Model representation approaches for the Allen mouse visual column

CodeJam Workshop

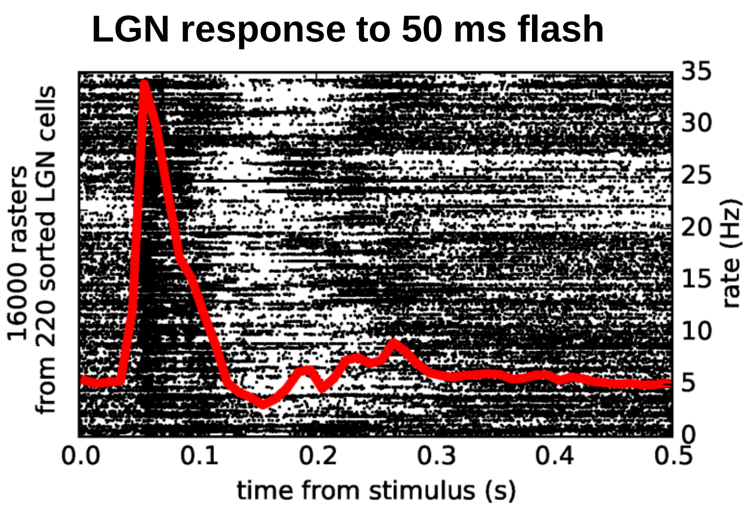
January 11, 2016

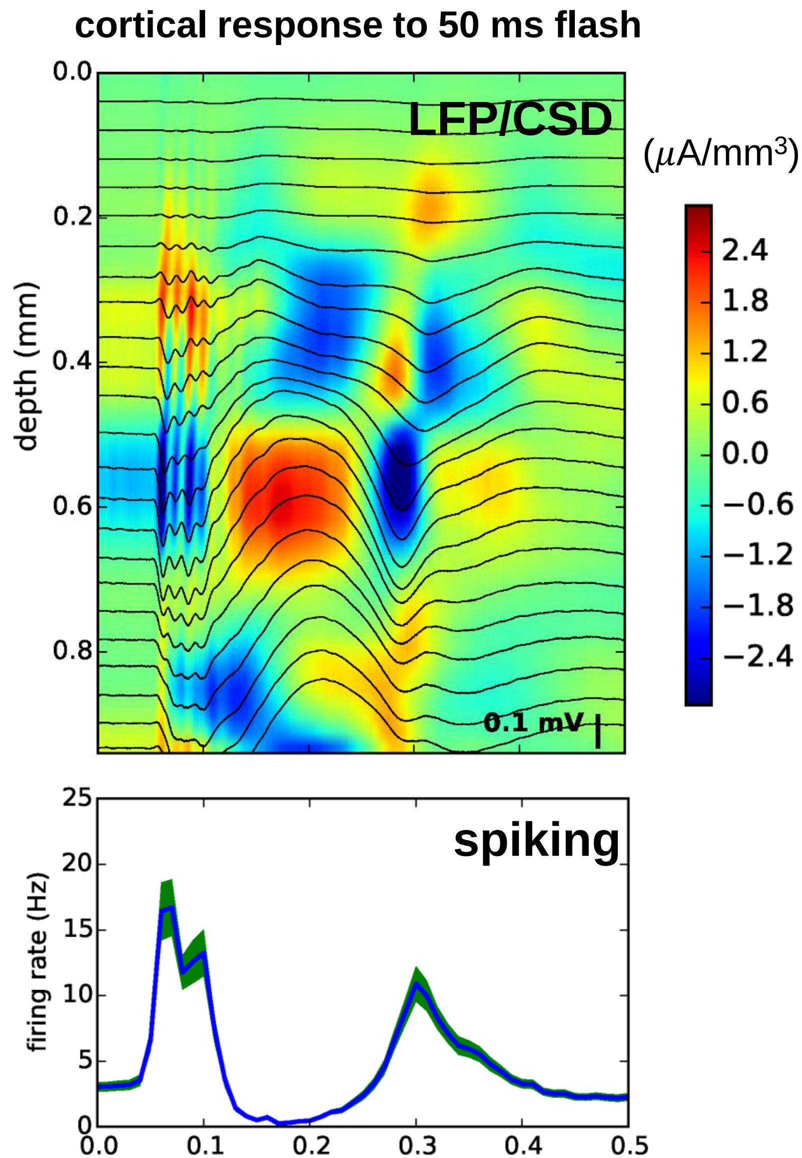
Sergey Gratiy
Sci I, MAT

ALLEN INSTITUTE *for* BRAIN SCIENCE
Fueling Discovery

Aim

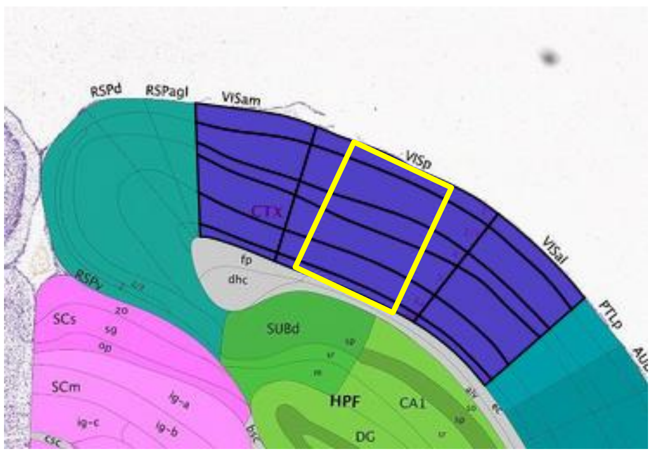
To infer the flow of neuronal activity in mouse V1 which underlies the extracellular signal produced in response to a set of visual stimuli.

