NSDF : Neuroscience Simulation Data Format

Chaitanya Chintaluri
c.chintaluri@nencki.gov.pl

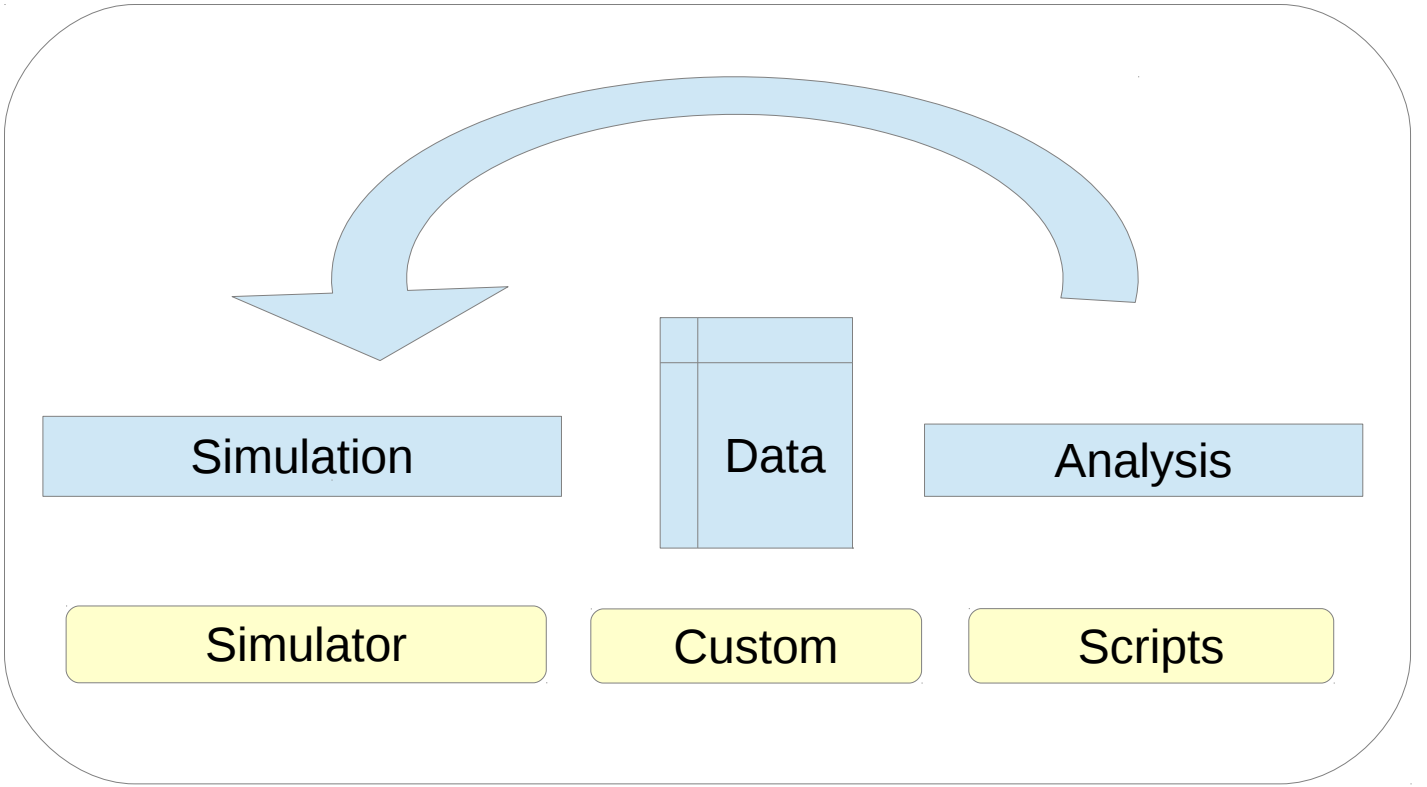
Prof. Daniel K. Wójcik

OSB Workshop, CNS, Prague, 23rd July, 2015

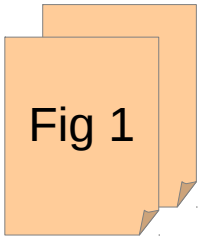
Computational research timeline

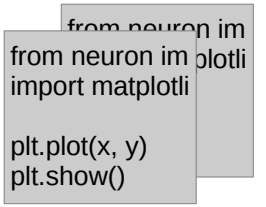
Idea

Programming



Publications





Code

Why save simulation data?

