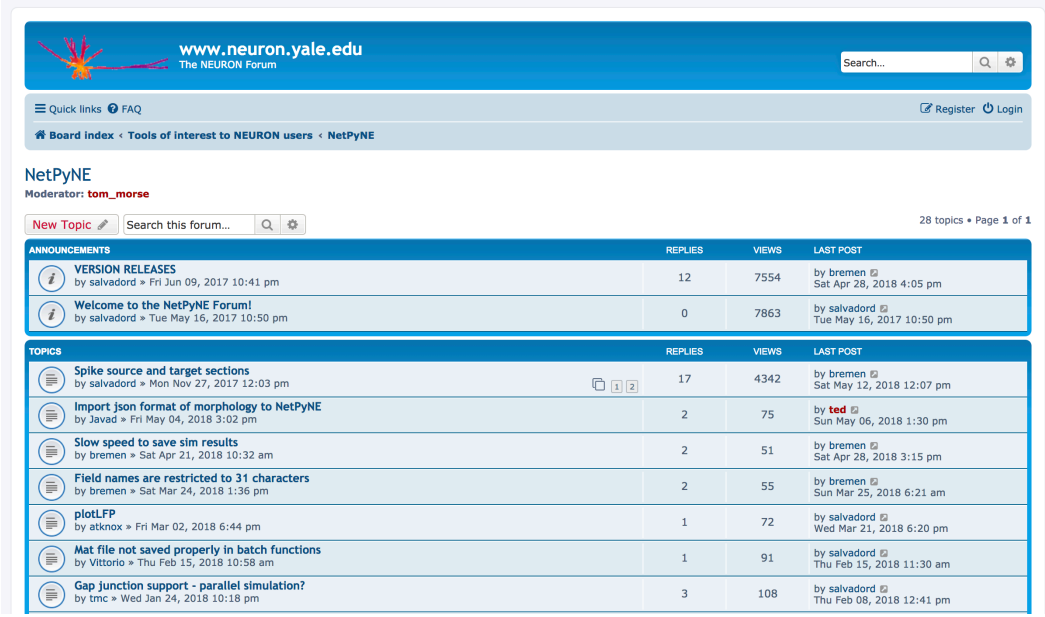
www.netpyne.org

Welcome to NetPyNE's documentation!

NetPyNE is a python package to facilitate the development and parallel simulation of biological cell networks using the NEURON simulator.

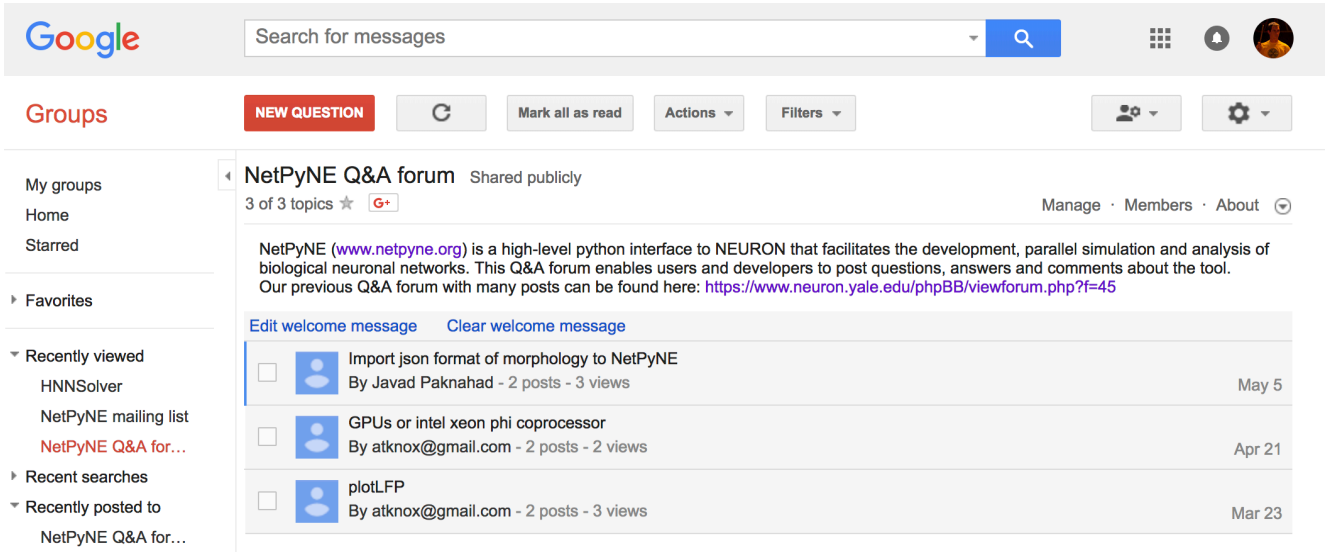
Table of Contents

NetPyNE: Q&A Forums

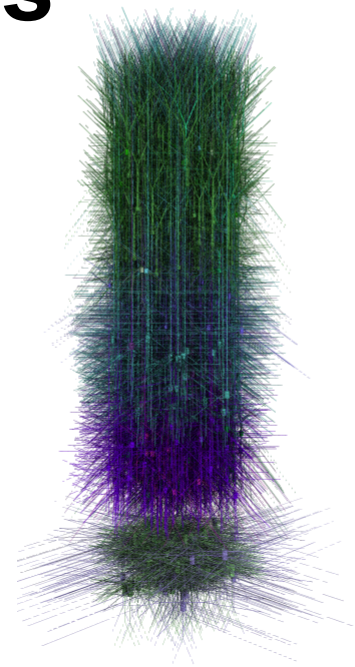


https://www.neuron.yale.edu/phpBB/viewforum.php?f=45&sid=99554ea5df10540d9b31e0c74929eaf0

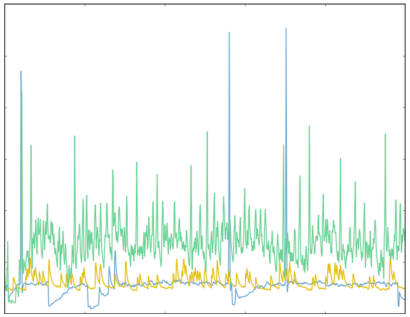
https://groups.google.com/forum/#!forum/netpyne-forum



NetPyNE: Existing models



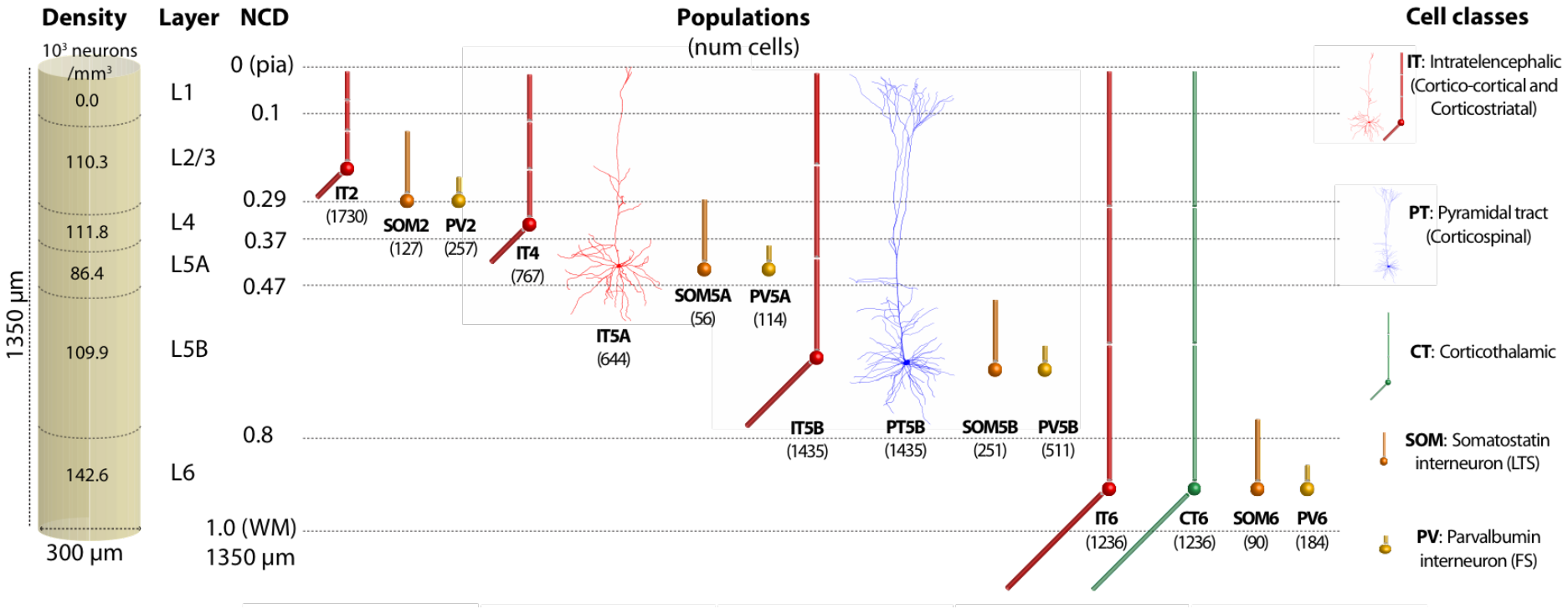
- Other models in progress:
  - Traub **thalamocortical** network (Padraig Gleeson, UCL)
  - Hippocampus **CA3** (Ben Tessler, SUNY DMC)
  - **Ischemia** in cortical network (Alex Seidenstein, SUNY DMC)
  - **STDP** in biophysically detailed networks (Anatoly Buchin, Allen Brain)
  - **Basal Ganglia** network (Lucas, UCD)
  - **LFP** oscillations (Christian Fink, Ohio Wesleyan)
  - **Dendritic** computations (Birgit Kriener, Oslo)
  - Thalamocortical **epilepsy** network (Andrew Knox, Cincinatti Hospital)
  - **V1** network with Allen Brain cells (SUNY DMC)
  - **Schizophrenia** in cortical network (Cristoph Metzner, Hertfordshire)
  - **Spinal cord** circuits (Vittorio Caggiano, IBM Watson)



- Full list of 43 models: https://drive.google.com/open?id=1bkWHakgZoEkYIkzrAS8sIKCvO5PSuUXLLRjNdN2pseY

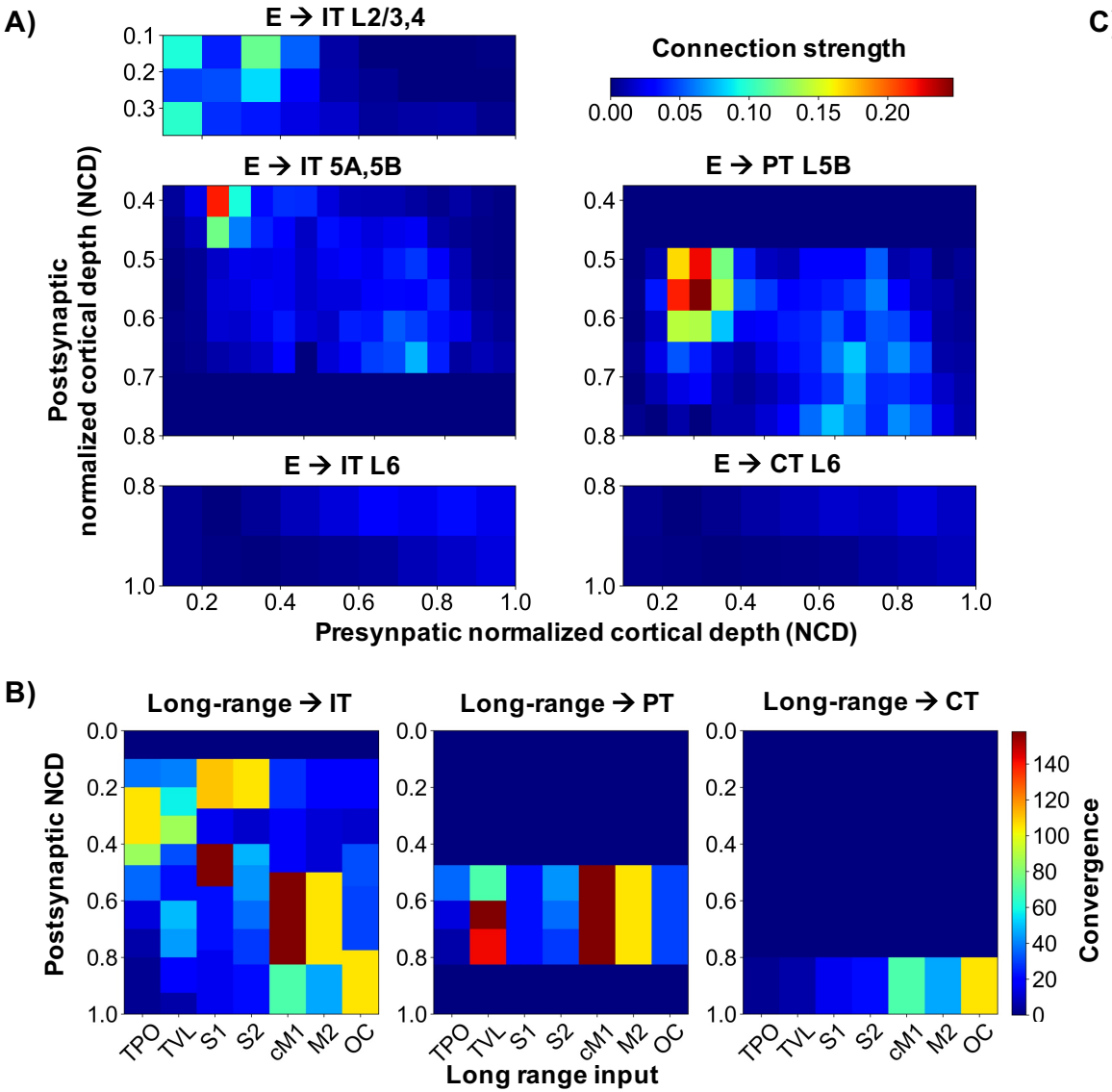
NetPyNE: M1 microcircuits

- Data-driven multiscale network model of **M1 microcircuits**



Mouse 6-layer M1 with **10,074 neurons** of 5 classes distributed in 15 populations;
Full scale cylindric volume of **300 µm** (diameter) x **1350 µm** (cortical depth)

NetPyNE: M1 microcircuits



NetPyNE: Acknowledgments

- **Contributors:**
  - Salvador Dura-Bernal (SUNY DMC)
  - Ben Suter (Northwestern)
  - Adrian Quintana (Eyeseetea Ltd)
  - Matteo Cantarelli (Metacell Ltd)
  - Facundo Rodriguez (Instituto Balseiro)
  - Padraig Gleeson (UCL)
  - Joe Graham (SUNY DMC)
  - Cliff Kerr (Sydney University)
  - Robert McDougal (Yale)
  - Michael Hines (Yale)
  - Gordon MG Shepherd (Northwestern)
  - William Lytton (SUNY DMC)

- **Lab website:** www.neurosimlab.org

- **NetPyNE Website:** www.netpyne.org

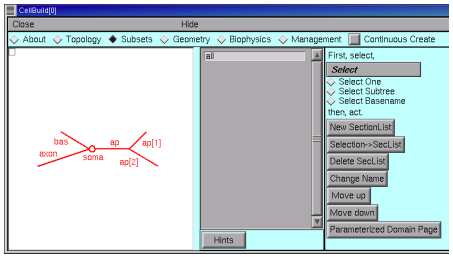
- **Github:** www.github.com/Neurosim-lab/netpyne
(open source development; contributions welcome)

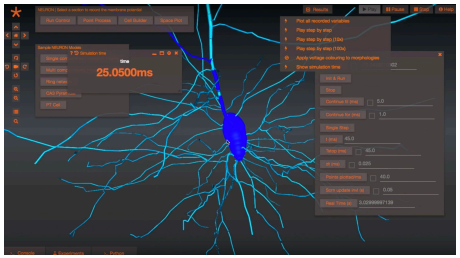
- **Funding:**
  - NIH Grant U01EB017695
  - NIH Grant R01EB022903
  - NIH Grant R01MH086638
  - NYS Grant DOH01-C32250GG-3450000