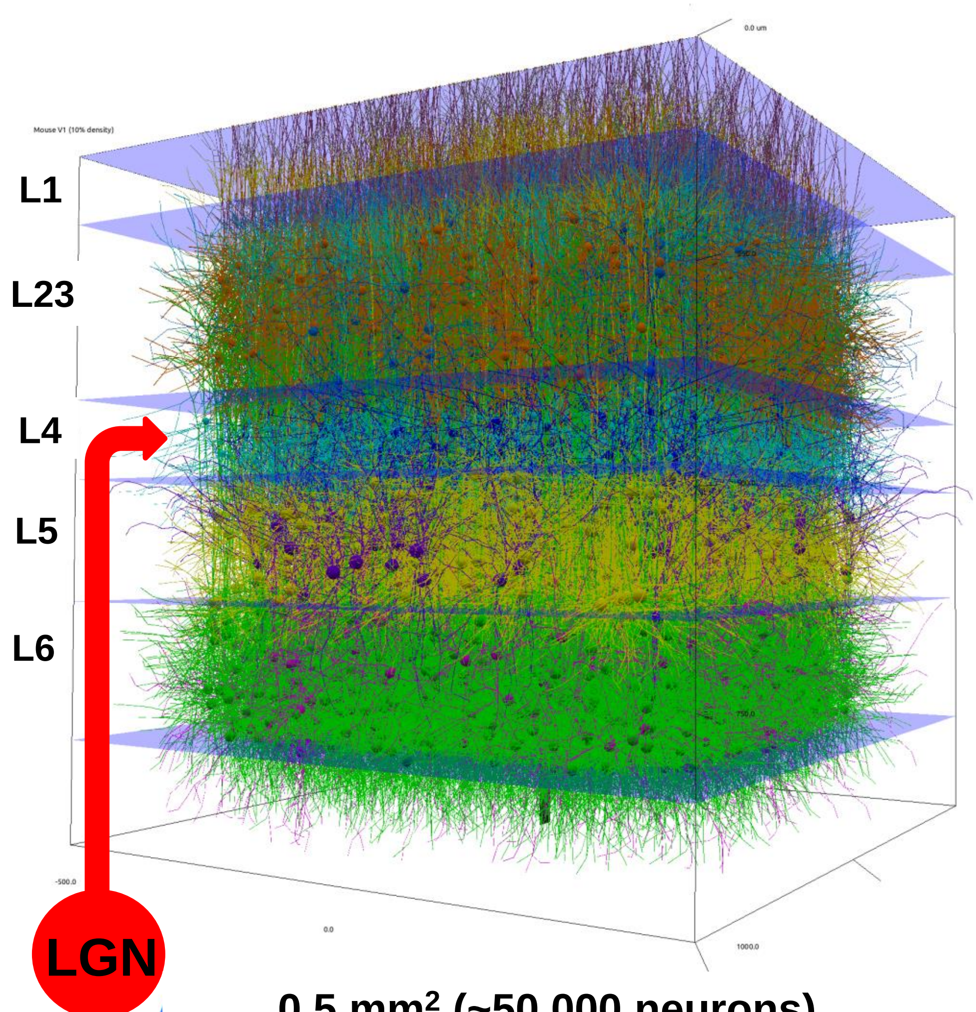




Approach

Build the biophysically detailed network model of mouse V1 detailed enough to predict selected set of in vivo data based on the LGN input.





0.5 mm^2 (~50,000 neurons)

Components of a network model

1. Cells
2. Connections (recurrent and external)
3. External inputs

Components of a network model

1. Cells

2. Connections (recurrent and external)

3. External inputs

Requirement for the approach:

- Could be used for networks at various levels of detail: biophysical => point

- Use standard software tools and file formats

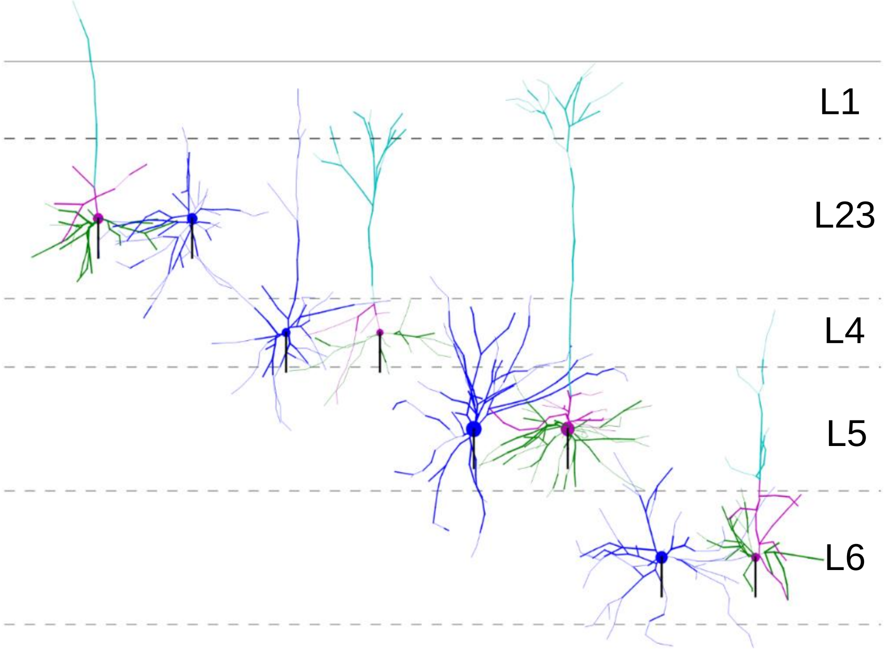
Cell parameters

Individual cells are instance of neuronal models of a particular cell type

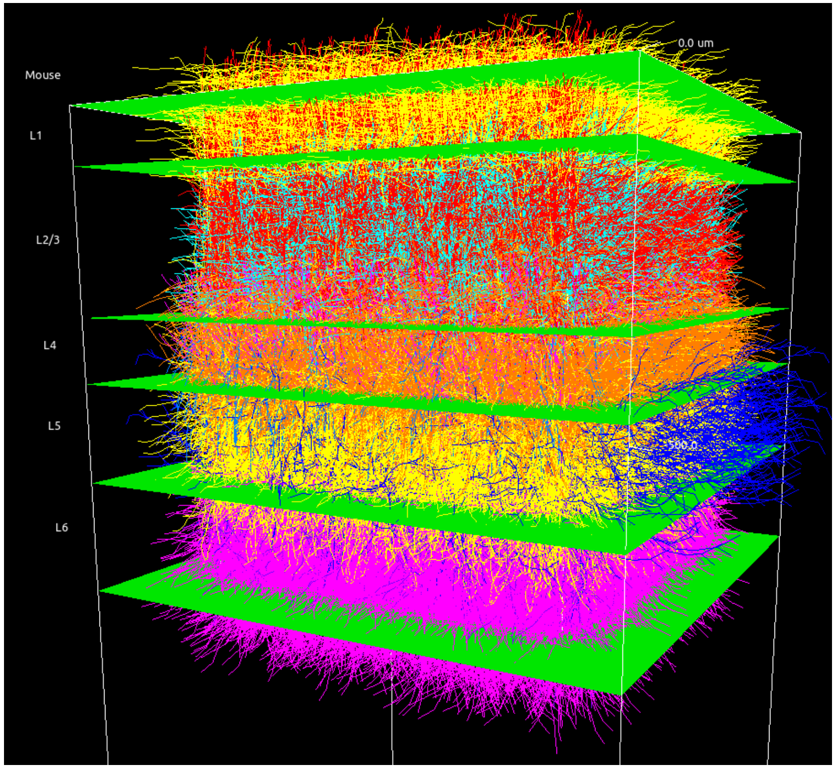
Biophysically detailed neuronal model is characterized by:

- morphology
- biophysical parameters

neuronal models

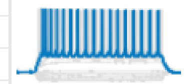


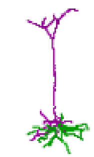
instances of the neuronal models



Representation of neuronal models in Allen Cell Types Database (celltypes.brain-map.org)

Cre Line:	Scnn1a-Tg2-Cre	**Upstroke:Downstroke:**	4.43
Area:	VISp	**Fast AP Trough (mV):**	-49.72
Layer:	4	**FI Curve Slope:**	0.22
Hemisphere:	right	**Rheobase (pA):**	70.00
Dendrite Type:	spiny	**Ramp AP Time (s):**	3.11
Apical Dendrite:	intact	**Resting Vm (mv):**	-79.30



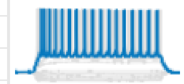


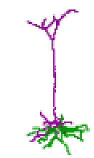
Directory structure:

```
473863035
├── 473863035_fit.json
├── Nr5a1-Cre_Ai14_IVSCC_-169250.03.02.01_471087815_m.swc