- Save researcher time – better work flow
- Technology transfer
- Software obsolescence
  - Unsupported compilers / hardware / libraries
- Variations in software and hardware
  - Differences between simulators
  - Simulator versions
  - OS / library changes
- Tool development
- Ground truth data / Databases

- Sa
- Te
- So
  -
- Va
  -
  -
  -
- To
- Gr

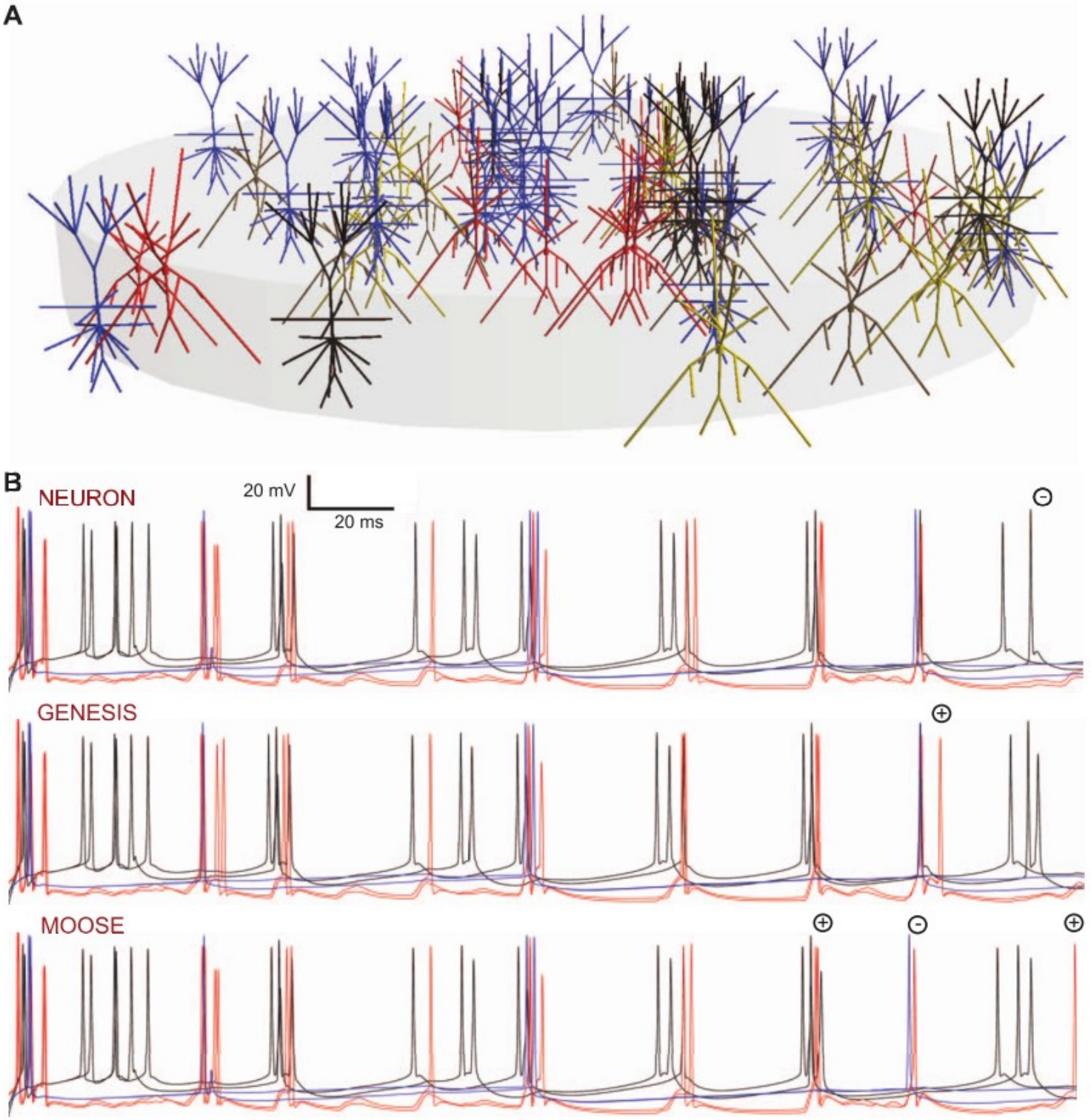
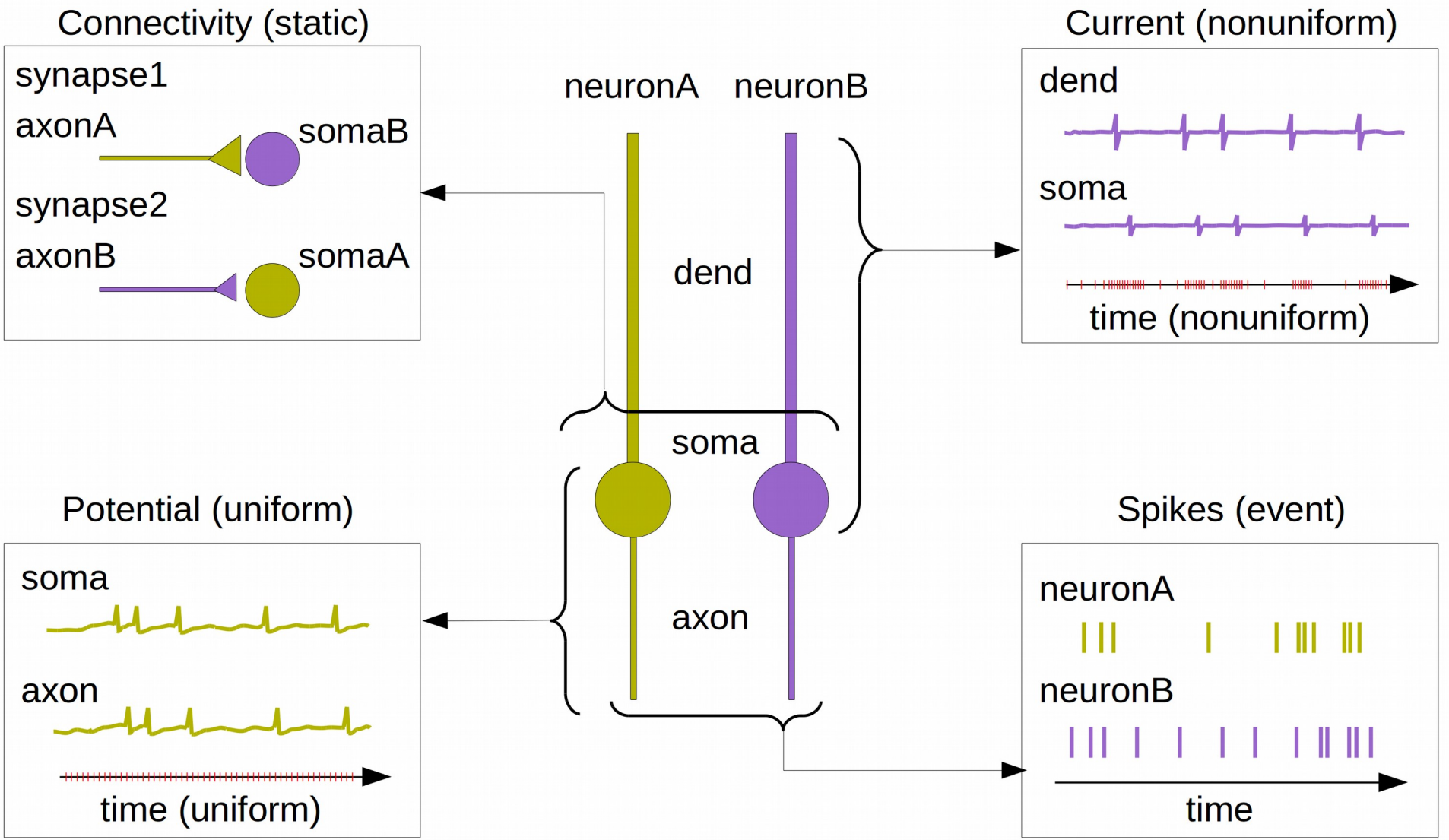


Fig.10, **NeuroML** (2010) PLoS Comp. Bio.