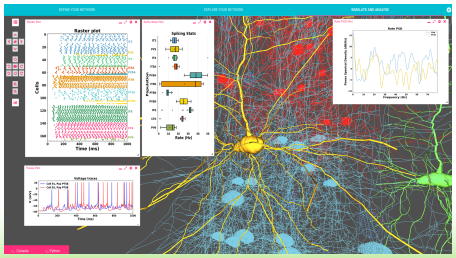
SUNY Downstate Medical Center

UCL

NetPyNE GUI Tutorial







OLD
NEURON GUI

NEURON

NEW
NEURON GUI

NEURON

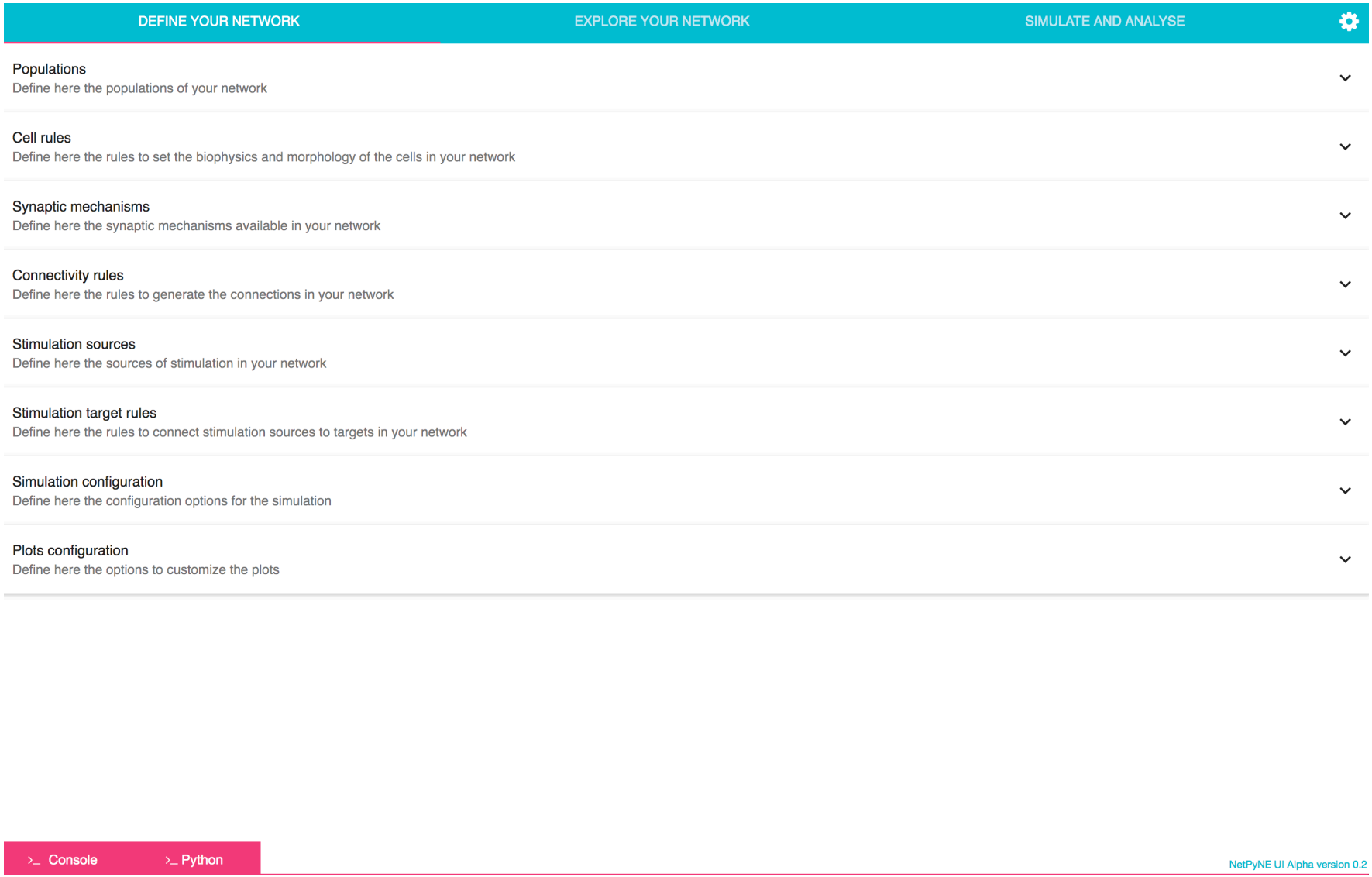
NetPyNE GUI

NetPyNE

NEURON

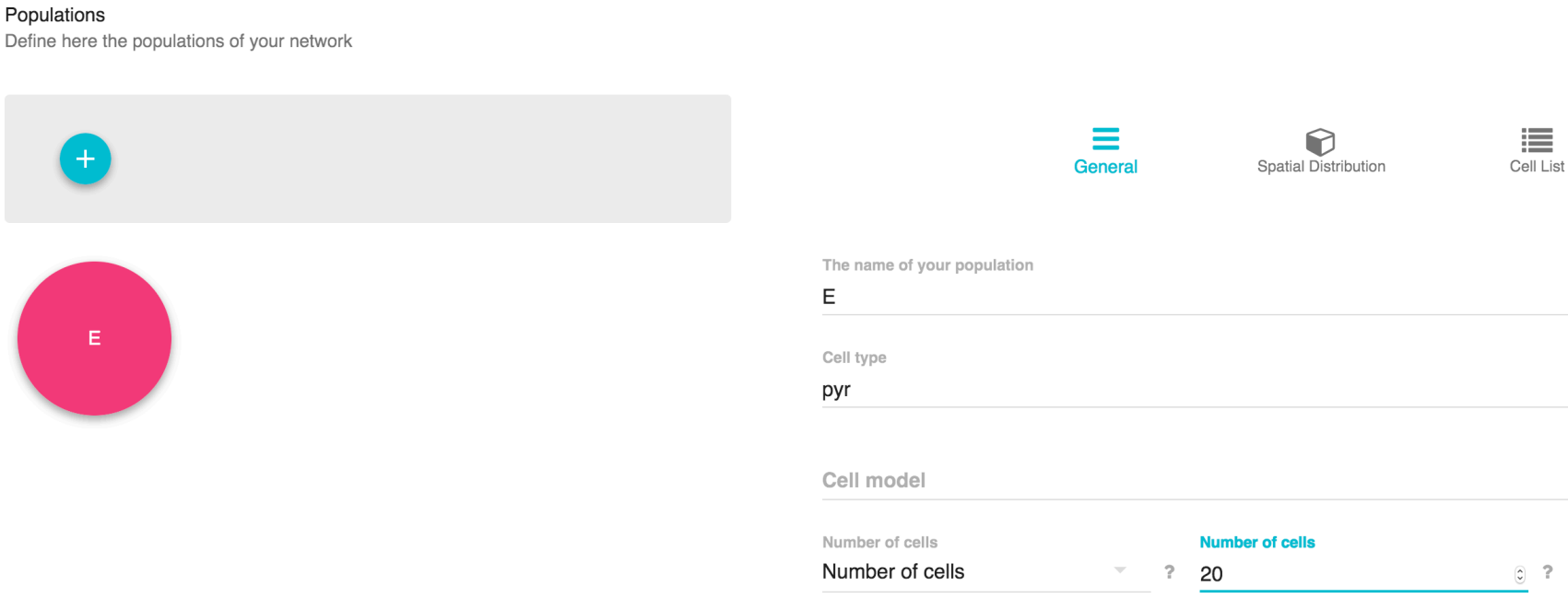
NetPyNE GUI Tutorial: Simple cell net

1) Open NetPyNE GUI on web browser



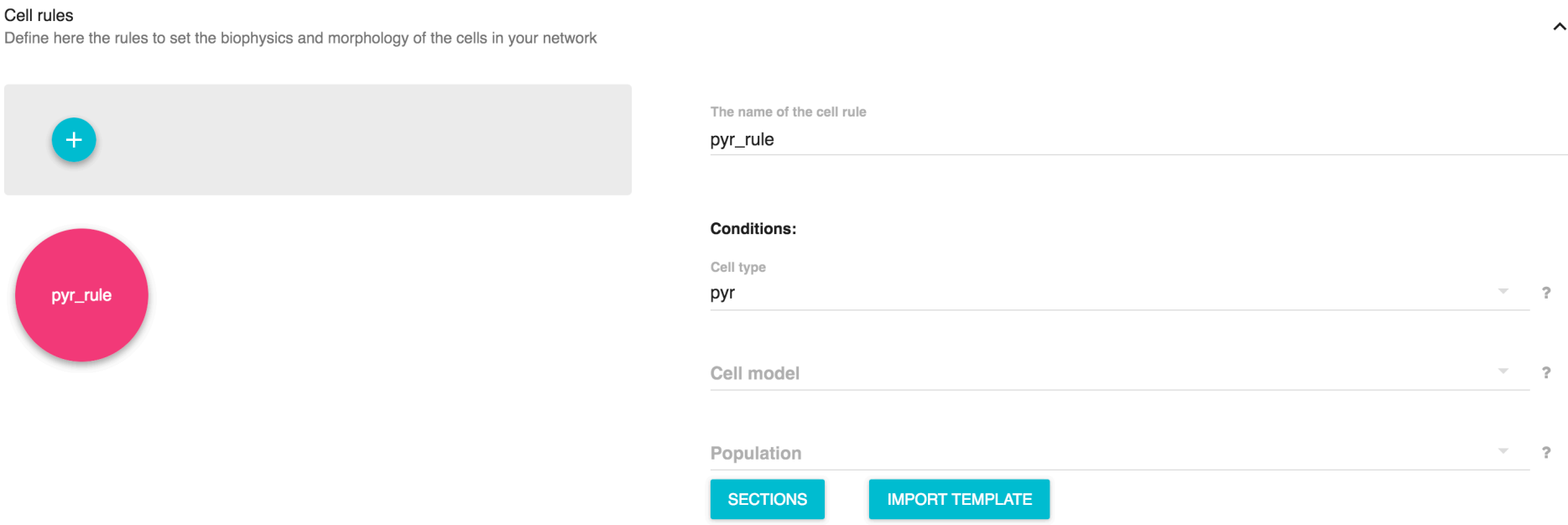
NetPyNE GUI Tutorial: Simple cell net

2) Add a population ’E’ of 20 ‘pyr’ cells



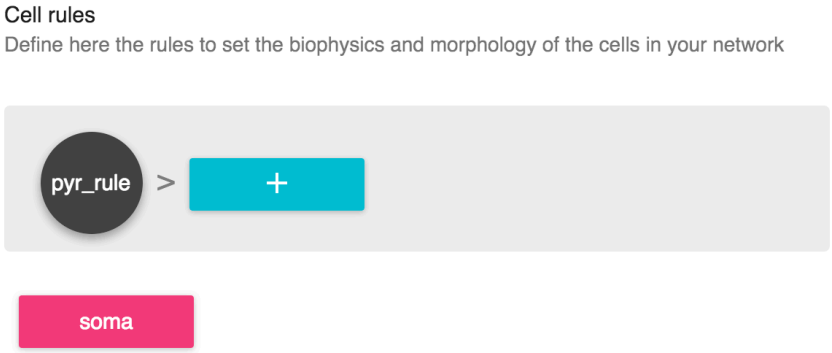
NetPyNE GUI Tutorial: Simple cell net

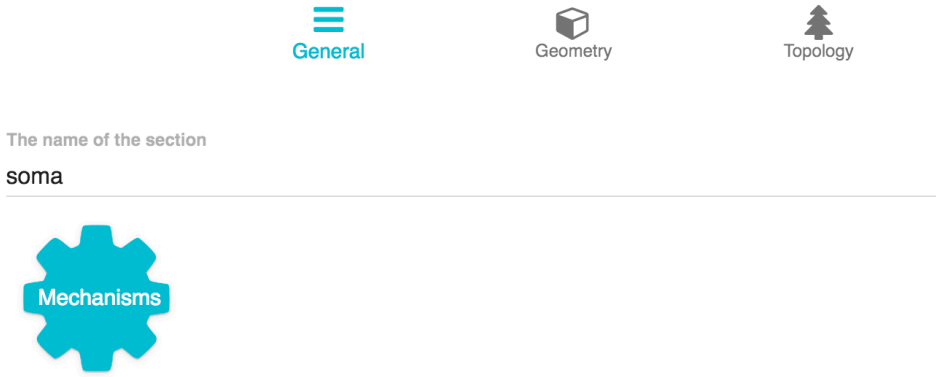
3) Add a cell rule ‘pyr_rule’ for ‘pyr’ cells



NetPyNE GUI Tutorial: Simple cell net

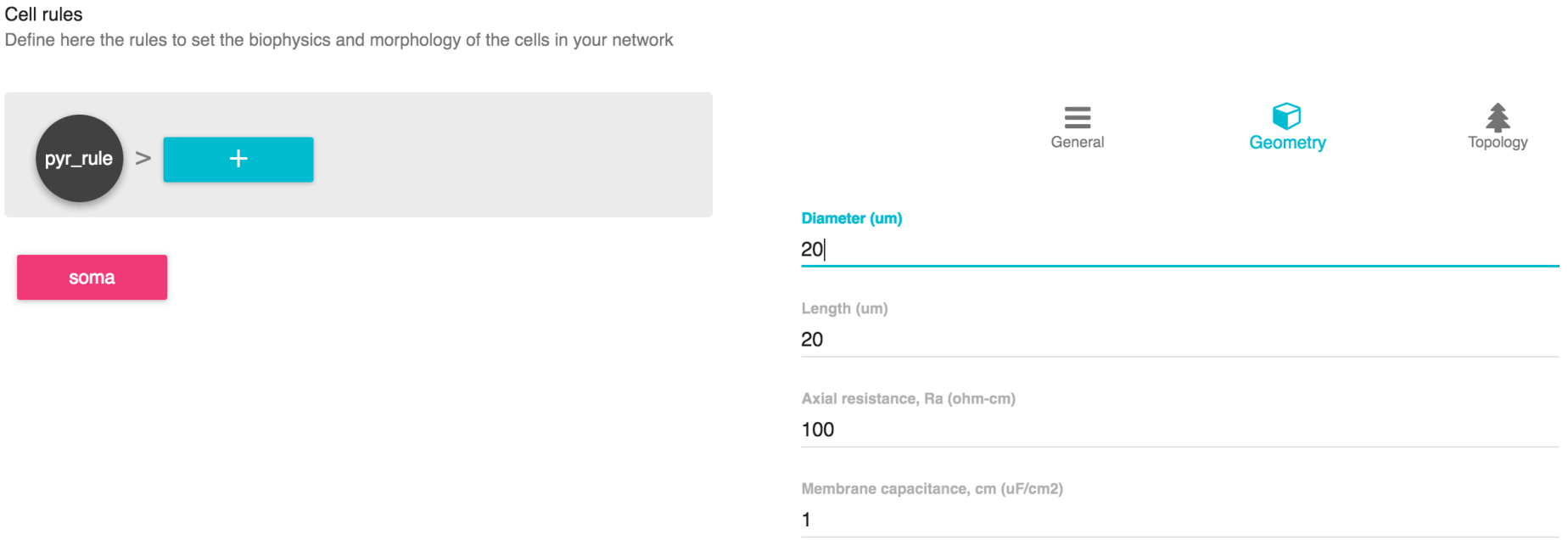
4) Add a ‘soma’ section in the ‘pyr_rule’





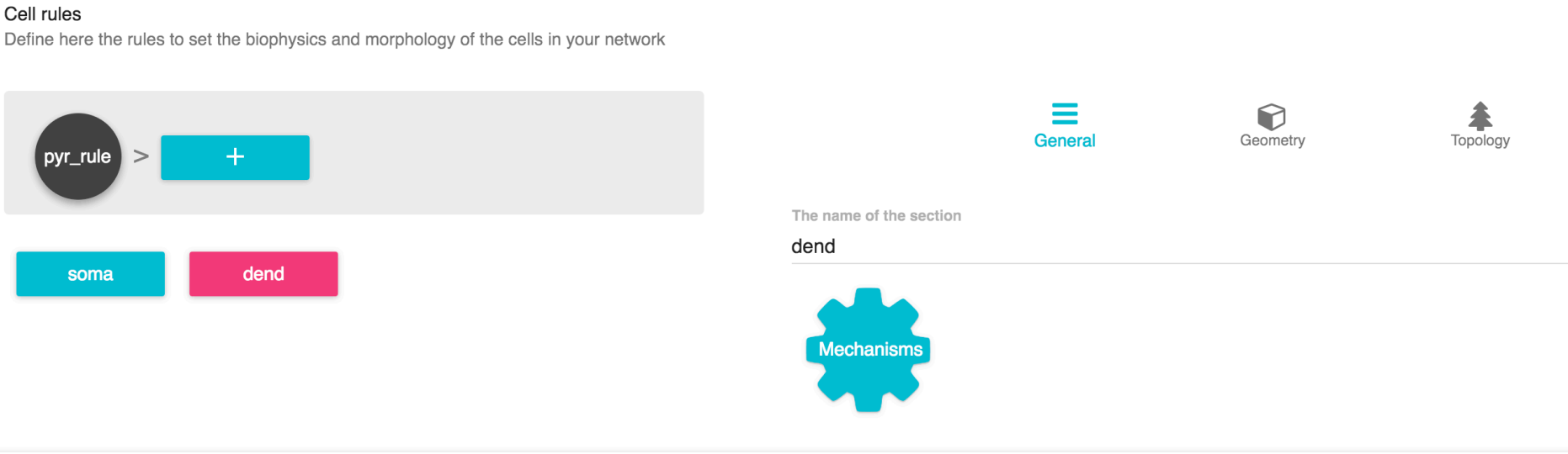
NetPyNE GUI Tutorial: Simple cell net

5) Add the geometry of the ‘soma’ section in the ‘pyr_rule’



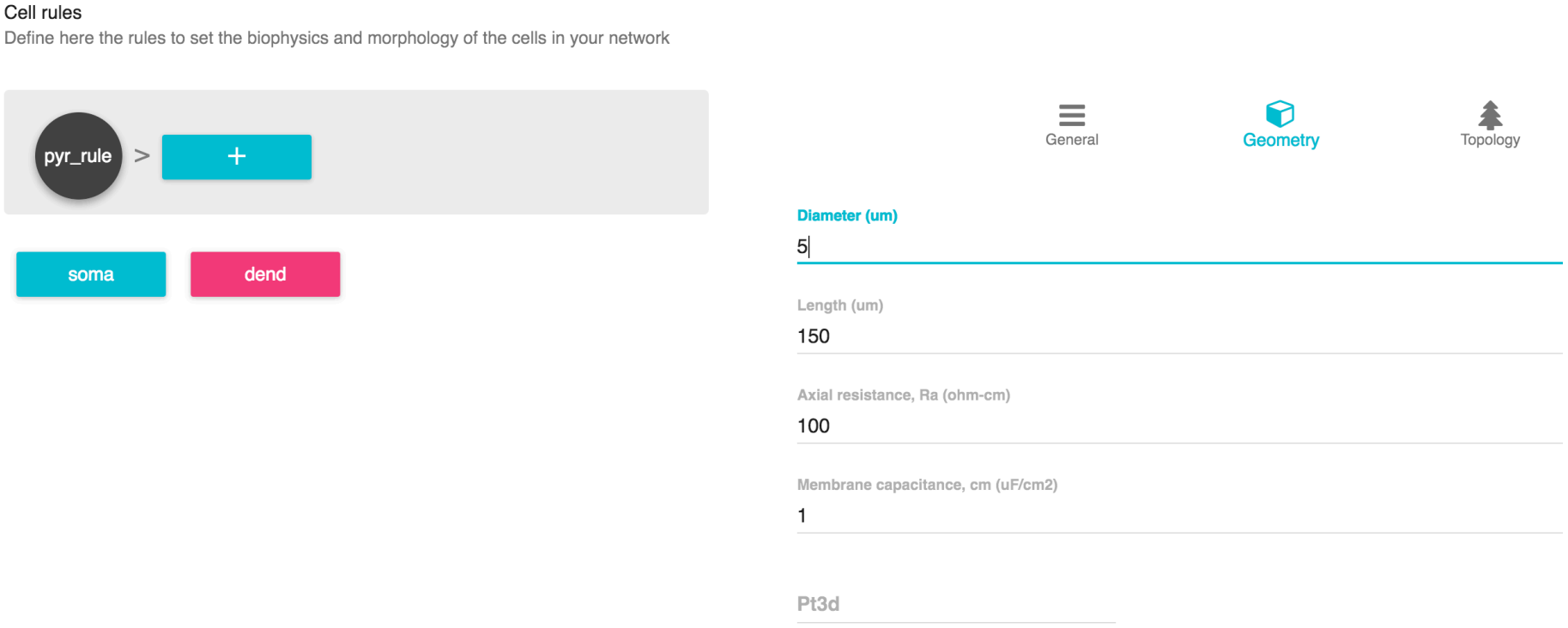
NetPyNE GUI Tutorial: Simple cell net

6) Add a ‘dend’ section in the ‘pyr_rule’



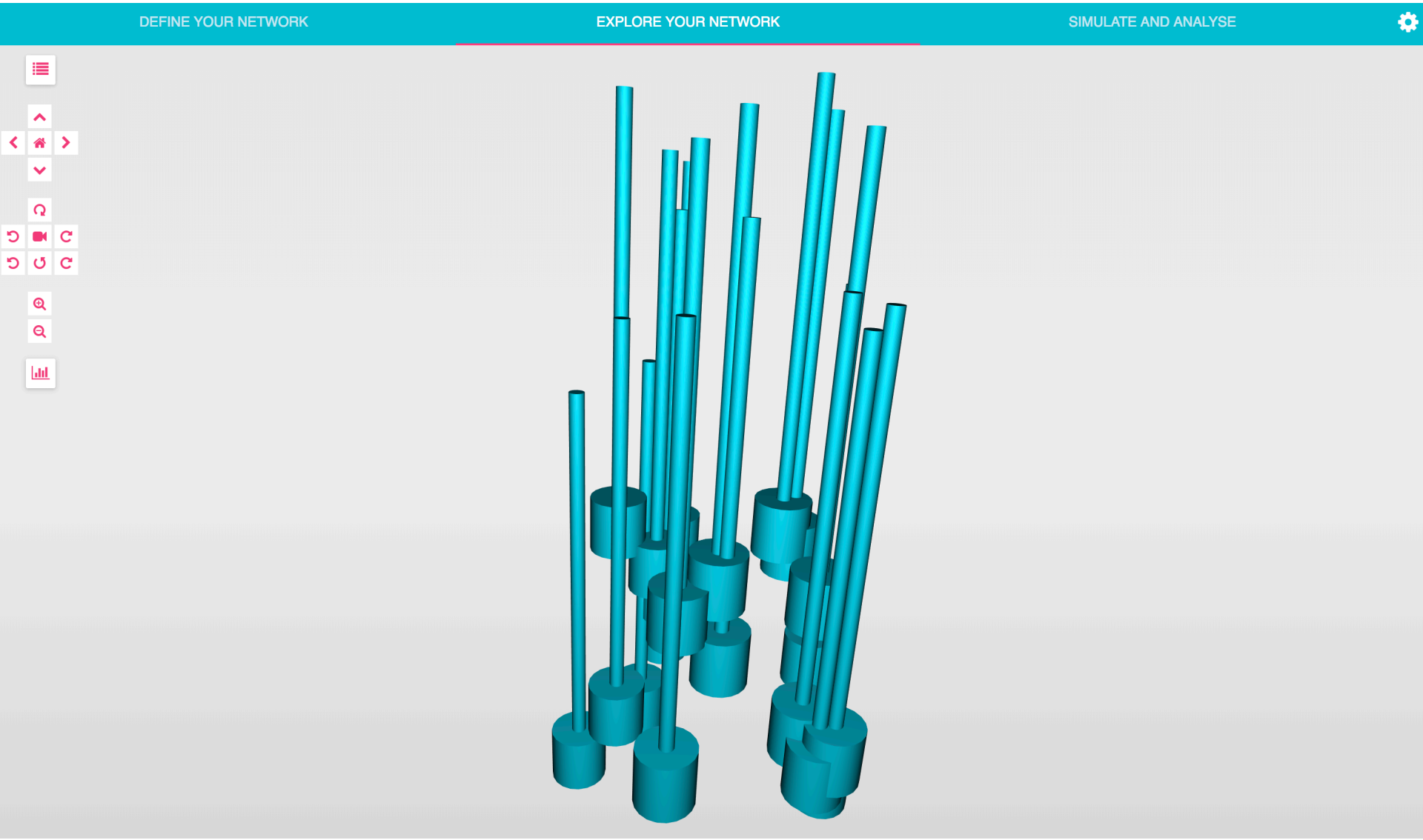
NetPyNE GUI Tutorial: Simple cell net

7) Add the geometry of the ‘dend’ section in the ‘pyr_rule’



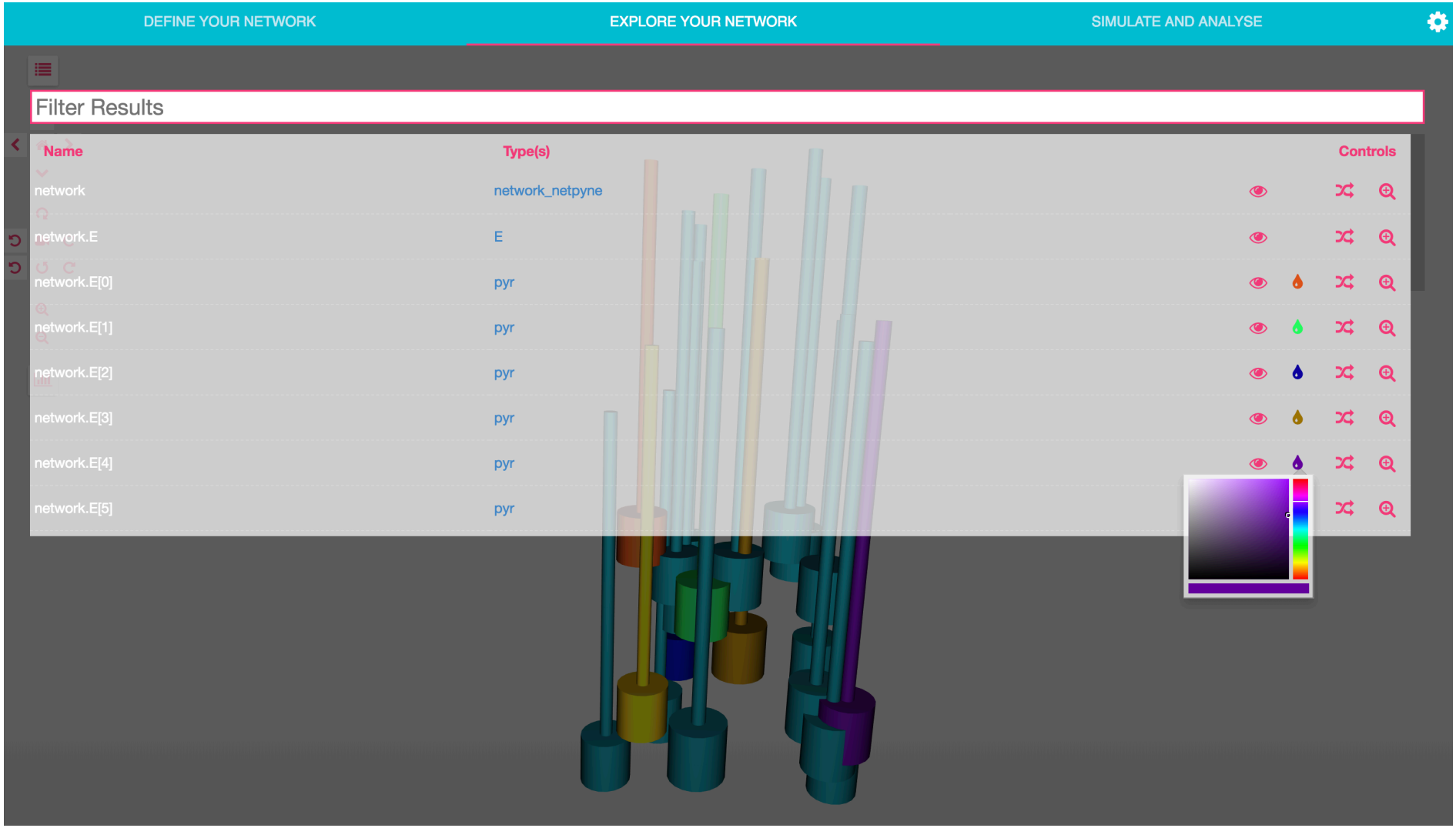
NetPyNE GUI Tutorial: Simple cell net

8) Explore your network … feel free to rotate, zoom and move around!