├── modfiles
...
└── README
```

Representation of neuronal models in Allen Cell Types Database (celltypes.brain-map.org)

Cre Line:	Scnn1a-Tg2-Cre	**Upstroke:Downstroke:**	4.43
Area:	VISp	**Fast AP Trough (mV):**	-49.72
Layer:	4	**FI Curve Slope:**	0.22
Hemisphere:	right	**Rheobase (pA):**	70.00
Dendrite Type:	spiny	**Ramp AP Time (s):**	3.11
Apical Dendrite:	intact	**Resting Vm (mv):**	-79.30





Directory structure:

```
473863035
├── 473863035_fit.json
├── Nr5a1-Cre_Ai14_IVSCC_-169250.03.02.01_471087815_m.swc
├── modfiles
...
└── README
```

_fit.json includes biophysical parameters:

- “passive” – fitted passive parameters
- “genome” – conductance mechanism and fitted conductance values
- “conditions” – experimental conditions
- “fitting” – fitting parameters

Representation of cells

Organize data using *relational* approach:

- unique key identifying each row
- rows in different tables are linked by keys

Model table
model_id (PK)
model_type
morphology
biophysparams

Cell table
gid (PK)
cell_type (FK)
model_id (FK)
x_soma
y_soma
z_soma
rotation angle(s)
orient_index