Kinds of simulation data



Our approach : NSDF Neuroscience Simulation Data Format

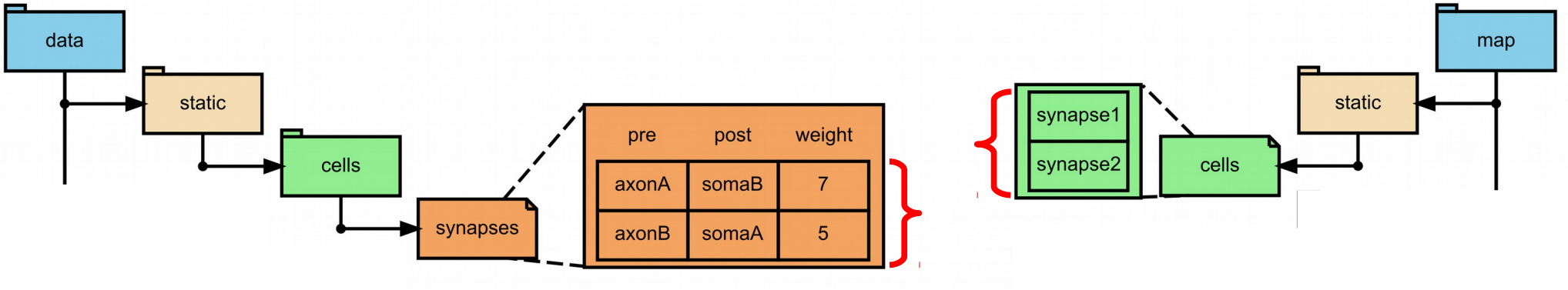
- A HDF5 sub-specification
  - NSDF imposes internal structuring within HDF5
  - No additional libraries necessary
  - All tools supporting HDF5 are extended to NSDF
  - A UNIX folder like organization

Our approach : NSDF Neuroscience Simulation Data Format

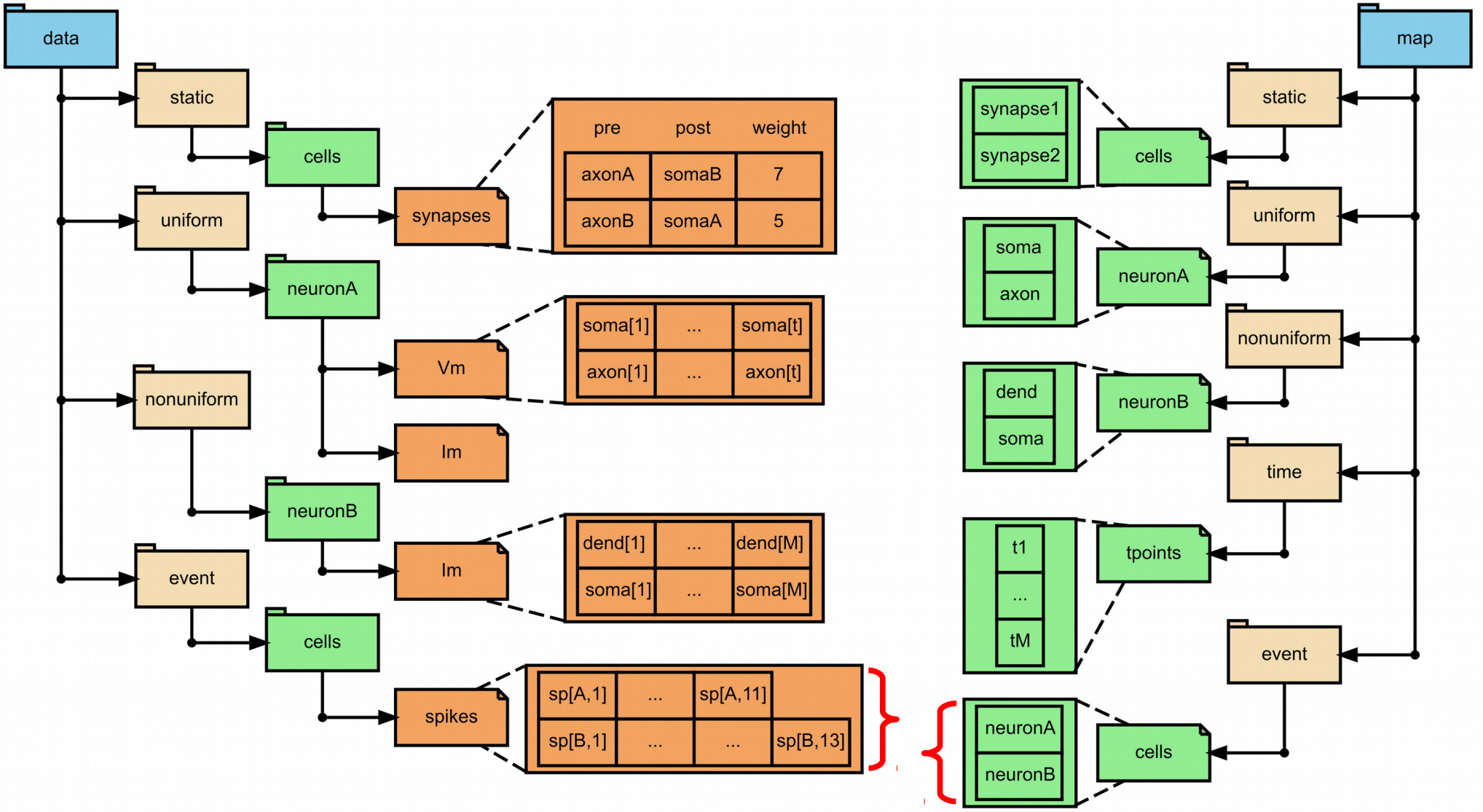
- A HDF5 sub-specification
  - NSDF imposes internal structuring within HDF5
  - No additional libraries necessary
  - All tools supporting HDF5 are extended to NSDF
  - A UNIX folder like organization