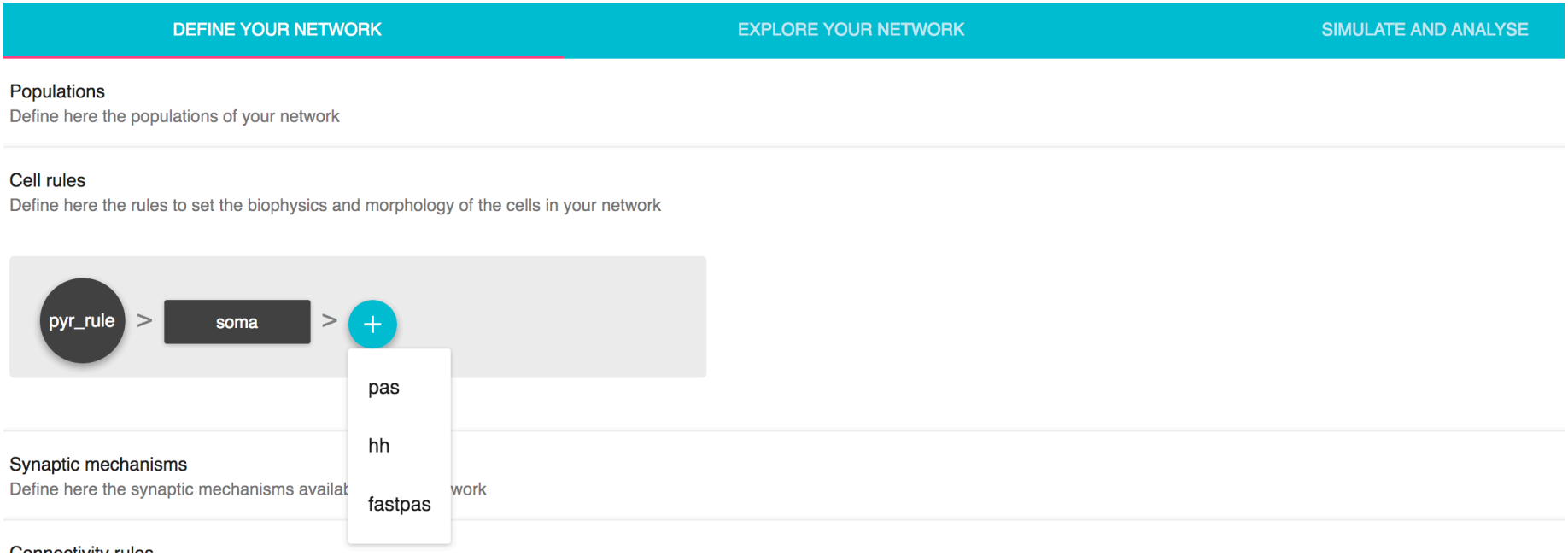
NetPyNE GUI Tutorial: Simple cell net

9) Customize the color of your population or cells



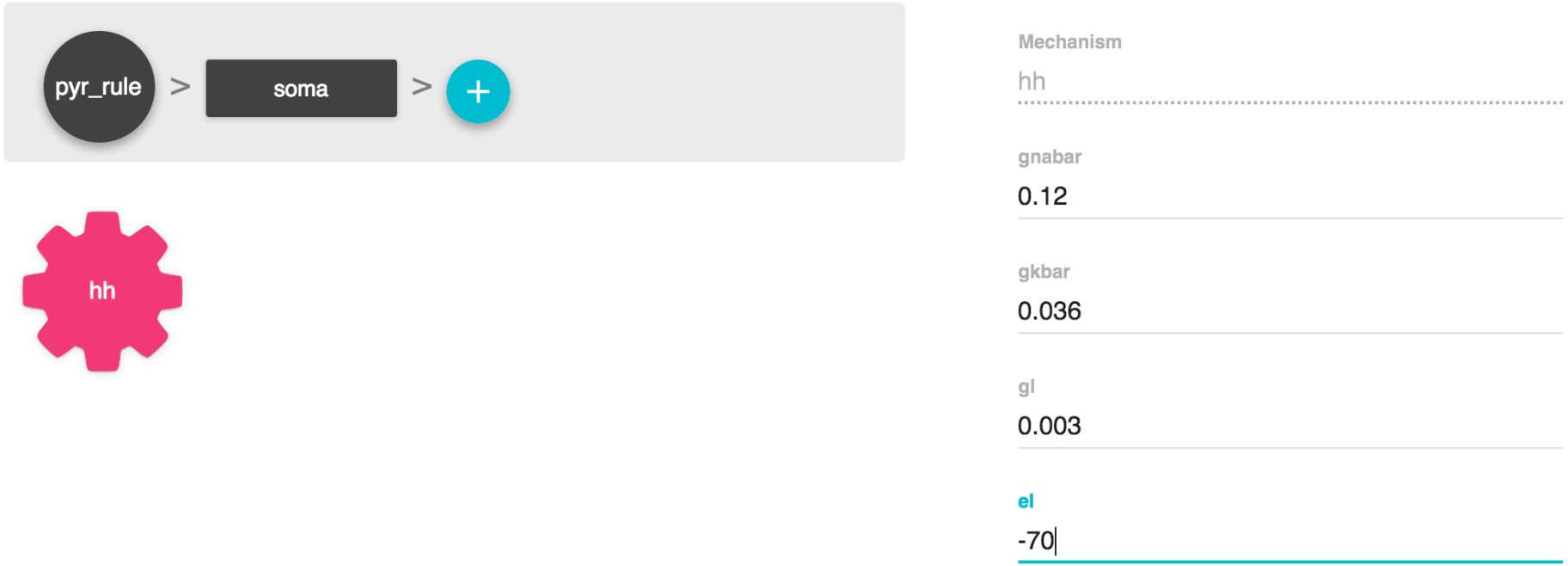
NetPyNE GUI Tutorial: Simple cell net

10) To add channels: go to ‘Define your network’ → ‘Cell rules’ → ‘pyr rule’ → ‘soma’ → ‘mechanisms’ → (+)



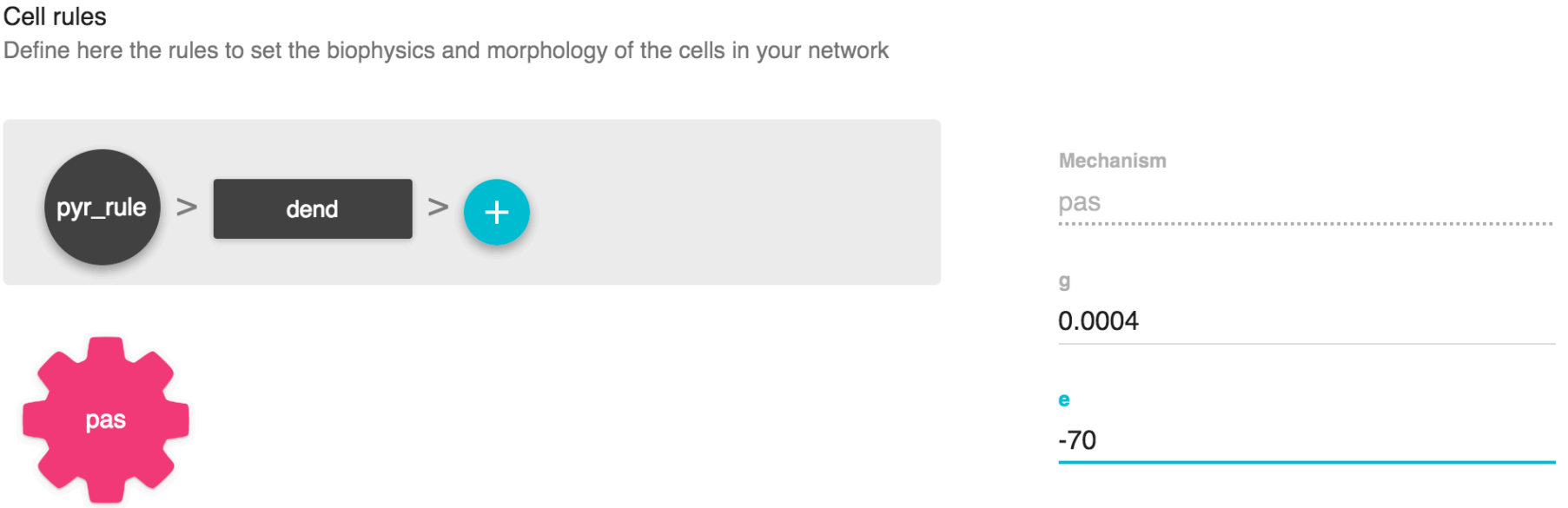
NetPyNE GUI Tutorial: Simple cell net

11) Add the ‘hh’ (Hodgkin-Huxley) mechanism to the ’soma’ with the following parameters:



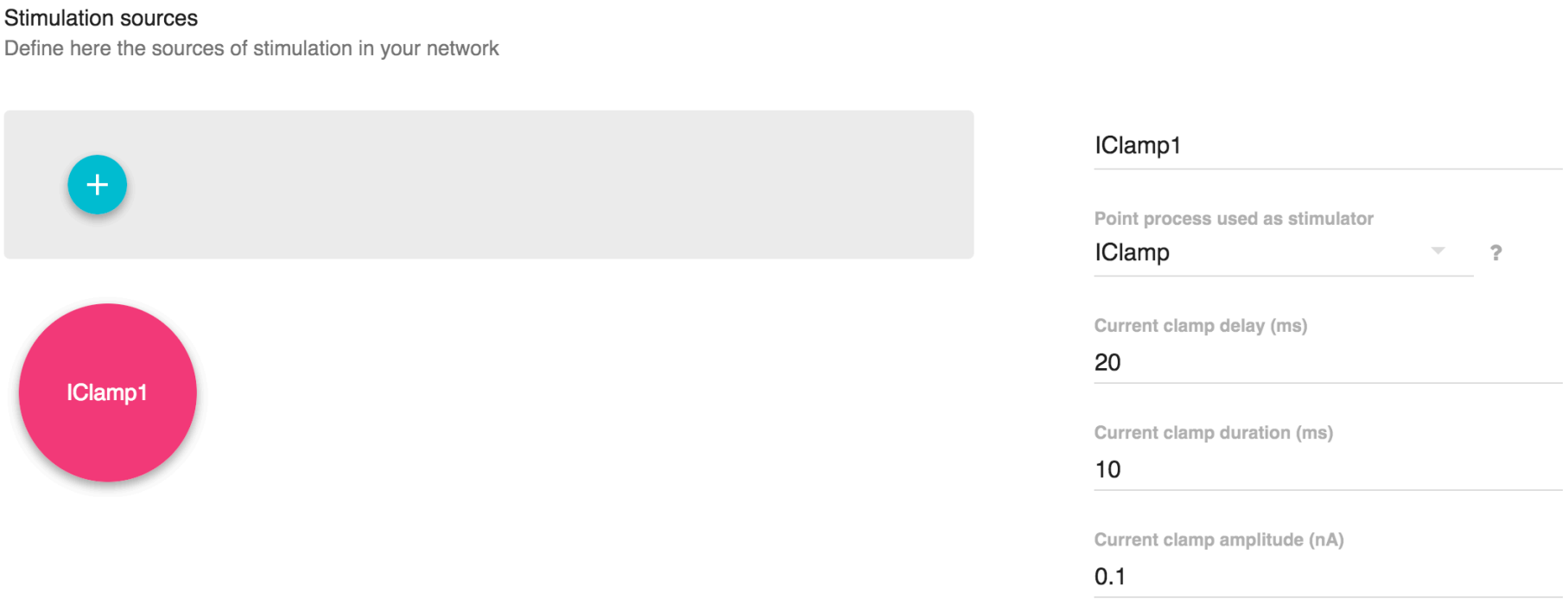
NetPyNE GUI Tutorial: Simple cell net

12) Add the ‘pas’ (passive) mechanism to the ‘dend’ with the following parameters:



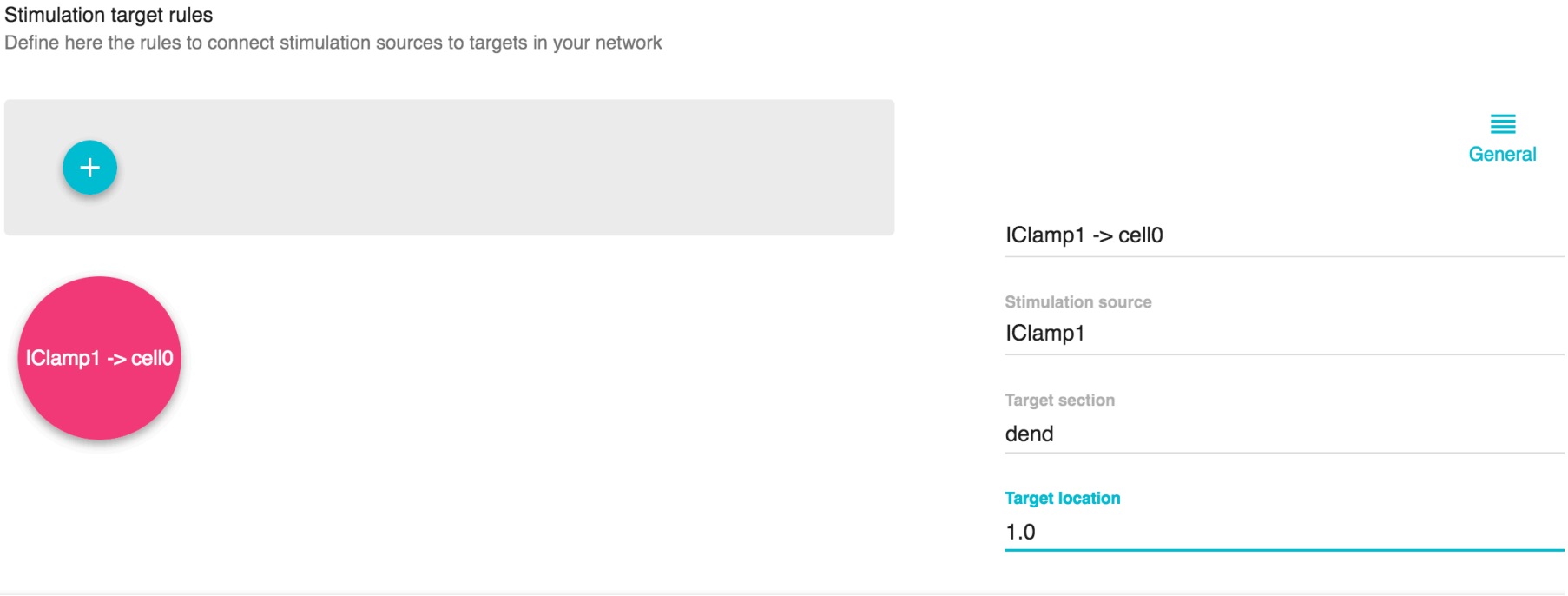
NetPyNE GUI Tutorial: Simple cell net

13) Add an IClamp (current clamp) source of stimulation



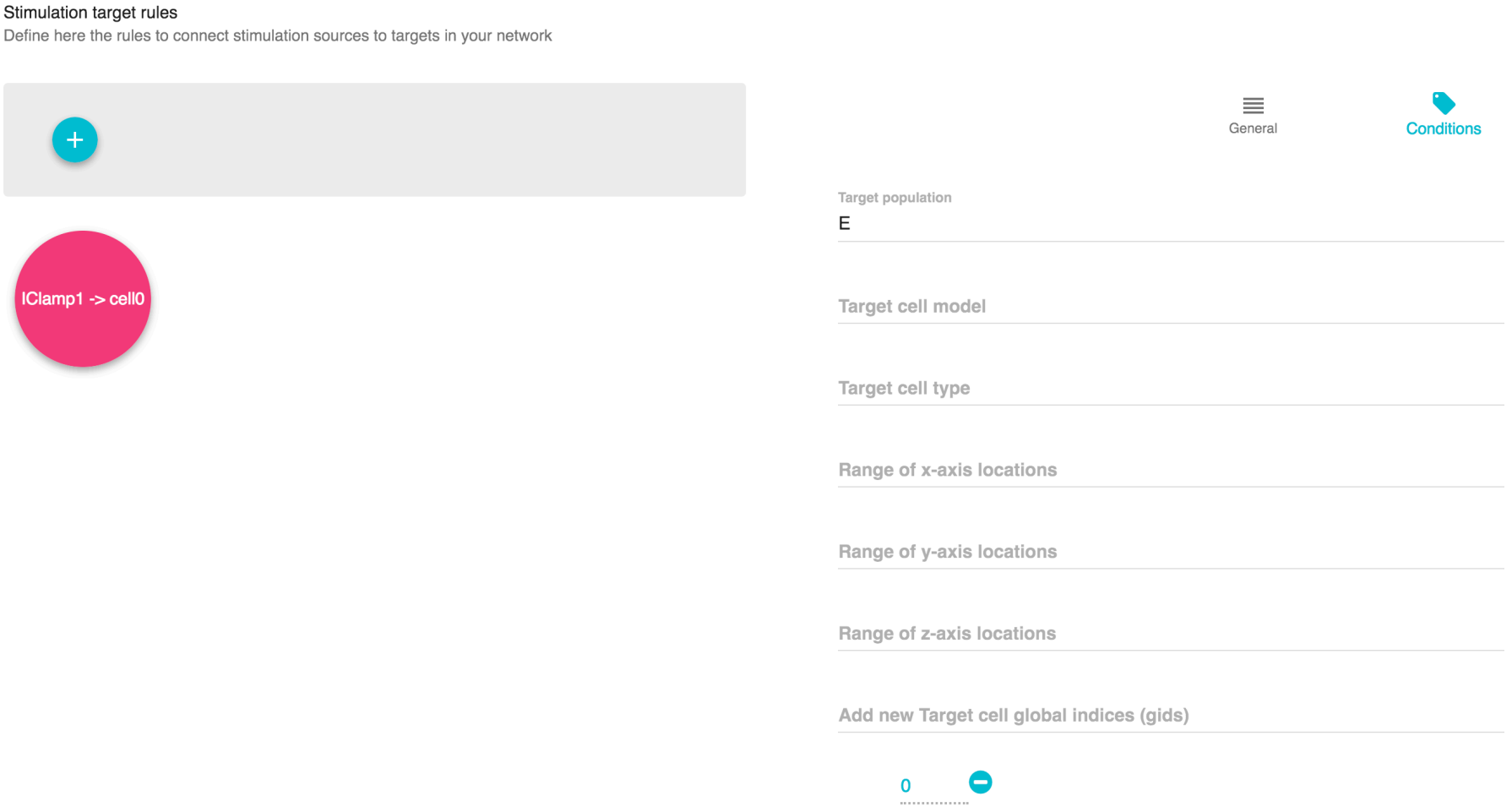
NetPyNE GUI Tutorial: Simple cell net

14) Create a stimulation target rule to place IClamp1 on the cell dendrite:



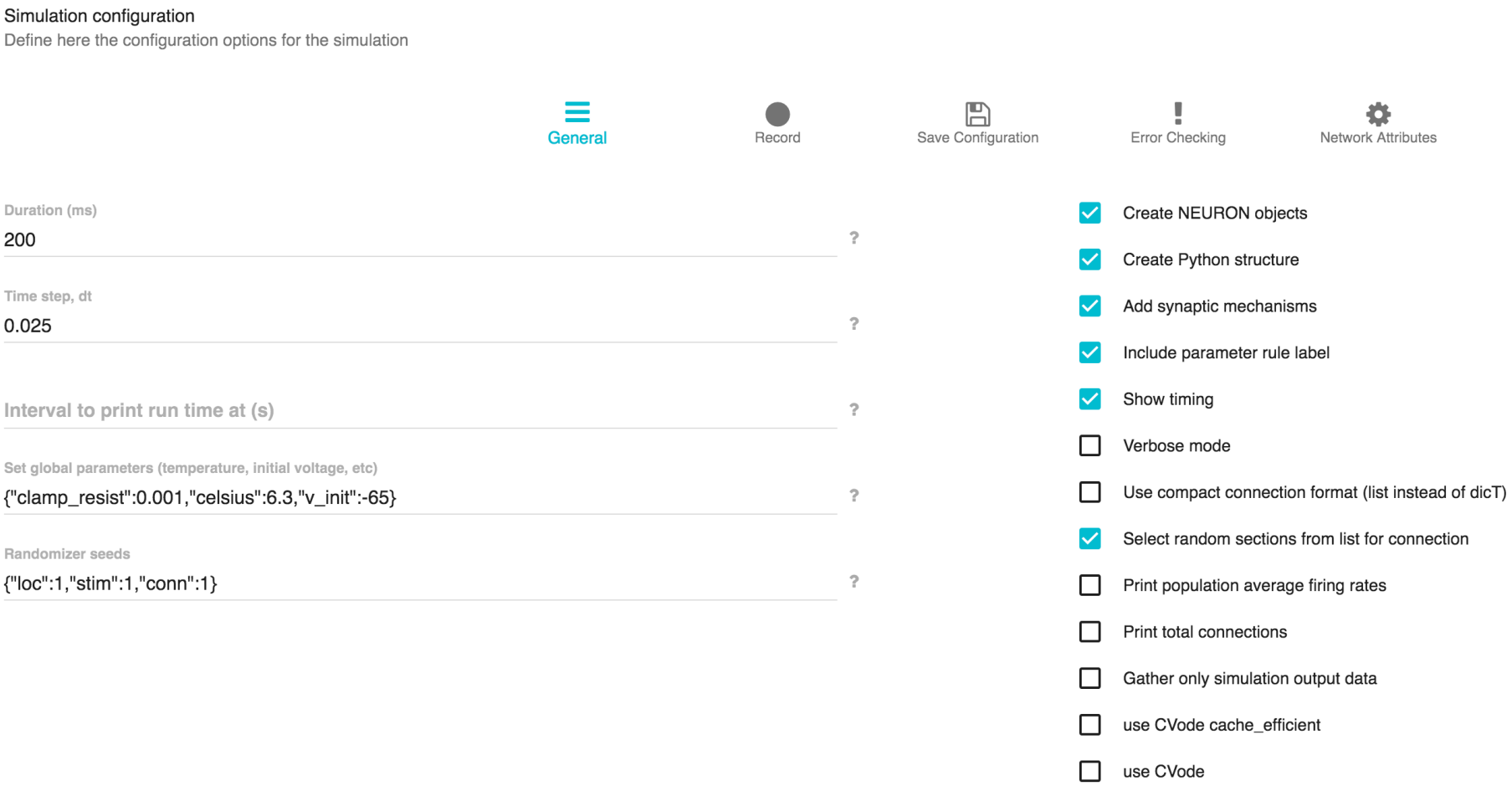
NetPyNE GUI Tutorial: Simple cell net

15) Place the IClamp1 just on one of the cells (with global index 0) using the target rule ‘conditions’:



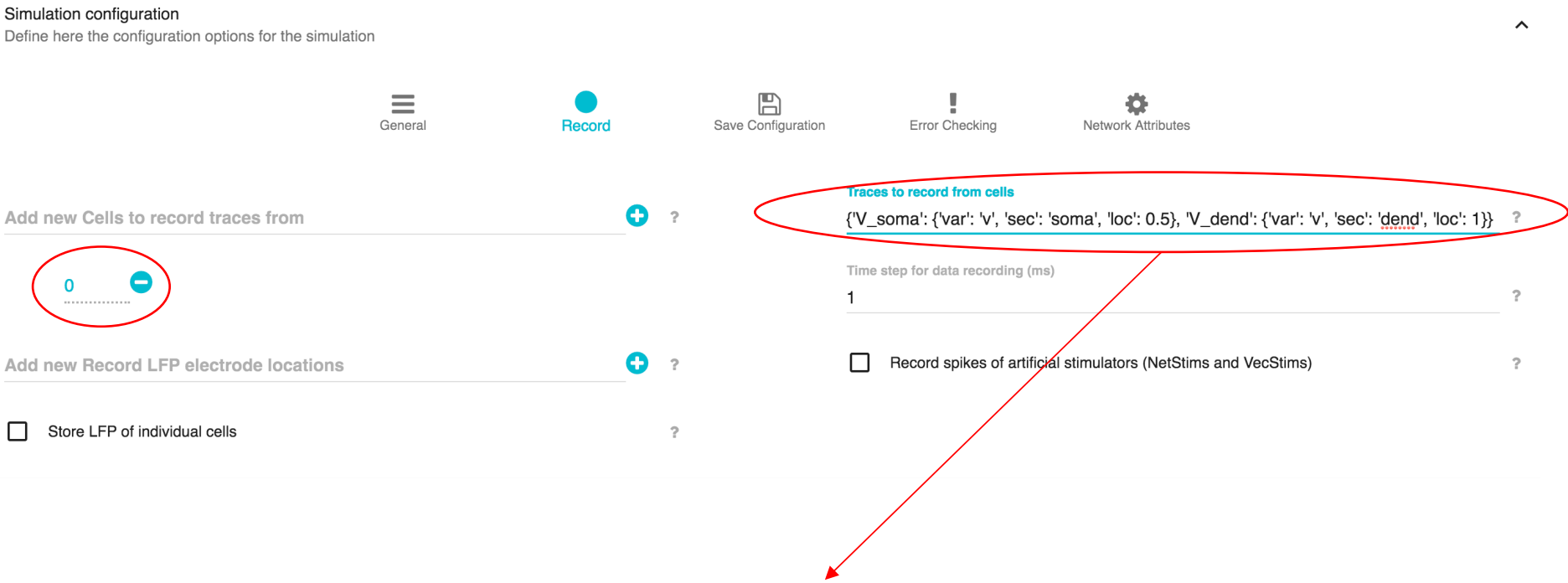
NetPyNE GUI Tutorial: Simple cell net

16) Set the simulation duration to 200ms in ‘Simulation configuration’



NetPyNE GUI Tutorial: Simple cell net

17) Record soma and dendrite voltage traces from from cell with id 0:

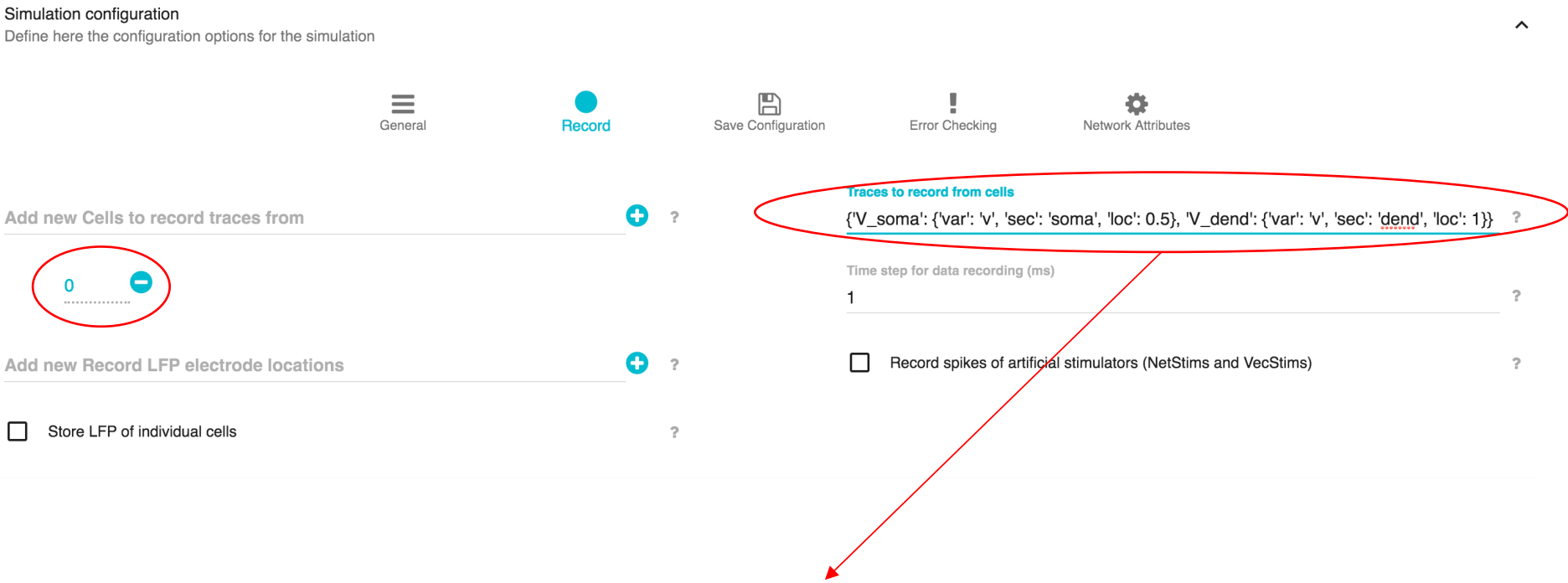


Specify traces to record using Python dictionary format:

```
{'V_soma': {'var': 'v', 'sec': 'soma', 'loc': 0.5},
 'V_dend': {'var': 'v', 'sec': 'dend', 'loc': 1.0}}
```

NetPyNE GUI Tutorial: Simple cell net

17) Record soma and dendrite voltage traces from from cell with id 0:

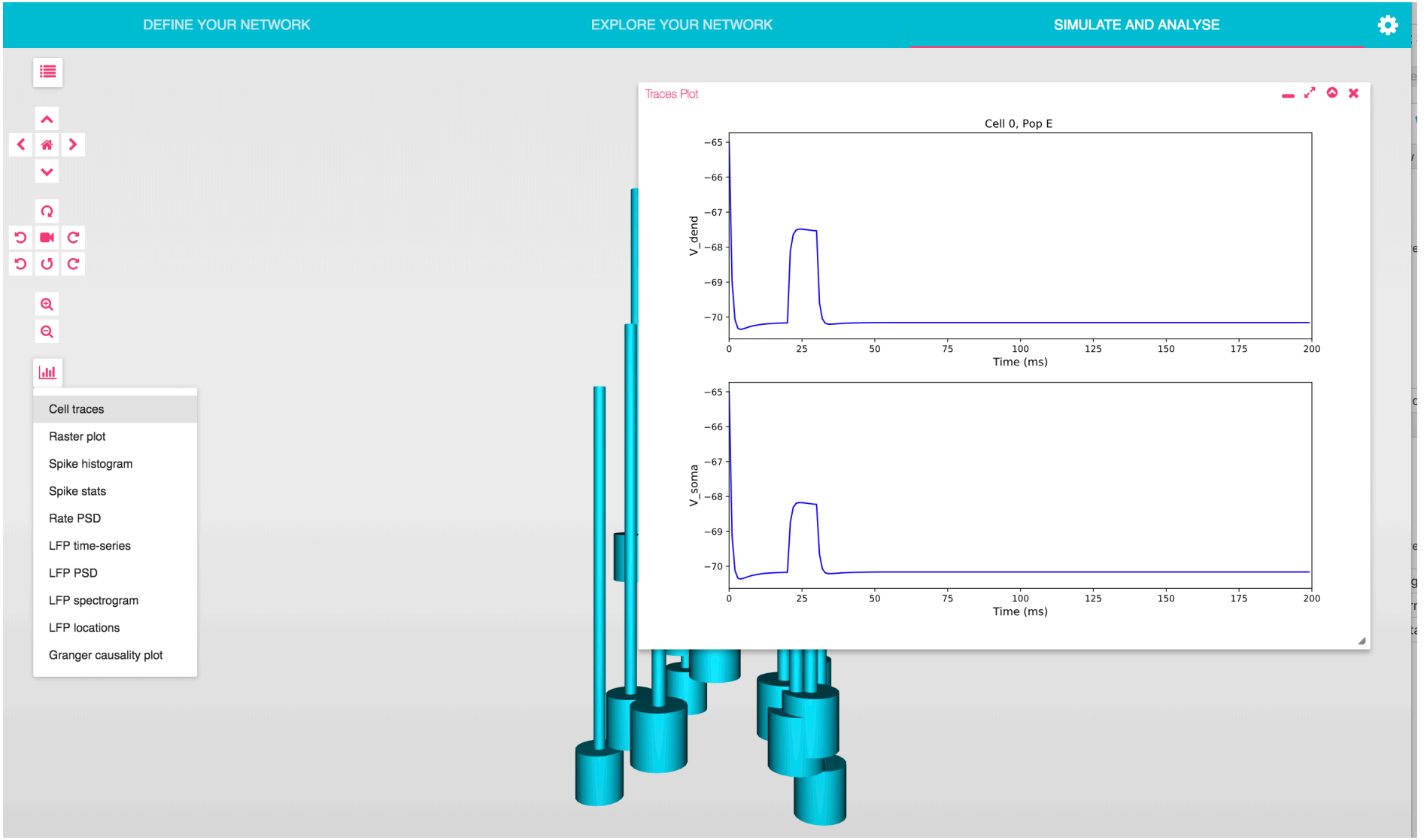


Specify trace s to record using Python dictionary format:

```
{'V_soma': {'var': 'v', 'sec': 'soma', 'loc': 0.5},
 'V_dend': {'var': 'v', 'sec': 'dend', 'loc': 1.0}}
```

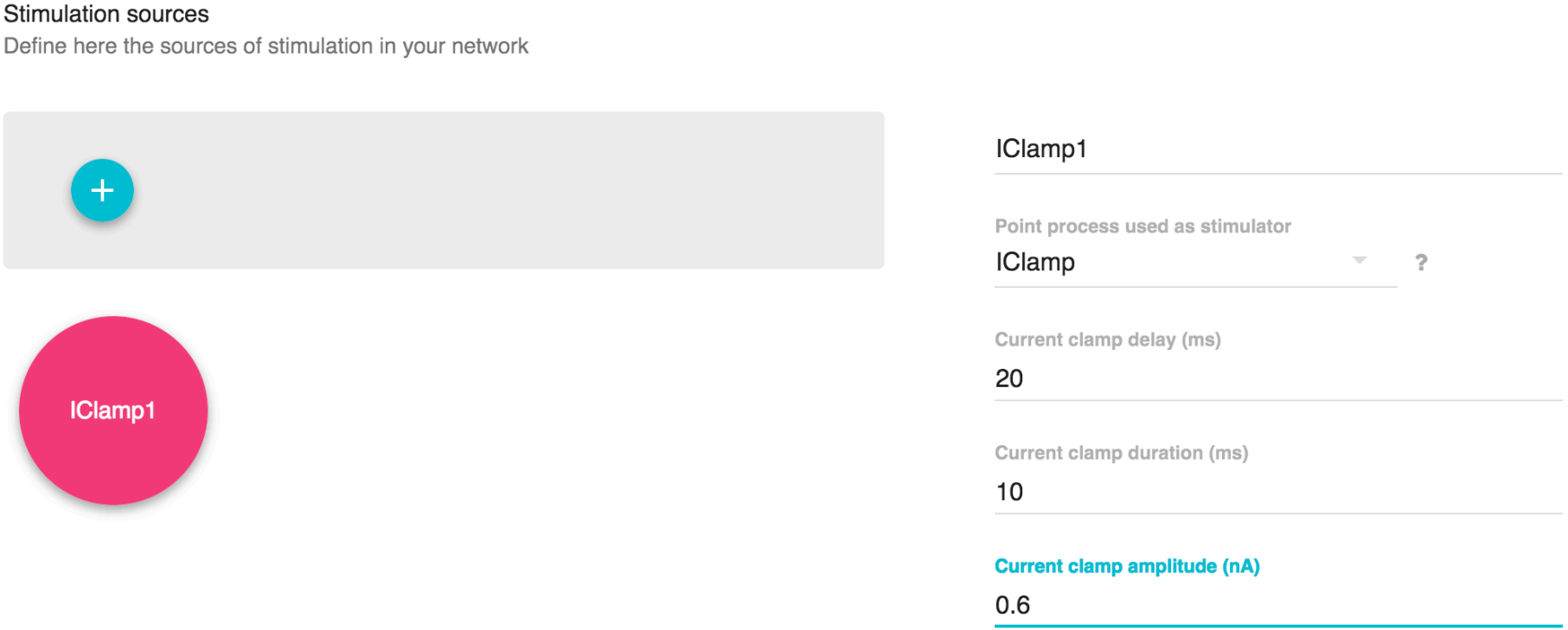
NetPyNE GUI Tutorial: Simple cell net

18) ‘Simulate and Analyze’ the network and plot ‘Cell traces’



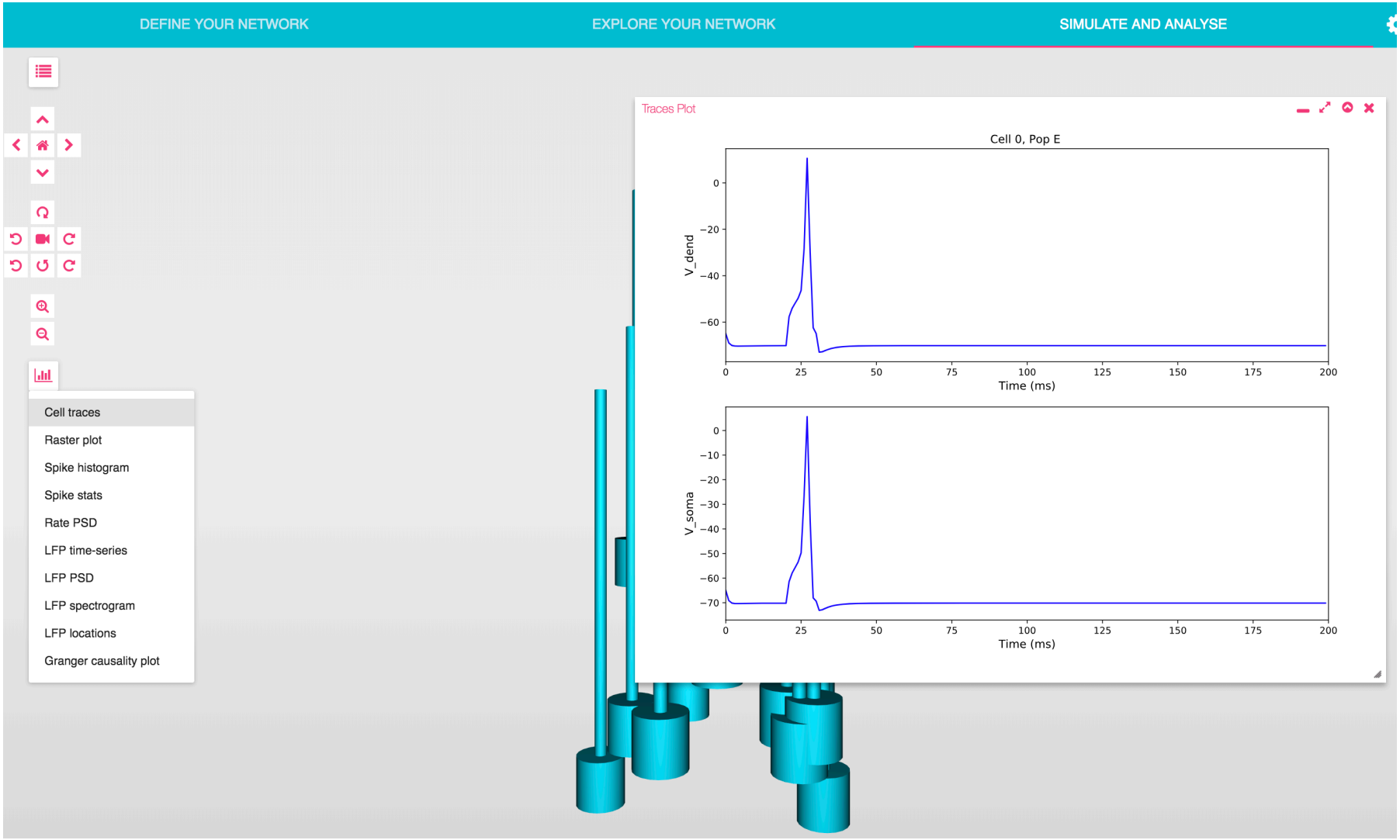
NetPyNE GUI Tutorial: Simple cell net

19) Increase 'IClamp1' amplitude so generate a spike (set to 0.6 nA)