Notation:

(PK) - primary key

(FK) - foreign key

Representation of cell information

CSV files (on disk) and Panda's Data Frames (in the code)

Cell table:

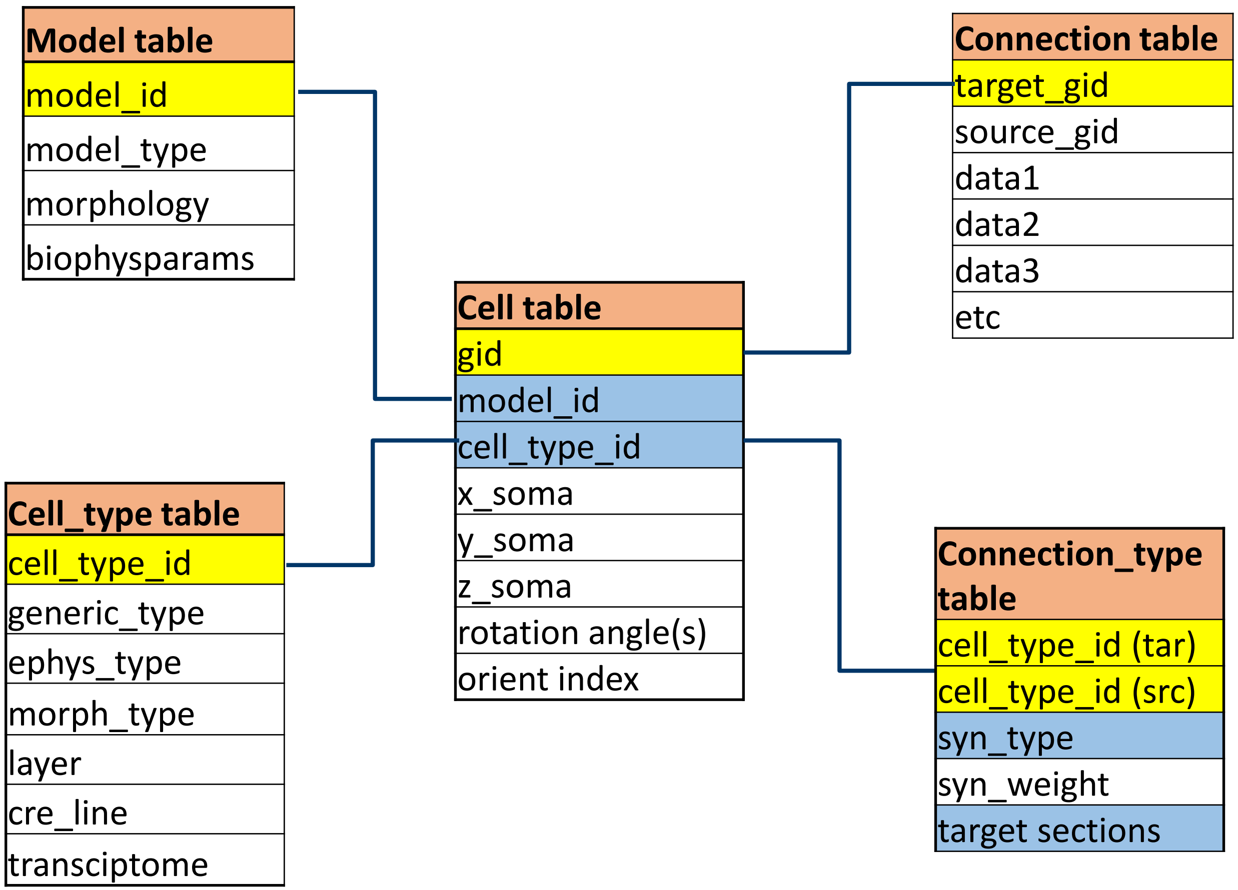
gid	cell_type	model_id	x_soma	y_soma	z_soma	rotation_angle	orient_index
0	e23	473863035	-294.473	193.2292	284.291	1.464351829	6.428571
1	e23	472451419	-63.3053	-109.132	255.0923	3.356761644	12.85714
2	e23	473863035	-137.913	-243.188	130.1602	5.995265603	19.28571
…							
92	e6	LIF_exc1	65.61834	57.73503	709.3448	1.949187072	164.3478
93	e6	471410185	-110.126	28.23057	778.1125	3.399628019	172.1739
94	e6	LIF_exc1	43.37374	45.49852	719.2077	1.929254707	180
95	i6	472301074	92.93924	-228.634	774.0796	2.499155347	30
96	i6	472301074	94.88828	-145.142	780.1953	1.689003763	60