- We impose a distinction between
  - Results of the simulation : **/data** entries
  - Labels of the resulting data : **/map** entries
  - Model that generated the data : **/model** entries

NSDF


data
static
cells
synapses
pre
post
weight
axonA
somaB
7
axonB
somaA
5
map
static
synapse1
synapse2
cells

NSDF


data
static
cells
synapses
pre
post
weight
axonA
somaB
7
axonB
somaA
5
uniform
neuronA
Vm
soma[1]
...
soma[t]
axon[1]
...
axon[t]
Im
nonuniform
neuronB
Im
dend[1]
...
dend[M]
soma[1]
...
soma[M]
event
cells
spikes
sp[A,1]
...
sp[A,11]
sp[B,1]
...
...
sp[B,13]
map
static
synapse1
synapse2
cells
uniform
soma
axon
neuronA
nonuniform
dend
soma
neuronB
time
t1
...
tM
tpoints
event
neuronA
neuronB
cells

Options in NSDF

Table 1 Variants in NSDF

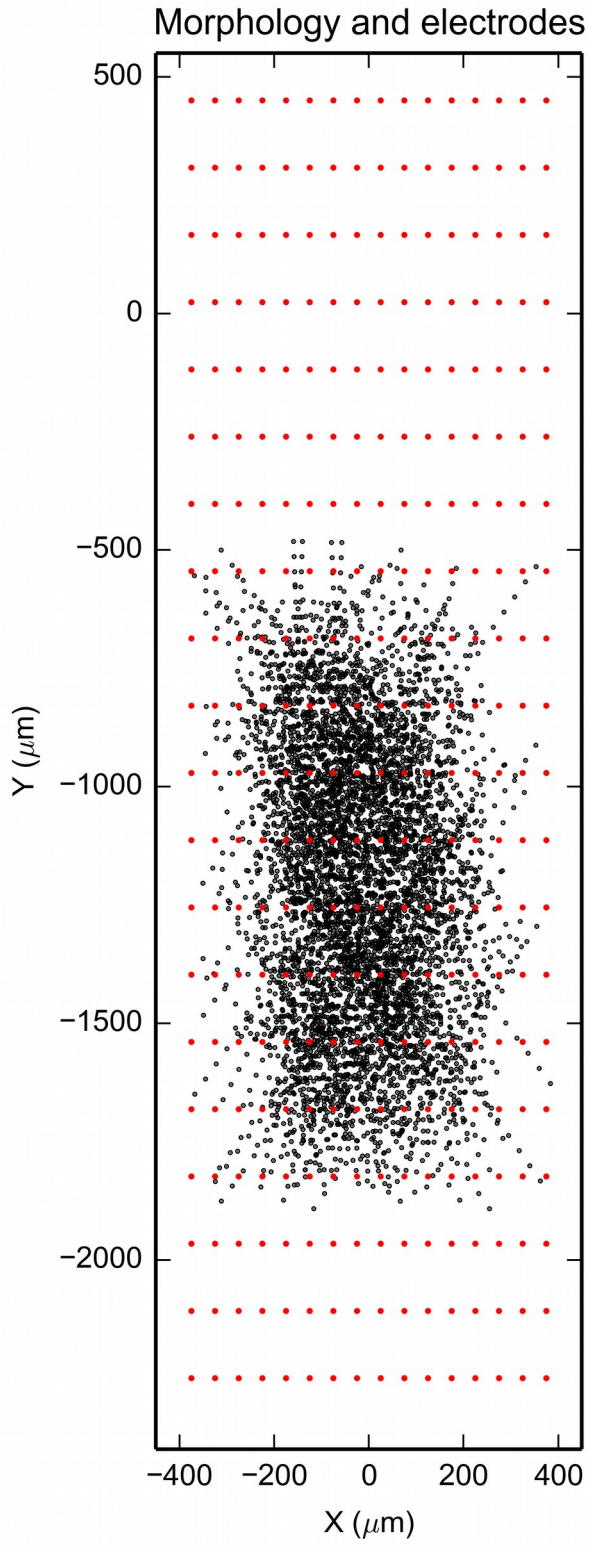
/data	Option	Variant	Shared Time	Level	DS_0	DS_1	Attributes		
static	1	NA	NA	4	source	#	unit	field	
	2	NA	NA	4	source	label	unit	field	
uniform	1	NA	True	4	source	-	unit	field	tstart*
	2	NA	True	4	source	time	unit	field	
nonuniform	1	NUREGULAR	True	4	source	time	unit	field	
	2	ONED	Agnostic	4	-	-	source		
				5	-	time	unit	field	source
	3	VLEN	False	4	source	time	unit	field	
	4	NANPADDED	False	4	source	time	unit	field	
event	1	ONED	NA	4	-	-	source		
				5	-	-	unit	field	source
	2	VLEN	NA	4	source	-	unit	field	
	3	NANPADDED	NA	4	source	-	unit	field	