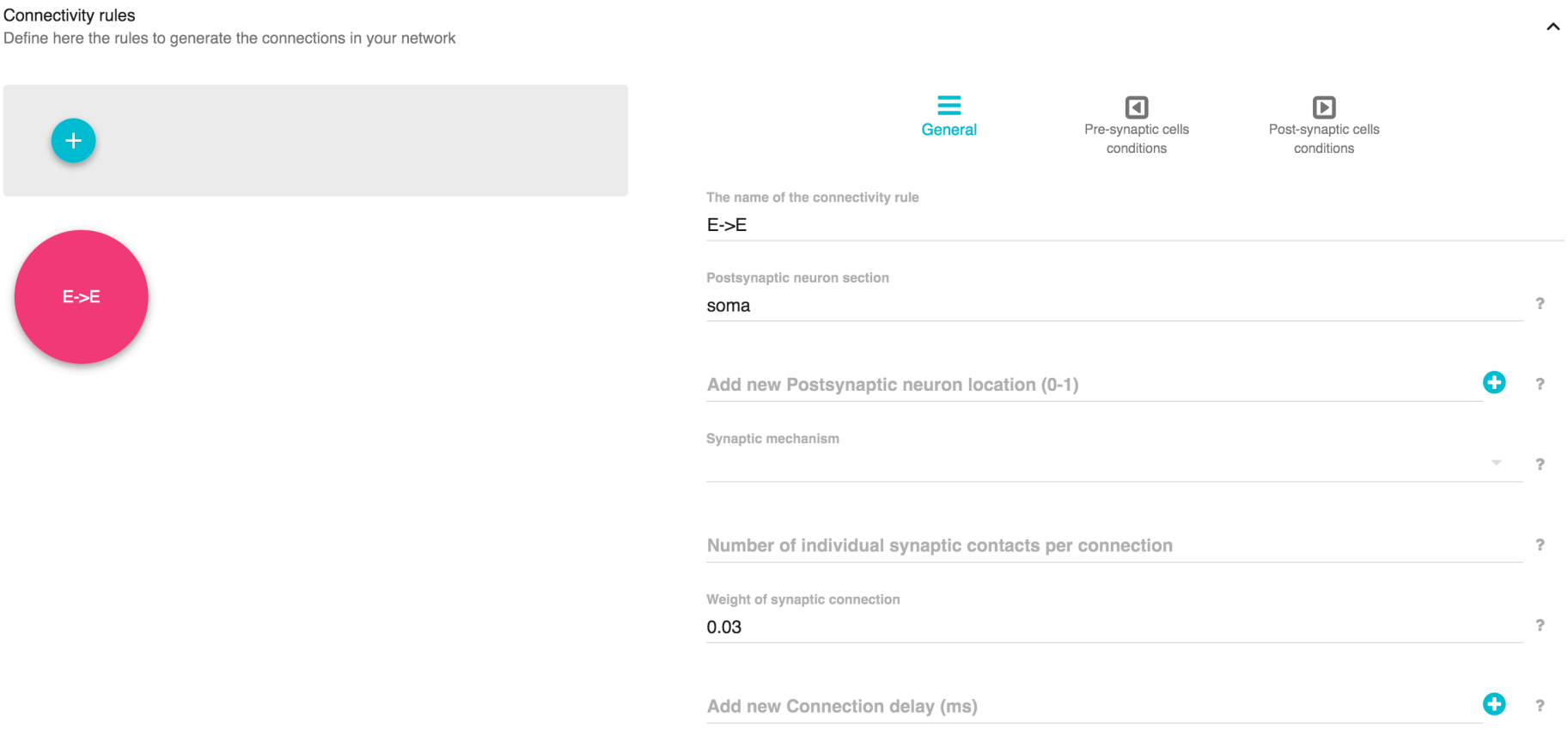
NetPyNE GUI Tutorial: Simple cell net

21) Simulate and plot traces (dendrite current clamp, soma spike and back-propagation to dendrite)



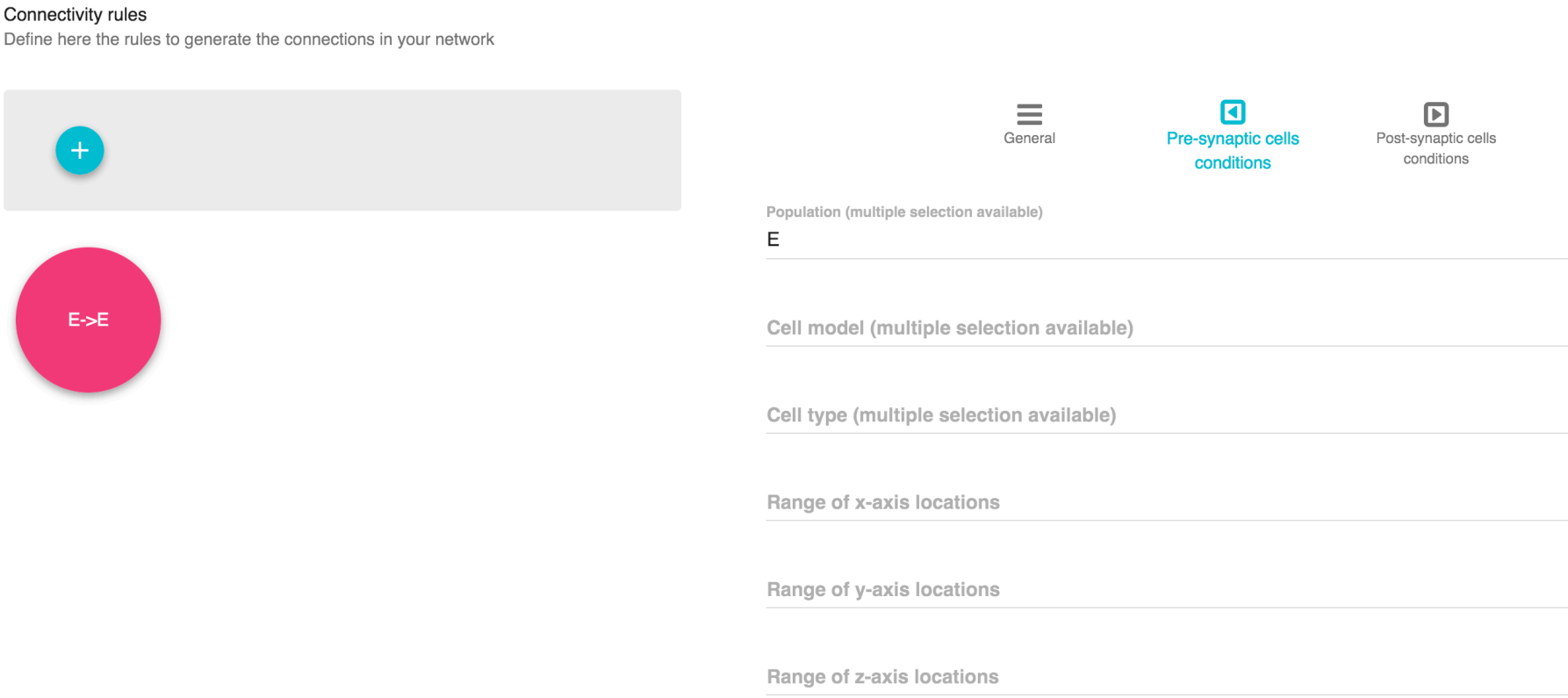
NetPyNE GUI Tutorial: Simple cell net

22a) Create recurrent connections (E->E) rule; first set weight to 0.03



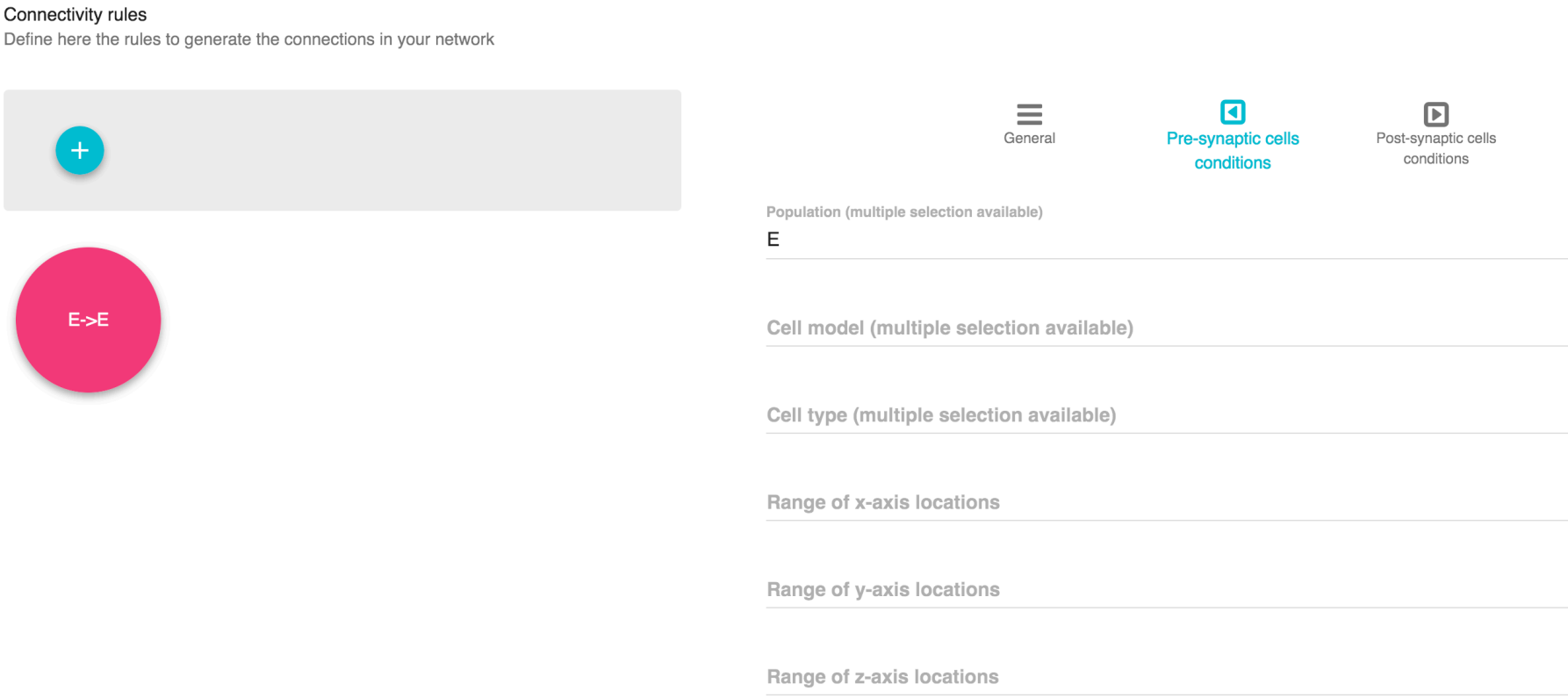
NetPyNE GUI Tutorial: Simple cell net

22b) Make presynaptic cells condition be ‘E’ population



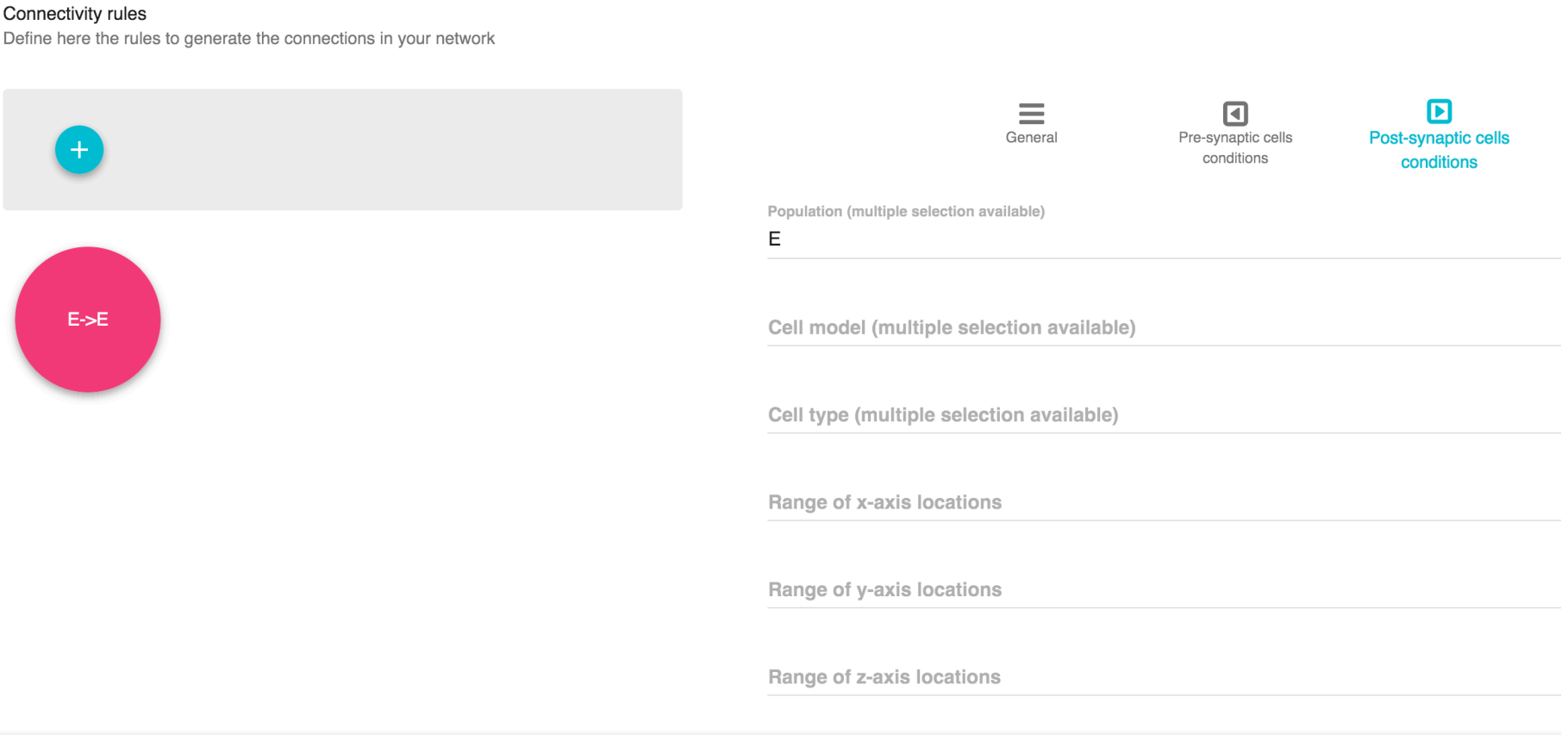
NetPyNE GUI Tutorial: Simple cell net

22b) Make presynaptic cells condition be ‘E’ population



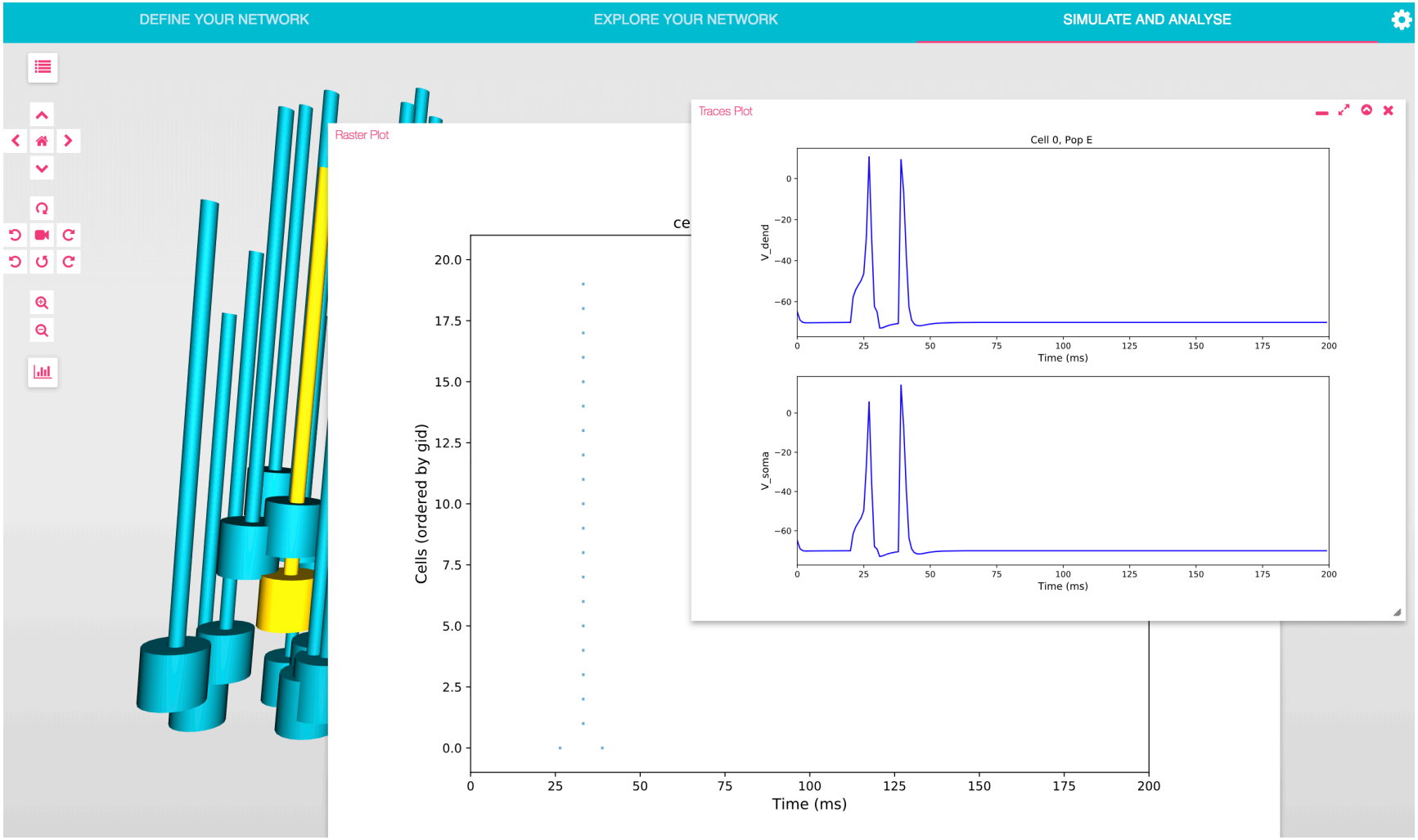
NetPyNE GUI Tutorial: Simple cell net

22c) Make postsynaptic cells condition be ‘E’ population



NetPyNE GUI Tutorial: Simple cell net

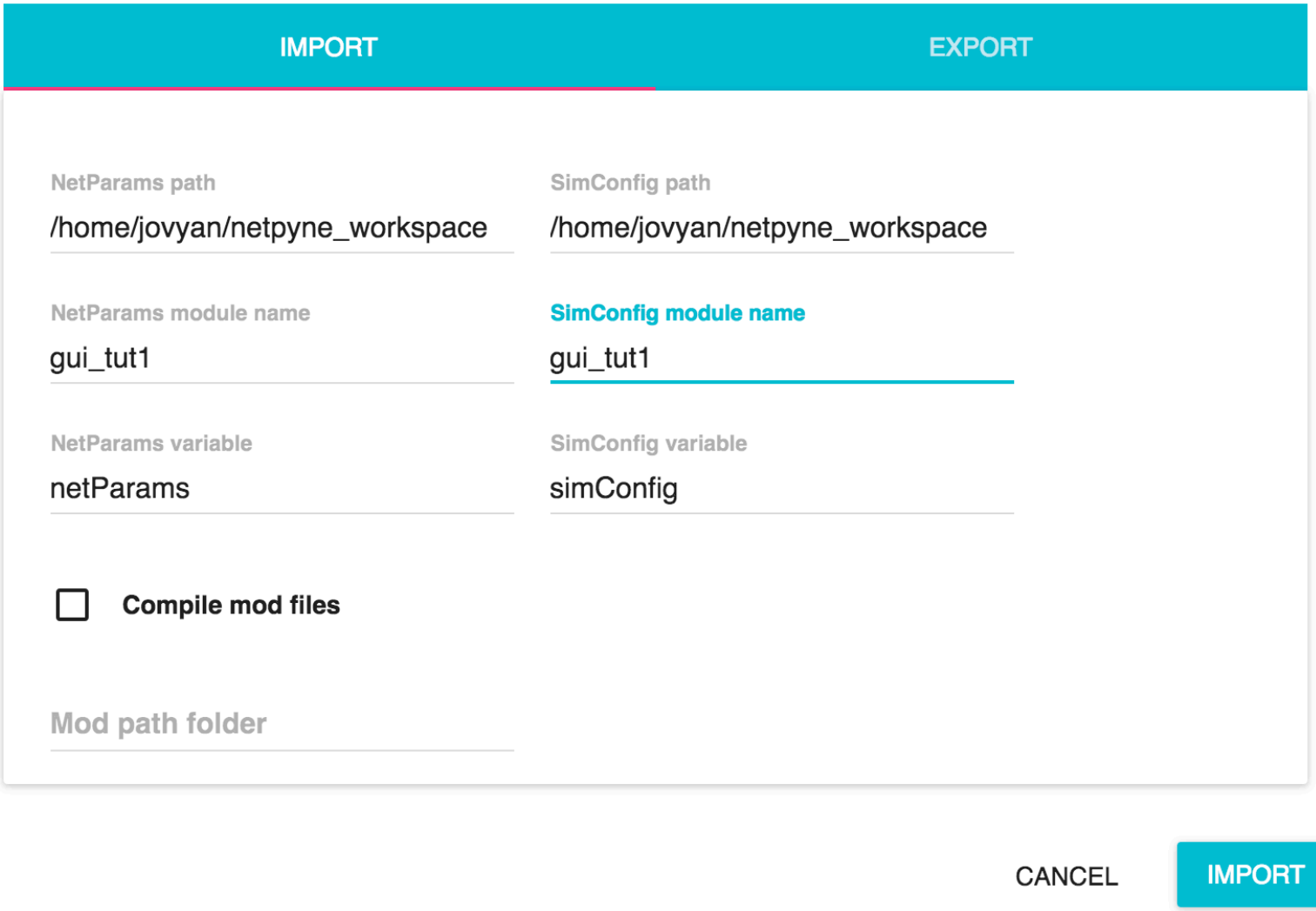
23) Simulate and plot traces and raster plot;
Cell 0 spikes due to IClamp -> triggers spikes in other cells due to conns -> cell 0 spikes again