Model table:

model_id	model_type	morphology	biophysparams
471410185	biophysical	excitatory/6a/471410185/Ntsr1-Cre_Ai14_GSL_-180761.02.01.01_475511036_m_rot.swc	excitatory/6a/471410185/471410185_fit.json
472301074	biophysical	inhibitory/Pvalb+/472301074/Pvalb-IRES-Cre_Ai14_IVSCC_-170931.06.01.01_464188986_m.swc	inhibitory/Pvalb+/472301074/472301074_fit.json
…			
473871773	biophysical	excitatory/5/473871773/Scnn1a-Tg3-Cre_Ai14_IVSCC_-177297.05.02.01_470986539_m_rot.swc	excitatory/5/473871773/473871773_fit.json
LIF_exc1	point	null	point/LIF_IntFire1_exc.json
LIF_inh1	point	null	point/LIF_IntFire1_inh.json

Typically Nmodels<<Ncells

Tentative representation of the network model



(PK) - primary key
(FK) - foreign key

Cell type definition example

Used in creating the above tables

```
'i23':{

    'type' : {'cre':'Pvalb-IRES','layer':'23','g':'interneuron','m':'basket','e':'inhibitory'},

    'models': {
        '472306616':  {
            'fraction': 0.5,
            'model_type': 'biophysical',
            'morphology': 'biophysical/inhibitory/Pvalb+/472306616/Pvalb-IRES-Cre_Ai14_IVSCC_-176848.03.01.01_470528201_m.swc',
            "parameters": 'biophysical/inhibitory/Pvalb+/472306616/472306616_fit.json'
        },
        '472912177':  {
            "fraction": 0.5,
            'model_type': 'biophysical',
            "morphology": 'biophysical/inhibitory/Pvalb+/472912177/Pvalb-IRES-Cre_Ai14_IVSCC_-176847.04.02.01_470522102_m.swc',
            "parameters": 'biophysical/inhibitory/Pvalb+/472912177/472912177_fit.json'
        }
    },

    'inputs':  {
        'e23':{'nsyns':1860,'prob':'pyr2int', 'weight':1.0E-5,'syn_type':'e2i','secs':'dendritic100'},
        'i23':{'nsyns':472, 'prob':'int2int', 'weight':2.0E-5,'syn_type':'i2i','secs':'all'},
        'e4' :{'nsyns':255, 'prob':'unilat',  'weight':4.0E-5,'syn_type':'e2i','secs':'dendritic'},
        'i4' :{'nsyns':104, 'prob':'unilat',  'weight':8.0E-5,'syn_type':'i2i','secs':'perisomatic'},
        'e5' :{'nsyns':456, 'prob':'unilat',  'weight':4.0E-6,'syn_type':'e2i','secs':'perisomatic150'},
        'e6' :{'nsyns':46,  'prob':'unilat',  'weight':4.0E-6,'syn_type':'e2i','secs':'distal'},

        'gray_e' :{'nsyns':150, 'prob':'uniform', 'weight':3.4E-4,'syn_type':'e2i','secs':'all'},
        'lgn_e'  :{'nsyns':100, 'prob':'uniform', 'weight':5.0E-5,'syn_type':'e2e','secs':'all'}
    },
},
```

Part of the larger network prescription file used by network builder

Representation of i-j connections

connection table
tar_gid
src_gid
data1
data2
data3
etc

Consider a network size:
N cells~100,000,
N synapses ~2000 per target cell
N sources per target ~400 (with ~5 synapses per connection)