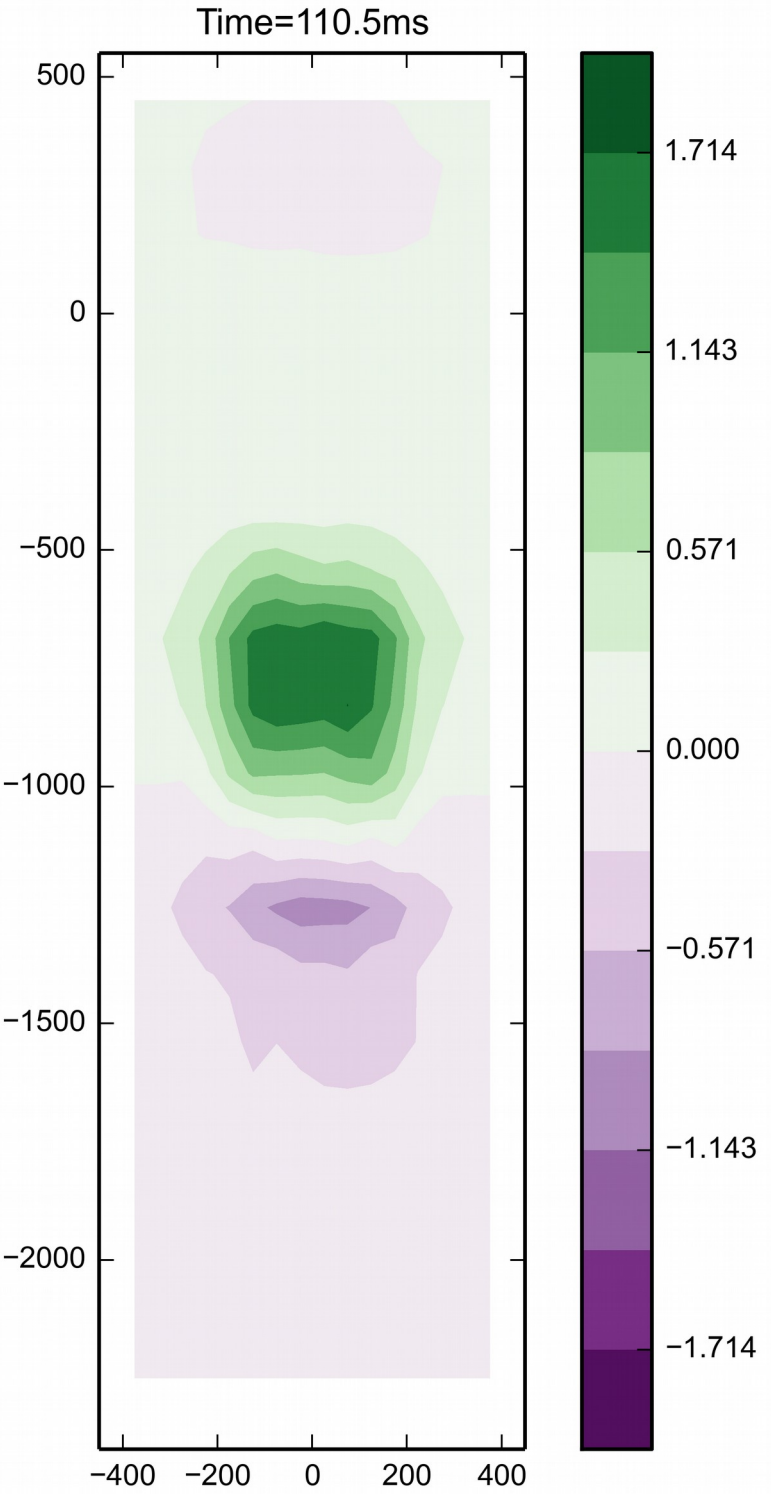
compound arrays

∗ additionally “dt” and “tunit”

```
def place_electrodes_1D(n):
    '''places n number of electrodes in ID along the column'''
    ele_x = np.ones((n,1))*25.
    ele_y = np.linspace(-2100., -80.0, num=n).reshape(n,1)
    ele_z = np.ones((n,1))*25.
    return np.hstack((ele_x, ele_y, ele_z))

def inv_distance(src_pos, ele_pos):
    '''computes the inverse distance between src_pos and ele_pos'''
    dist_matrix = np.zeros((src_pos.shape[0], ele_pos.shape[0]))
    for ii,electrode in enumerate(ele_pos):
        dist_matrix[:, ii] = scipy.spatial.distance.cdist(src_pos, electrode.re\
shape(1,3)).flatten()
    dist_matrix = 1 / dist_matrix #inverse distance matrix
    return dist_matrix

def pot_vs_time(h, pop_name, field_name, src_pos, ele_pos):
    '''returns potentials, at ele_pos, due to src_pos, over time'''
    src_time = h['/data/uniform/'+pop_name+'/'+field_name].value
    ele_src = inv_distance(src_pos, ele_pos).T
    return np.dot(ele_src, src_time)*(1 / (4*np.pi*0.3))

def fetch_mid_pts(h, pop_name):
    '''Fetches mid points of a compartment'''
    all_pts = h['/data/static/morphology/'+pop_name]