NetPyNE GUI Tutorial: Simple cell net

Note: If you have any errors with step-by-step, try loading the **“simple cell net”** tutorial directly from file

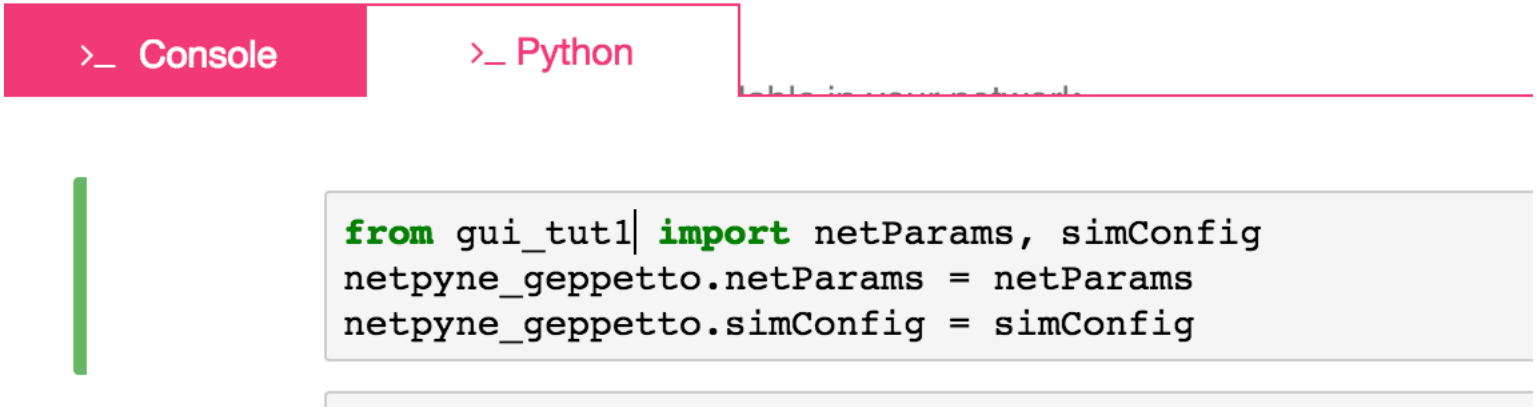
Method 1:



NetPyNE GUI Tutorial: Simple cell net

Note: If you have any errors with step-by-step, try loading the **"simple cell net"** tutorial directly from file

Method 2:



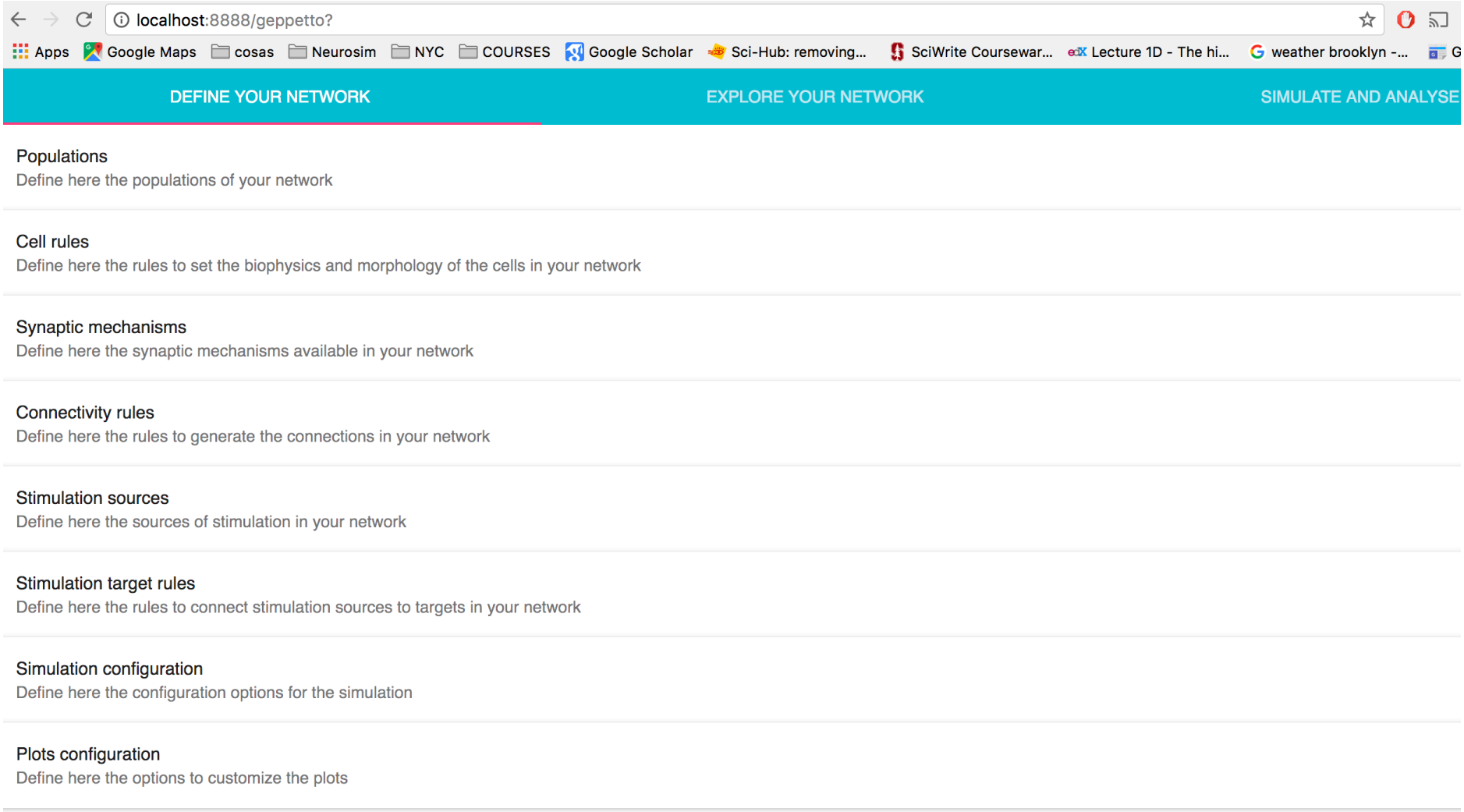
Copy paste from here:

```
from gui_tut1 import netParams, simConfig
netpyne_geppetto.netParams = netParams
netpyne_geppetto.simConfig = simConfig
```

To execute press Shift + Enter

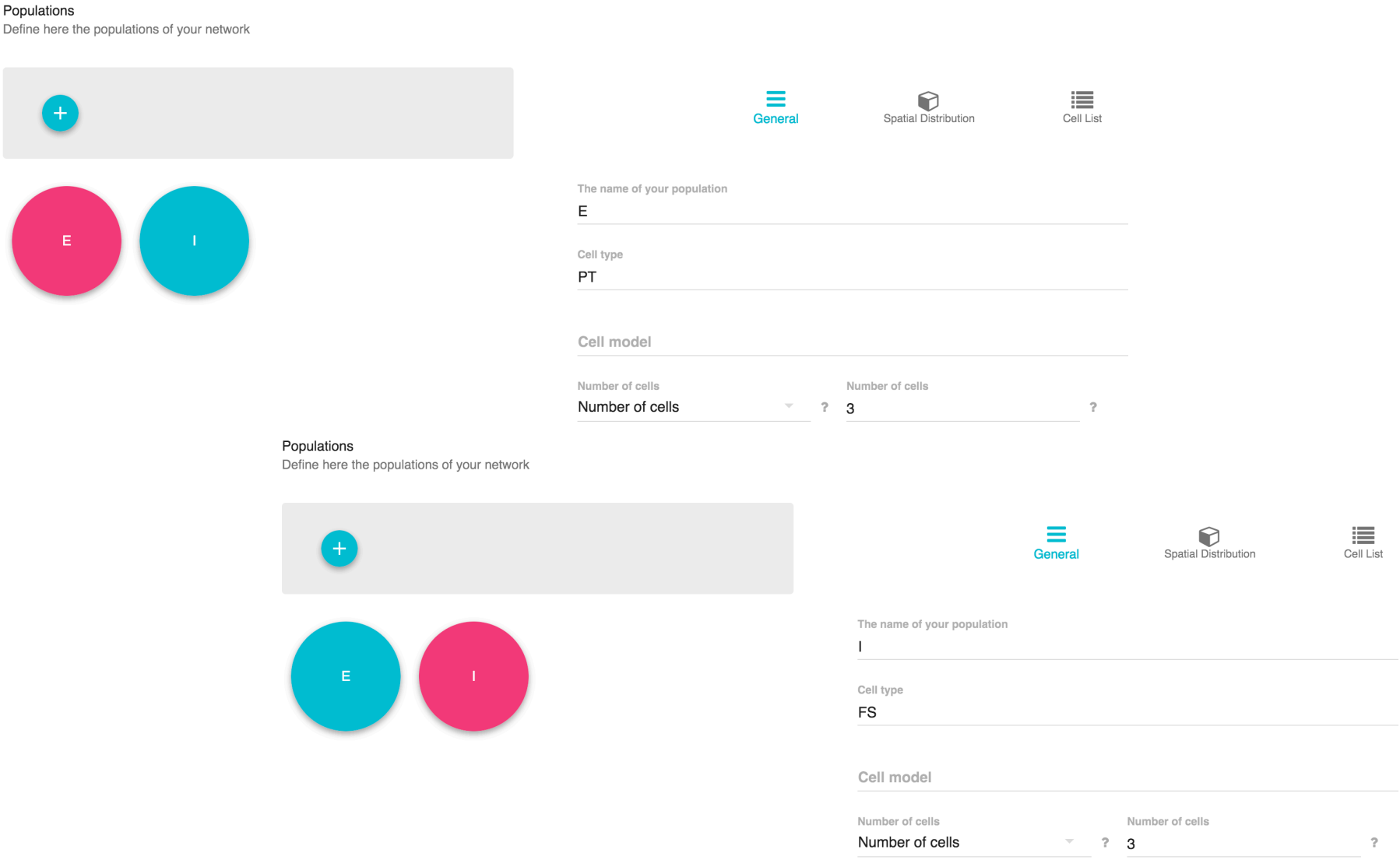
NetPyNE GUI Tutorial: Complex cell net

1) Reload webpage to start from scratch



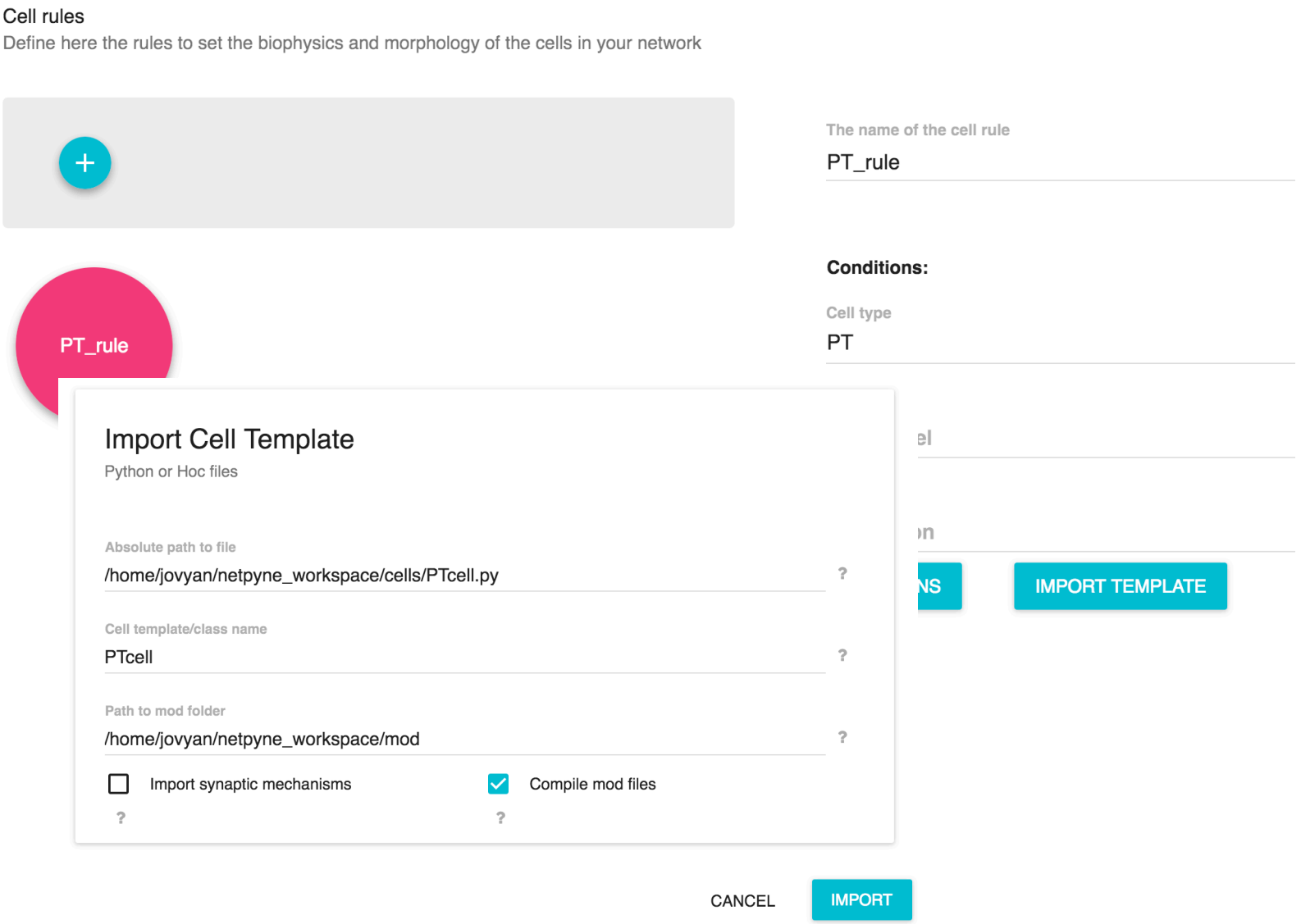
NetPyNE GUI Tutorial: Complex cell net

2) Add 2 populations of 3 cells : - 'E' (excitatory) of cell type 'PT' (pyramidal-tract corticospinal)
- 'I' (inhibitory) of cell type 'FS' (fast-spiking interneuron)



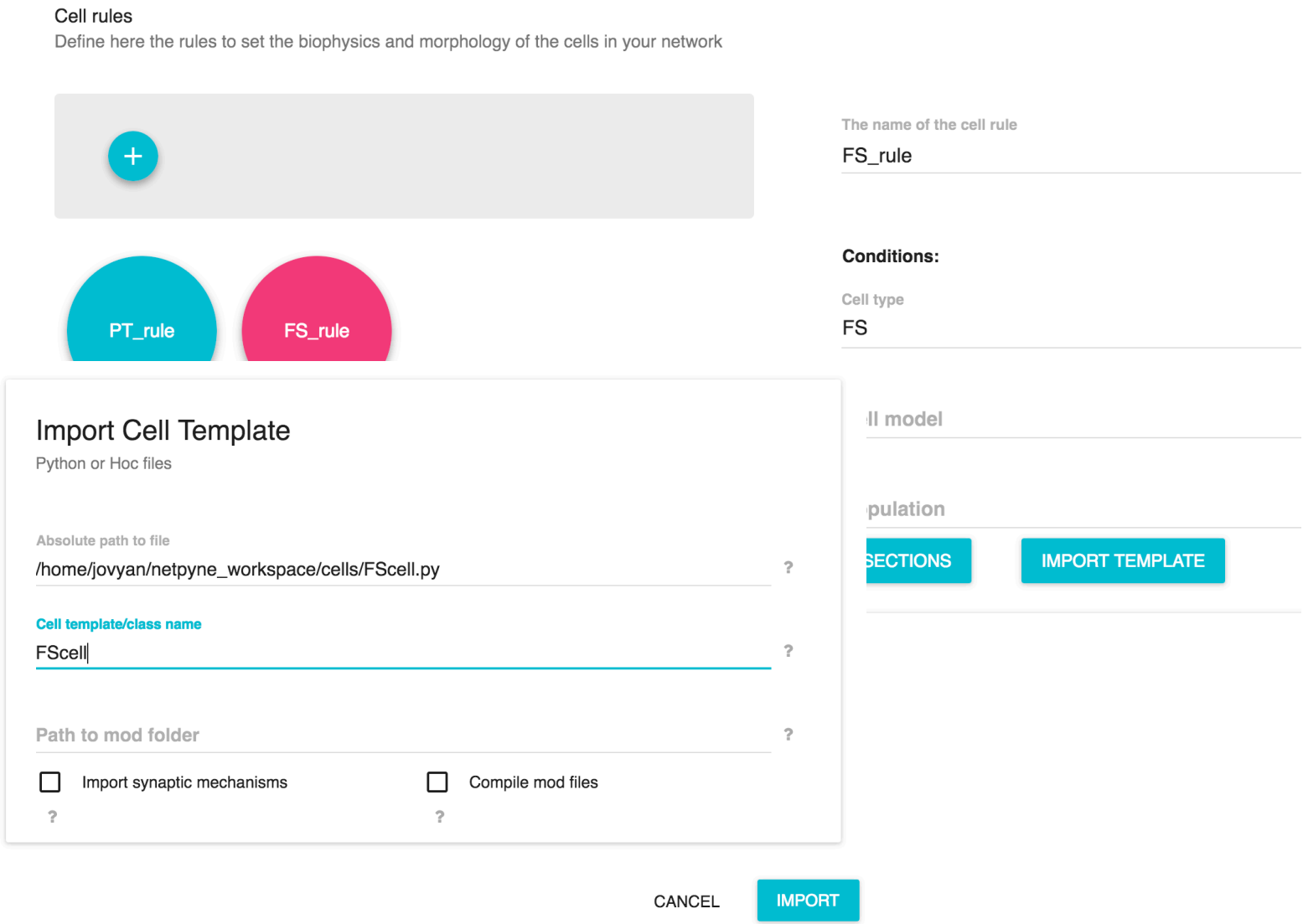
NetPyNE GUI Tutorial: Complex cell net

3) Create PT rule and import from template (need to copy files to workspace)



NetPyNE GUI Tutorial: Complex cell net

4) Create FS rule and import from template (need to copy files to workspace)



NetPyNE GUI Tutorial: Complex cell net

5) Visualize network

NetPyNE GUI Tutorial: Complex cell net

6) Add AMPA and GABA synapses

Synaptic mechanisms

Define here the synaptic mechanisms available in your network

AMPA

NMODL mechanism name

Exp2Syn ?

Time constant for exponential 1 (ms)

0.5

Time constant for exponential 2 (ms)

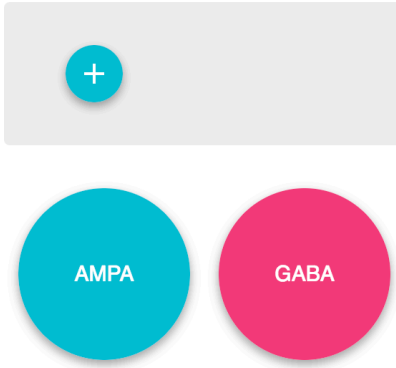
1

Reversal potential (mV)

0

Synaptic mechanisms

Define here the synaptic mechanisms available in your network



GABA

NMODL mechanism name

Exp2Syn ?