Binary file size for storing different level of details about recurrent connections in sparse coordinate format:

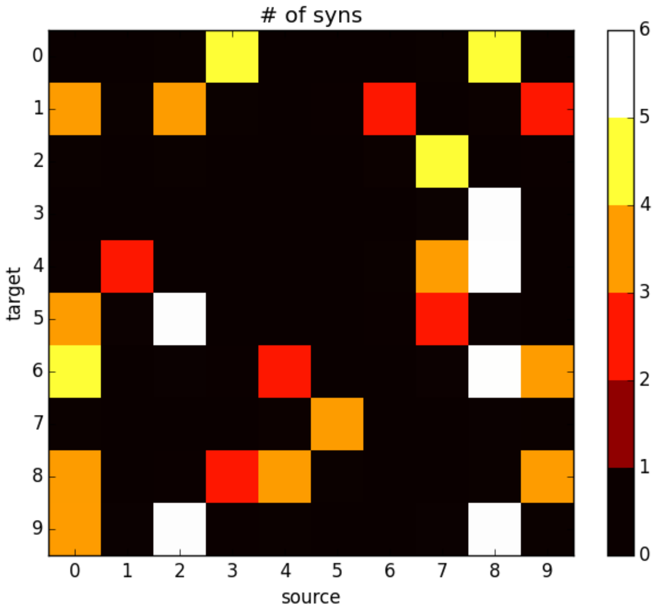
1) tar_gid, src_gid, nsyn: (uint32+uint32+uint8)*4E+2*1E+5 = ~300 MB
2) tar_gid, src_gid, tar_sec_id: (3*uint32)*2E+3*1E+5 = ~2.4 GB
3) add another 0.8 GB for each additional uint32 (or float32)

ij-connections storage

Use:

- hdf5 to be able to load connections for a sub-set of gids
- sparse storage
- 1 dataset per connection property across all gids

10 cell connection example

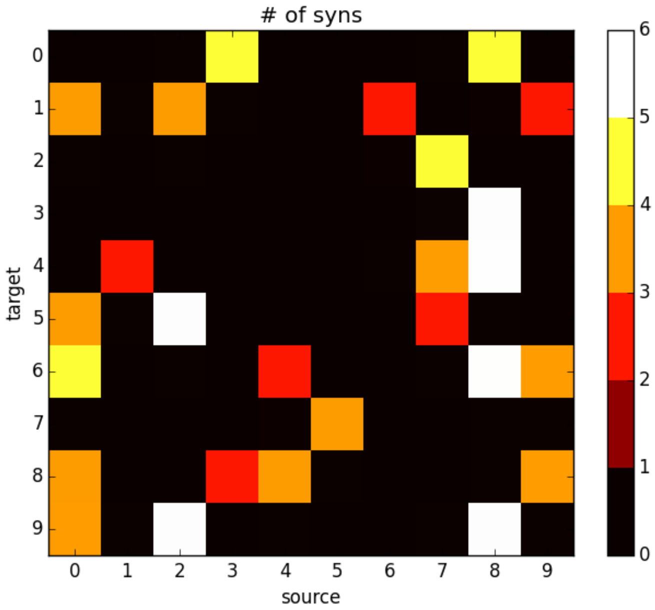


sparse coordinate format:

row	column	data
0	3	4
0	8	4
1	0	3
1	2	3
1	6	2
1	9	2
2	7	4
3	8	5
4	1	2
4	7	3
4	8	5
5	0	3
5	2	5
5	7	2
6	0	4
6	4	2
6	8	6
6	9	3
7	5	3
8	0	3
8	3	2
8	4	3
8	9	3
9	0	3
9	2	5
9	8	5

Problem: Using coordinate format we do not have a way of knowing which lines to load from file for a particular tar_gid.