    x = (all_pts['x0']+all_pts['x1']) / 2.
    y = (all_pts['y0']+all_pts['y1']) / 2.
    z = (all_pts['z0']+all_pts['z1']) / 2.
    return np.hstack((x, y, z))
```

```
#Populations of interest
pop_names = ['pyrRS23','pyrFRB23','bask23','axax23','LTS23',
             'spinstel4', 'tuftIB5', 'tuftRS5', 'nontuftRS6',
             'bask56', 'axax56', 'LTS56']

#Field Contributed by
field_name = 'i'

#Electrodes
num_ele = 20
ele_pos = place_electrodes_1D(num_ele)
pot_sum = np.zeros((num_ele, time_steps))

#Simulation of interest
h = h5.File('traub_syn.h5', 'r')

for pop_name in pop_names:
    src_pos = fetch_mid_pts(h, pop_name)
    pot_sum += pot_vs_time(h, pop_name, field_name, src_pos, ele_pos)
    print 'Done Computing for pop_name', pop_name
h.close()

#Plotting
plt.imshow(pot_sum[::-1], cmap=plt.cm.PRGn)
plt.show()
```


Morphology and electrodes
500
0
−500
−1000
−1500
−2000
Y (μm)
−400
−200
0
200
400
X (μm)


Time=110.5ms
500
0
−500
−1000
−1500
−2000
−400
−200
0
200
400
1.714
1.143
0.571
0.000
−0.571
−1.143
−1.714

Summary

- Save your simulation data
- NSDF designed specifically for Neuroscience Simulation data.