Time constant for exponential 1 (ms)

0.5

Time constant for exponential 2 (ms)

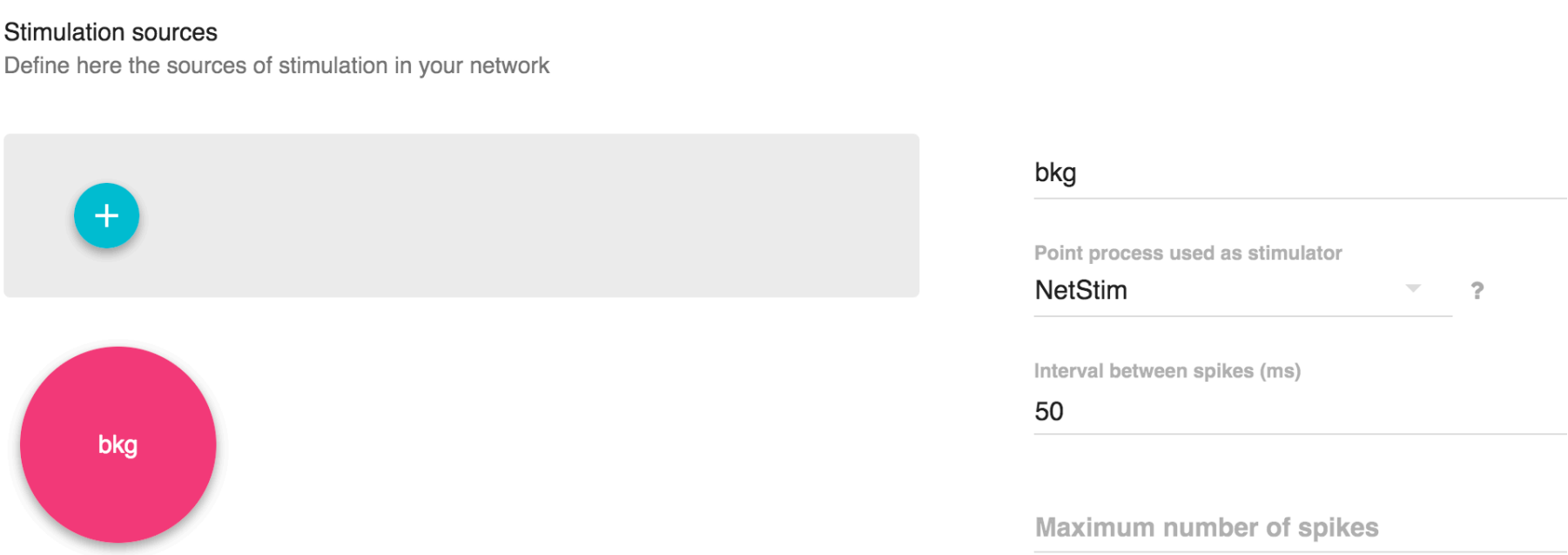
1

Reversal potential (mV)

-100

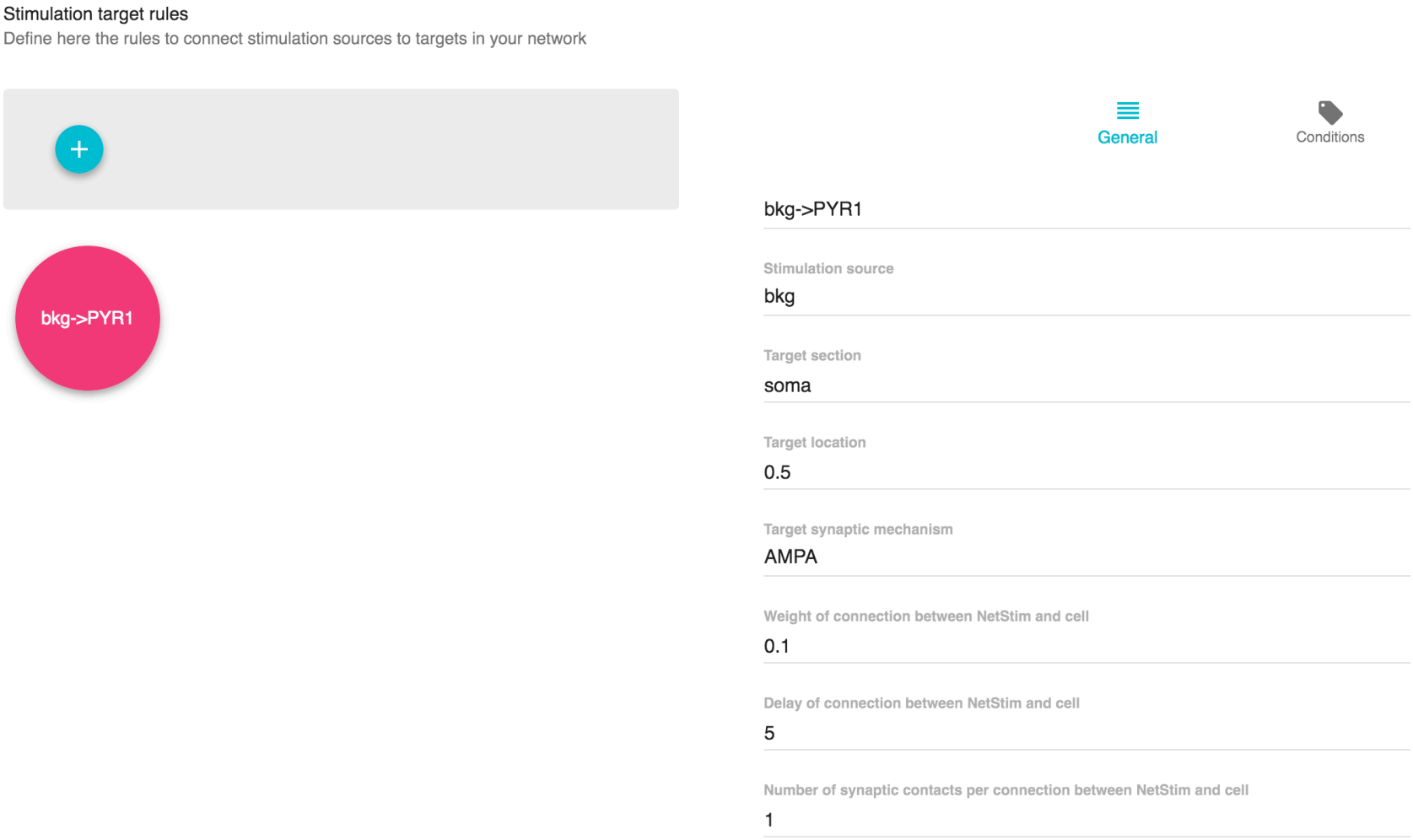
NetPyNE GUI Tutorial: Complex cell net

7a) Add background stimulation Netstim (spike generator) to PT cells



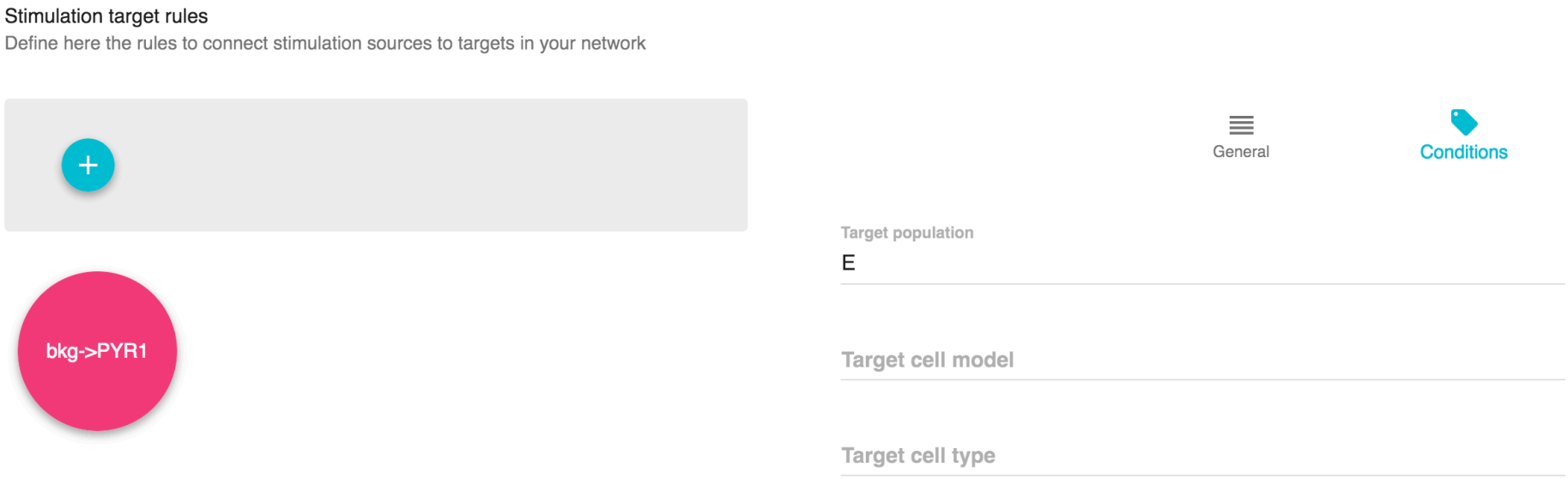
NetPyNE GUI Tutorial: Complex cell net

7b) Add background stimulation Netstim (spike generator) to PT cells



NetPyNE GUI Tutorial: Complex cell net

7c) Add background stimulation Netstim (spike generator) to PT cells (E population)

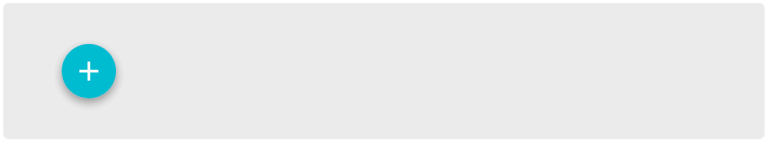


NetPyNE GUI Tutorial: Complex cell net

8) Connect E->I

Connectivity rules

Define here the rules to generate the connections in your network



General | Pre-synaptic cells conditions | Post-synaptic cells conditions

The name of the connectivity rule

E->I

Postsynaptic neuron section

soma

Add new Postsynaptic neuron location (0-1)

0.5

Synaptic mechanism

Number of individual synaptic contacts per connection

1

Weight of synaptic connection

0.03

Add new Connection delay (ms)

5

General | Pre-synaptic cells conditions | Post-synaptic cells conditions

Population (multiple selection available)

E

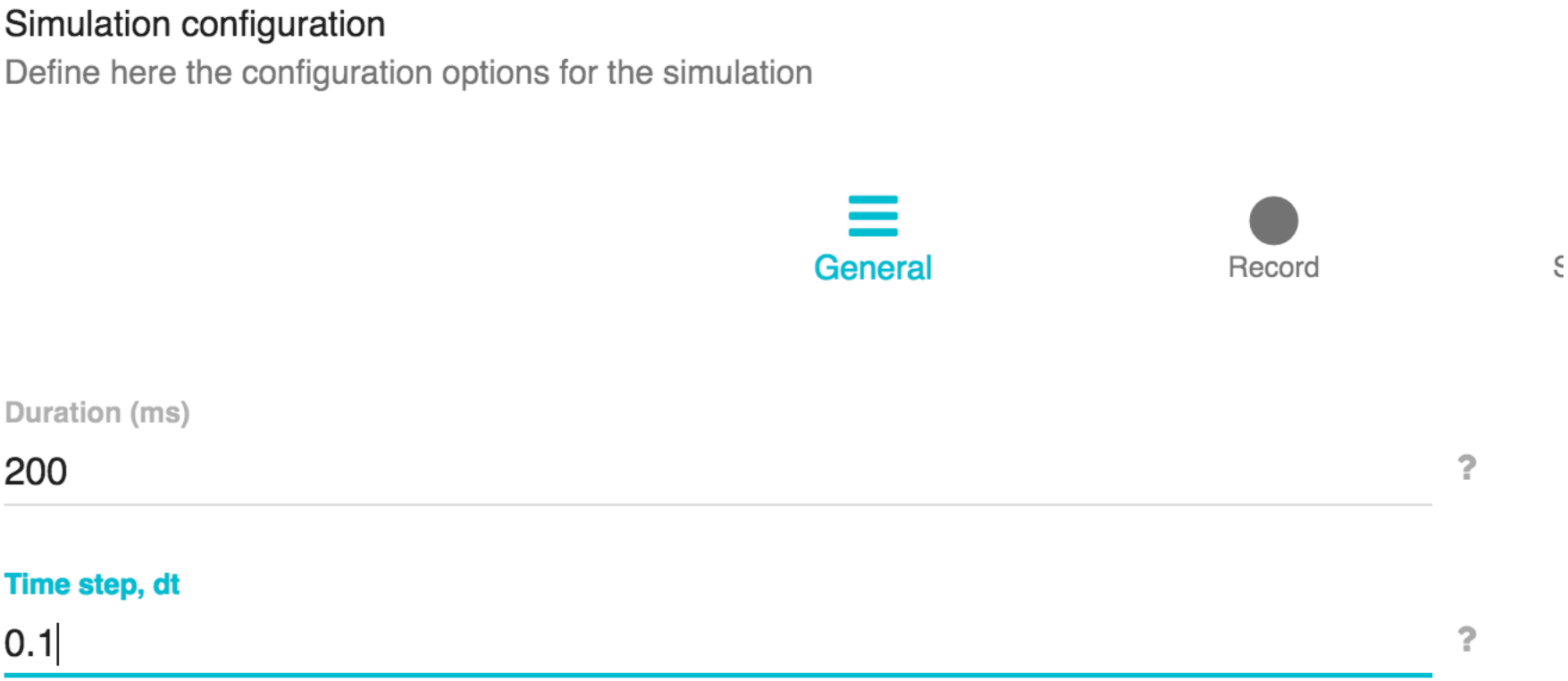
General | Pre-synaptic cells conditions | Post-synaptic cells conditions

Population (multiple selection available)