10 cell connection example



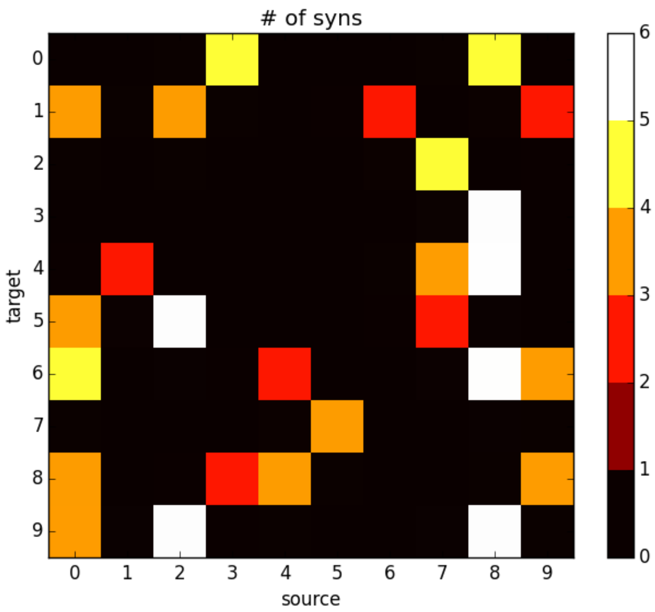
sparse coordinate format:

row	column	data
0	3	4
0	8	4
1	0	3
1	2	3
1	6	2
1	9	2
2	7	4
3	8	5
4	1	2
4	7	3
4	8	5
5	0	3
5	2	5
5	7	2
6	0	4
6	4	2
6	8	6
6	9	3
7	5	3
8	0	3
8	3	2
8	4	3
8	9	3
9	0	3
9	2	5
9	8	5

Problem: Using coordinate format we do not have a way of knowing which lines to load from file for a particular tar_gid.

For instance: connections for tar_gid=5 are in lines [11:14]

Reading slices of connectivity data



Compressed sparse row (CSR) format:

indptr
0
2
6
7
8
11
14
18
19
23
26

row	column	data
0	3	4
0	8	4
1	0	3
1	2	3
1	6	2
1	9	2
2	7	4
3	8	5
4	1	2
4	7	3
4	8	5
5	0	3
5	2	5
5	7	2
6	0	4
6	4	2
6	8	6
6	9	3
7	5	3
8	0	3
8	3	2
8	4	3
8	9	3
9	0	3
9	2	5
9	8	5

For a particular `tar_gid`:

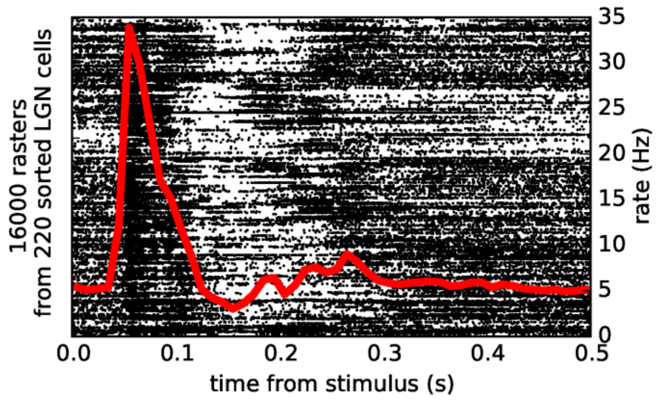
```
src_gids = column[indptr[tar_gid]:indptr[tar_gid+1]]
nsyns = data[indptr[tar_gid]:indptr[tar_gid+1]]
```

External Inputs

Random Poisson input

Parameters specified in the model prescription file

Deterministic input from thalamic recordings



dictionary of numpy arrays:
{lgn_id: spike_train}

Conclusions

- Use relational approach
- Use standard file formats: CSV, JSON, HDF5
- Can handle networks at different level of detail
- Flexibility in specifying the level of detail of the connectivity parameters

Acknowledgments

Catalin Mitelut (Univ. of British Columbia) for column visualization

MAT and Technology teams at the Allen Institute

The End