I

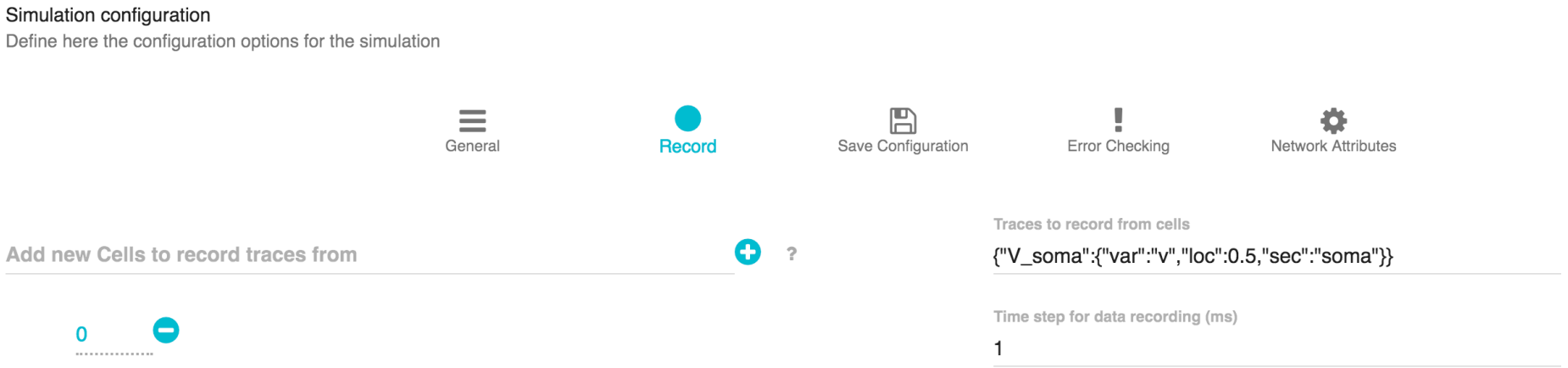
NetPyNE GUI Tutorial: Complex cell net

9) Set duration to 200 ms and time step to 0.1



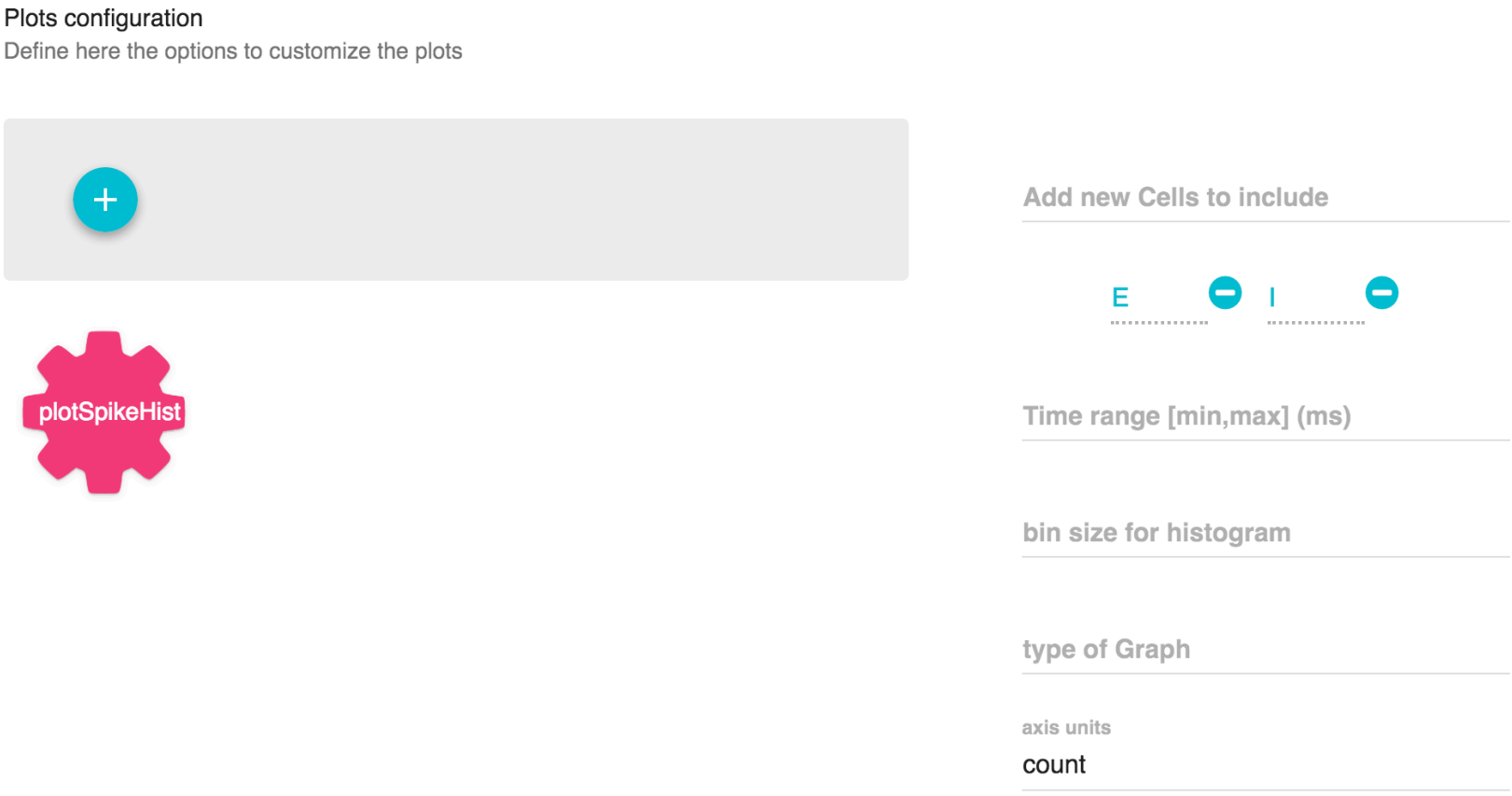
NetPyNE GUI Tutorial: Complex cell net

10) Record voltate trace from cell 0



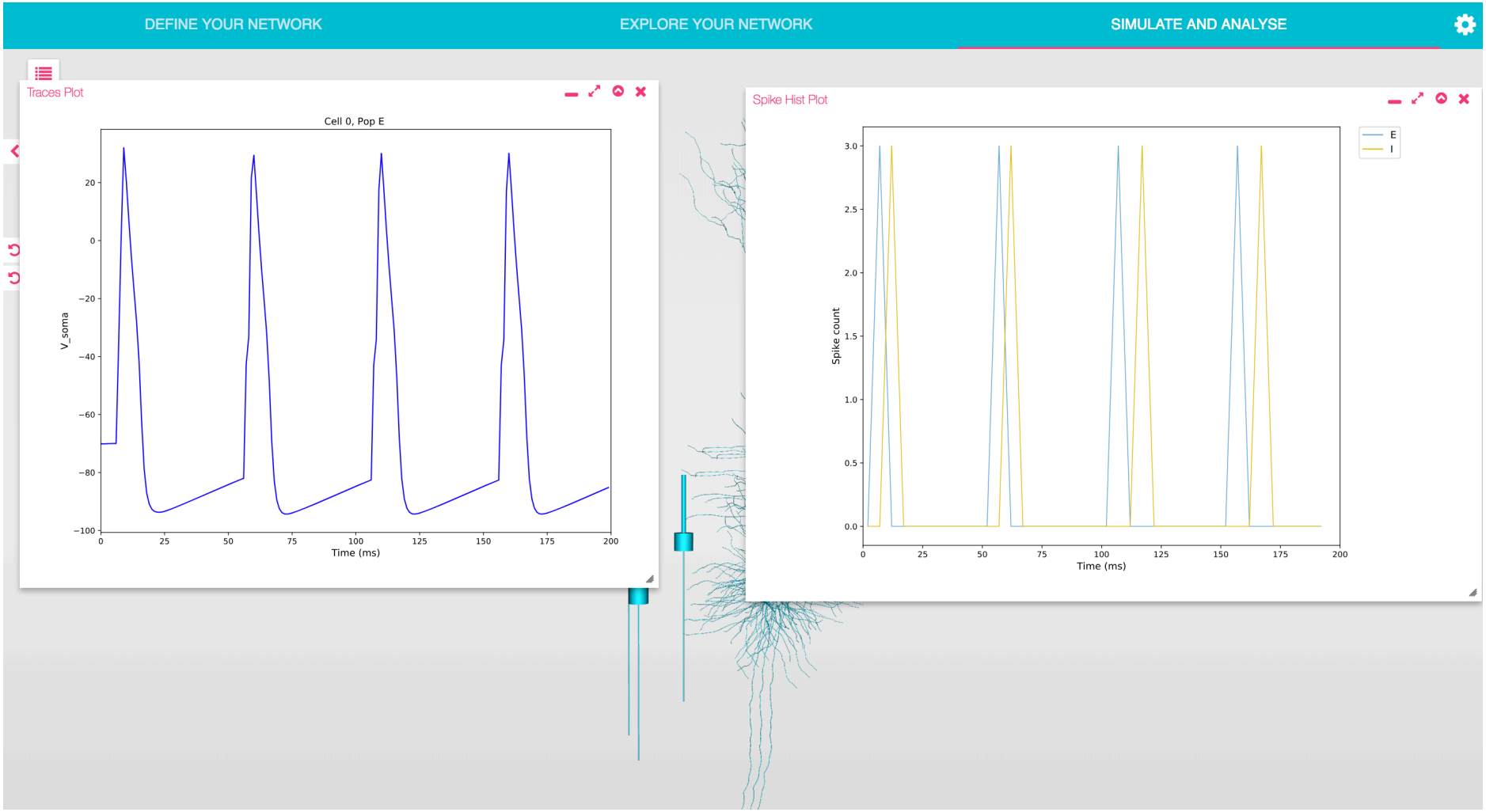
NetPyNE GUI Tutorial: Complex cell net

11) Set spike histogram options



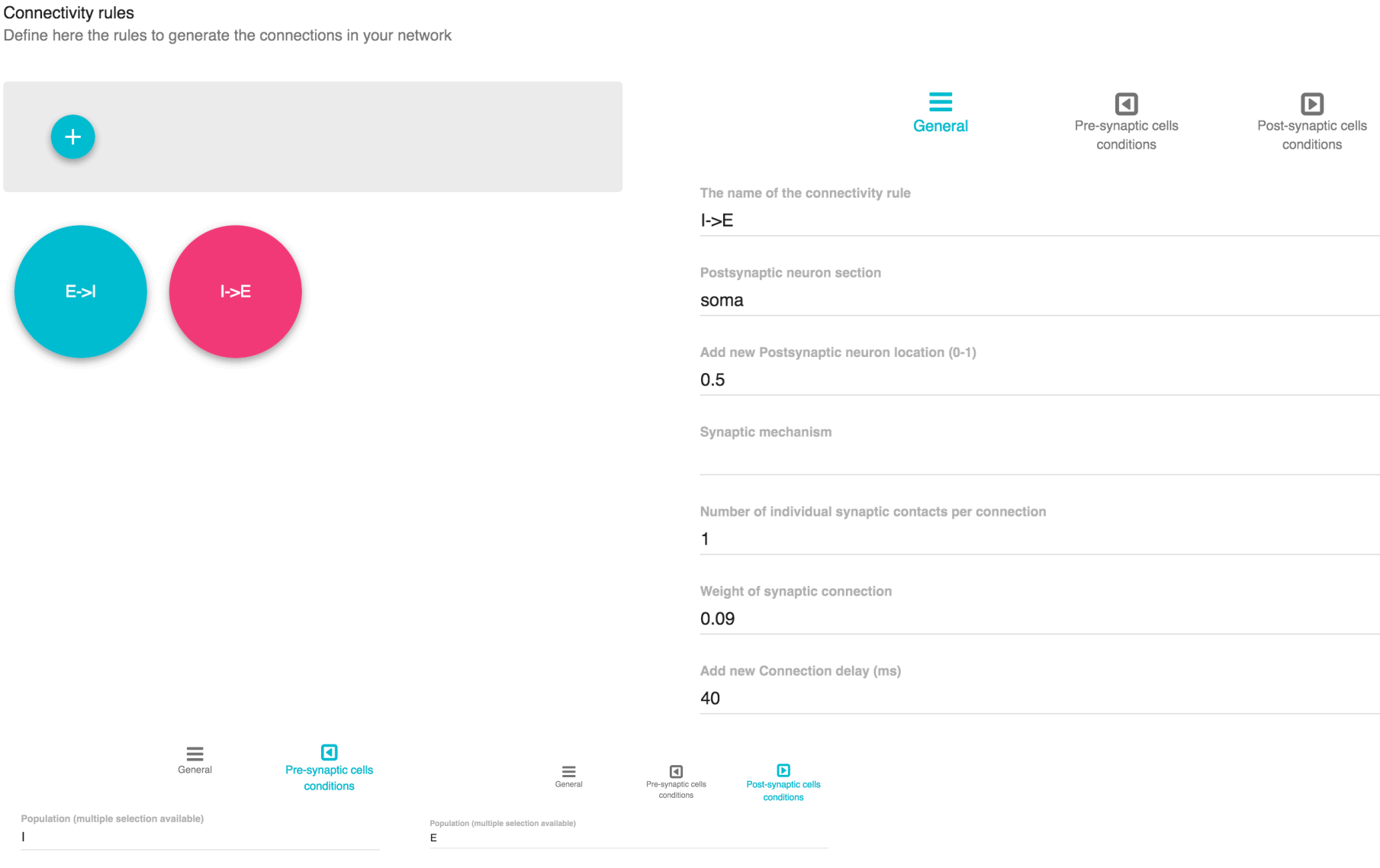
NetPyNE GUI Tutorial: Complex cell net

11) Simulate and visualize traces and spike histogram



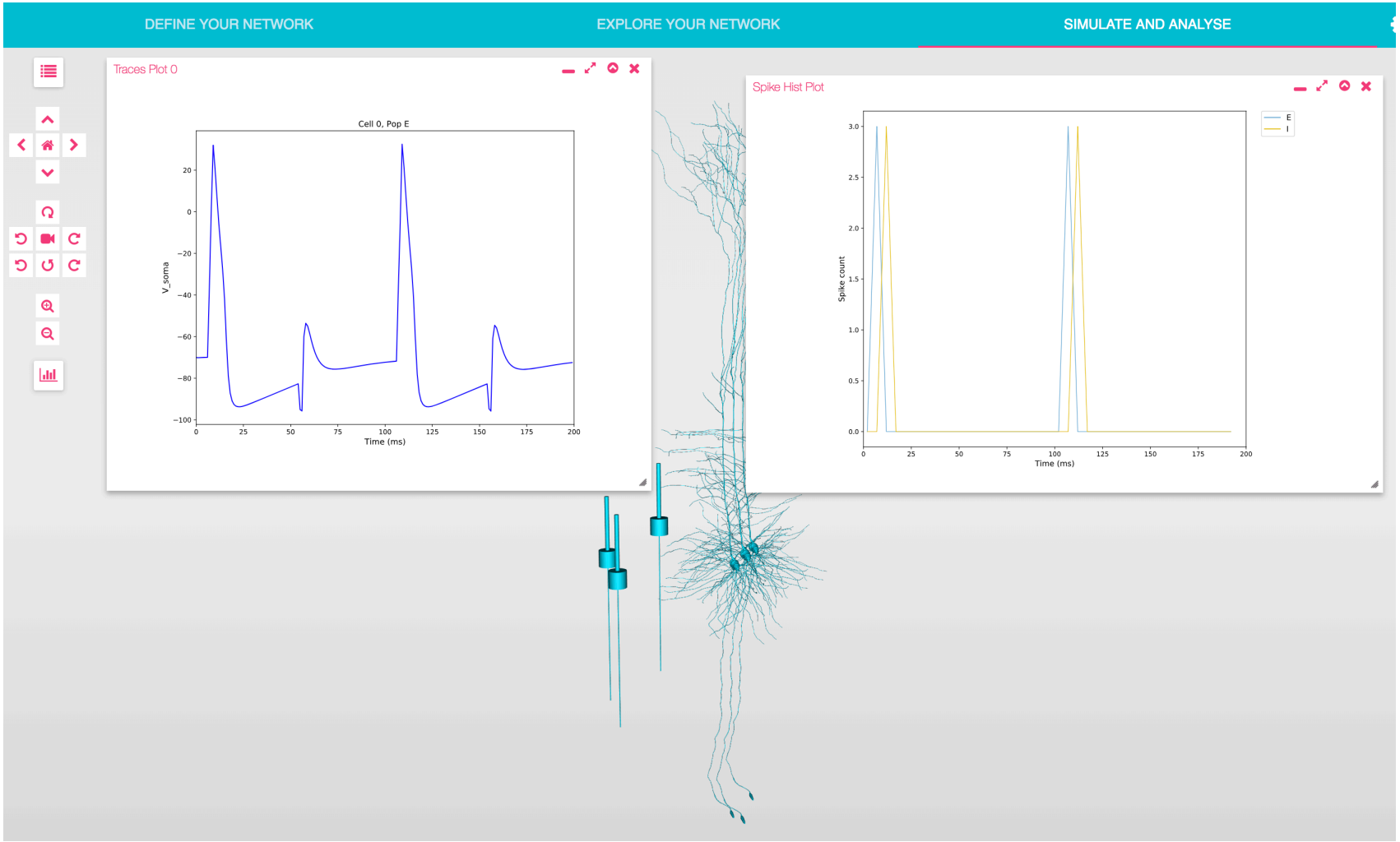
NetPyNE GUI Tutorial: Complex cell net

12) Connect I->E with delay of 40ms



NetPyNE GUI Tutorial: Complex cell net

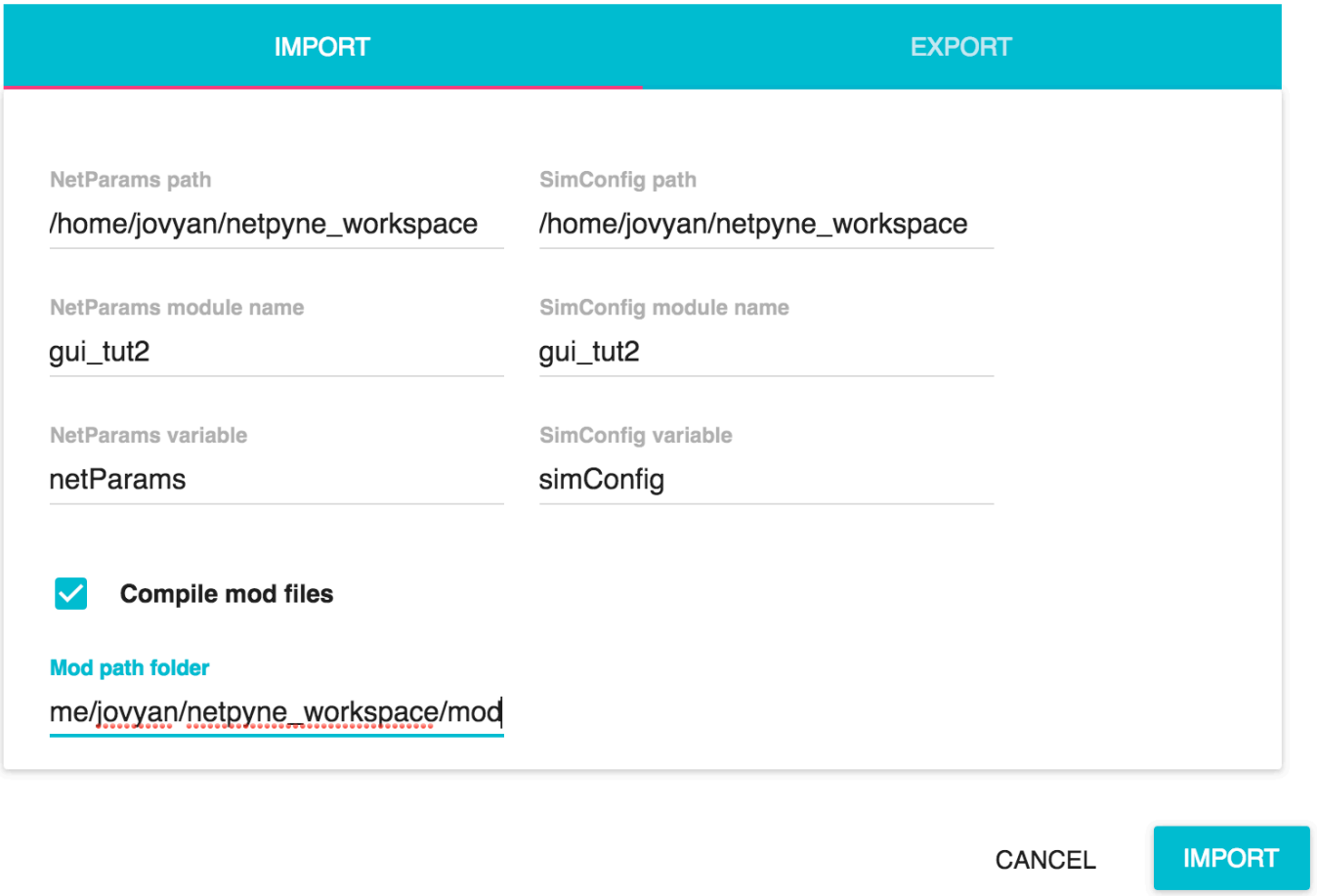
13) Simulate and plot traces and spike histogram



NetPyNE GUI Tutorial: Complex cell net

Note: If you have any errors with step-by-step, try loading the **“complex cell net”** tutorial directly from file

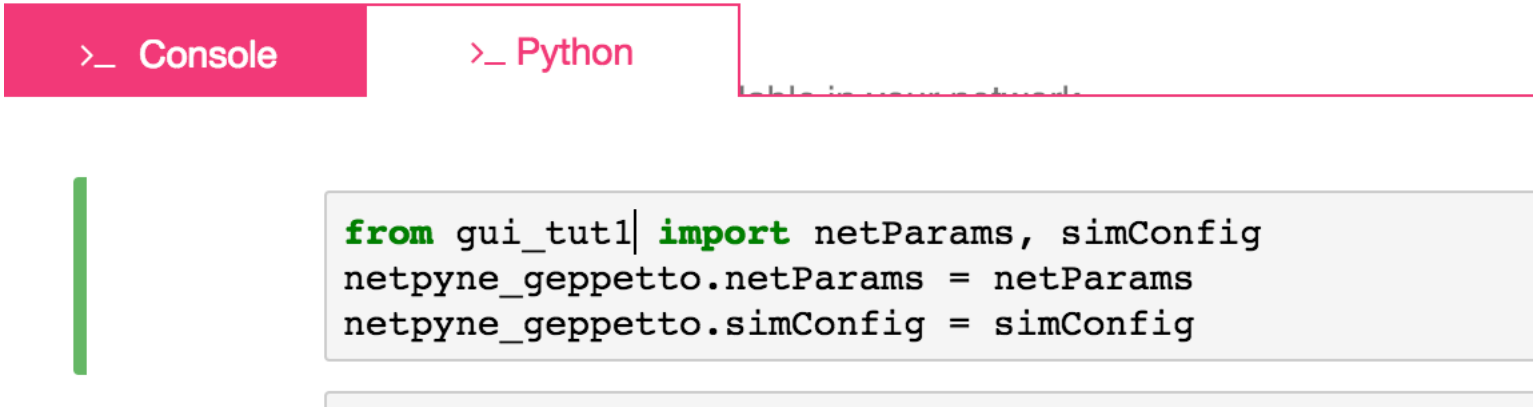
Method 1:



NetPyNE GUI Tutorial: Complex cell net

Note: If you have any errors with step-by-step, try loading the **"simple cell net"** tutorial directly from file

Method 2:



Copy paste from here:

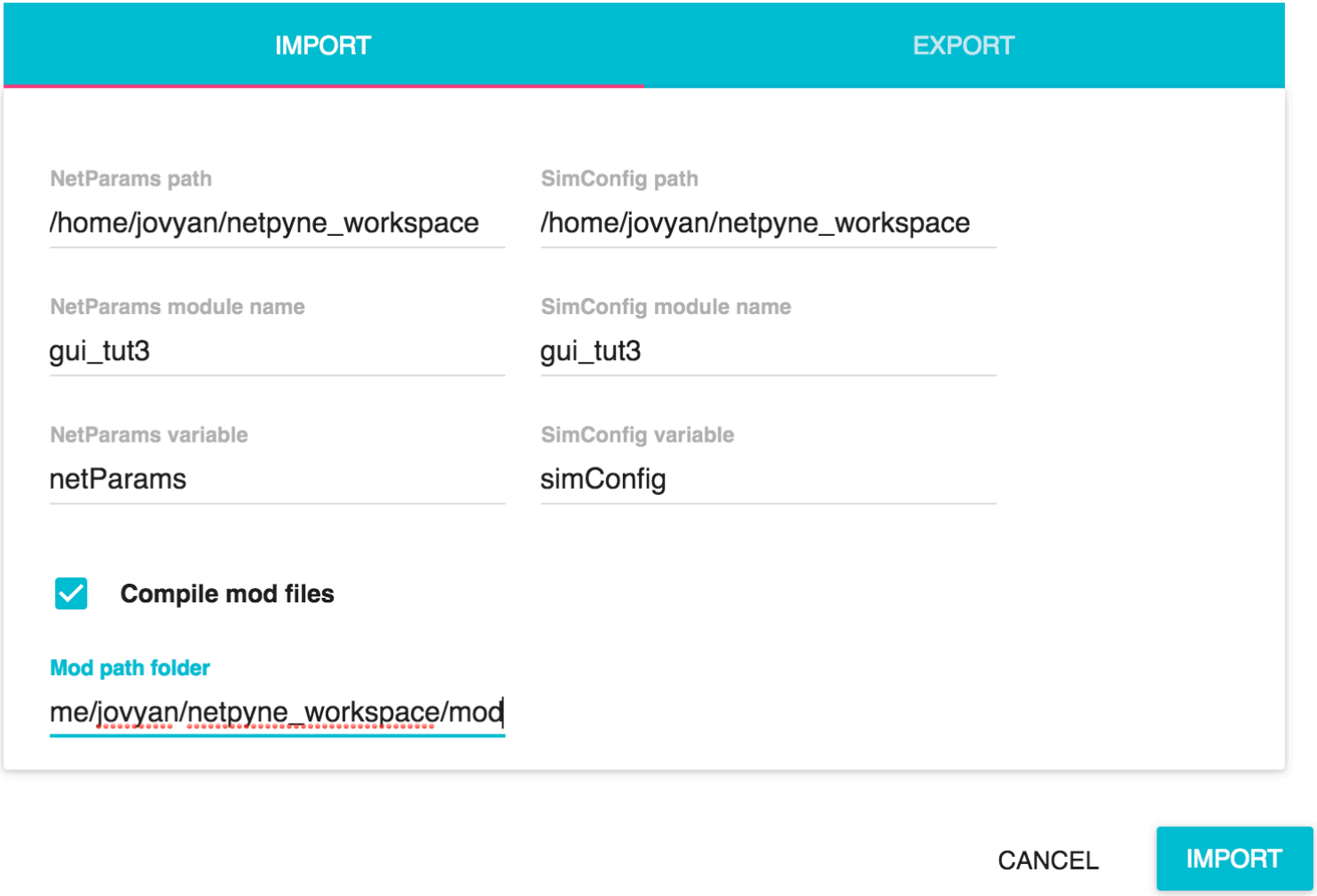
```
from gui_tut2 import netParams, simConfig
netpyne_geppetto.netParams = netParams
netpyne_geppetto.simConfig = simConfig
```

To execute press Shift + Enter

NetPyNE GUI Tutorial: Multilayer net

Load the **“multilayer net”** tutorial directly from file

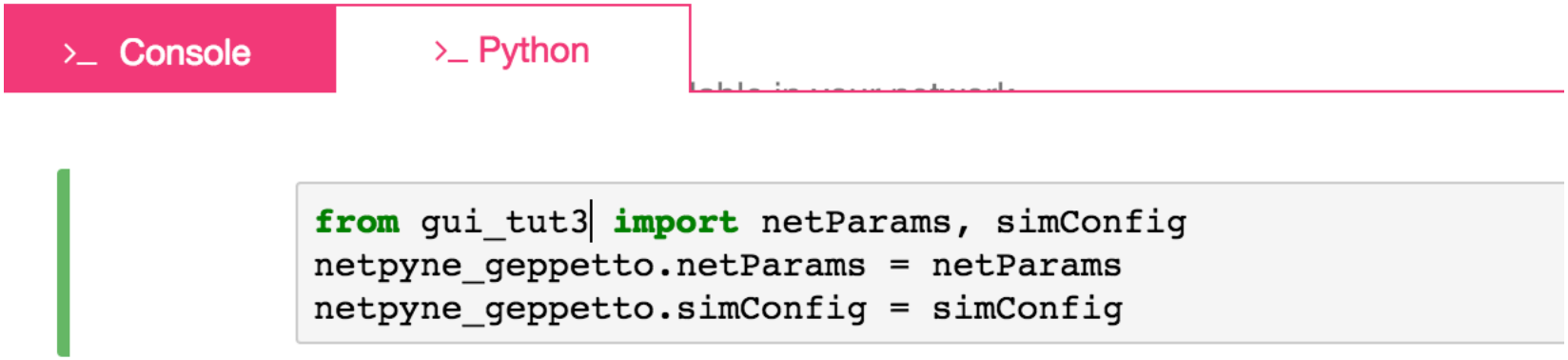
Method 1:



NetPyNE GUI Tutorial: Multilayer net

Load the **“multilayer cell net”** tutorial directly from file

Method 2:



Copy paste from here:

```
from gui_tut3 import netParams, simConfig
netpyne_geppetto.netParams = netParams
netpyne_geppetto.simConfig = simConfig
```

To execute press Shift + Enter

NetPyNE GUI Tutorial: Multilayer net

1) Simulate and plot raster and LFP