References

- Python helper library : github.com/nsdf
- Sample datasets available : bit.ly/nsdf

Acknowledgments

National Centre for Biological Sciences, Bangalore, India.

Prof. Upinder S. Bhalla

Dr. Subhasis Ray

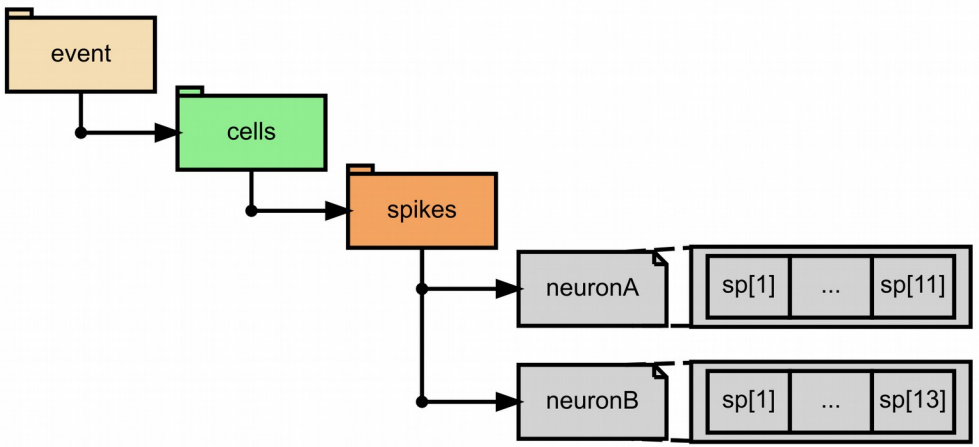
Nencki Institute of Experimental Biology, Warsaw, Poland.

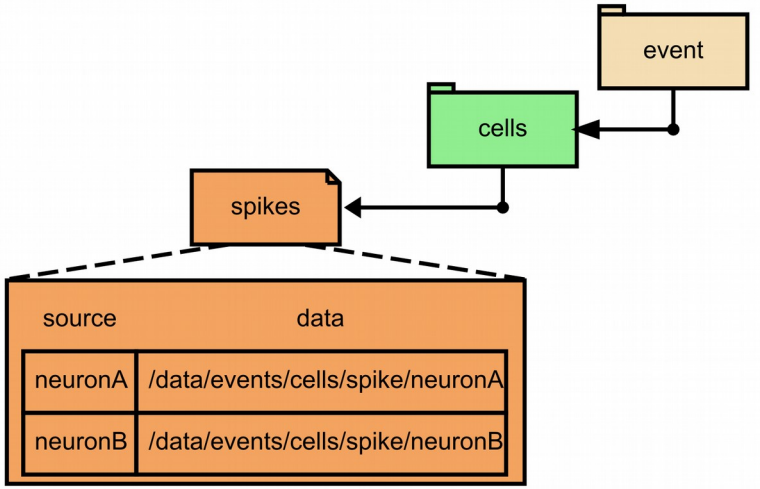
Prof. Daniel Wójcik

Matteo Cantarelli and
Johannes Rieke for their critical comments.

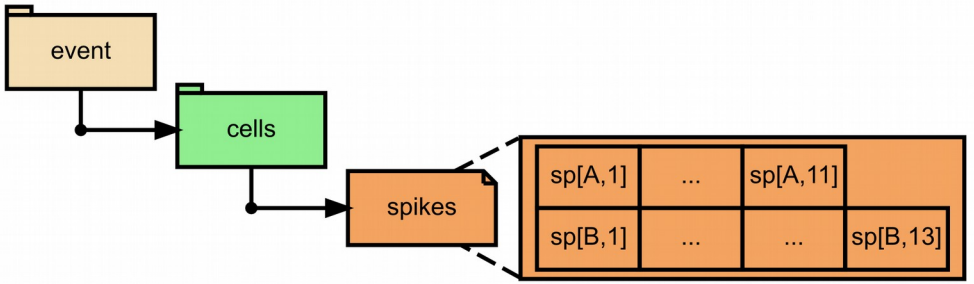
Event variants

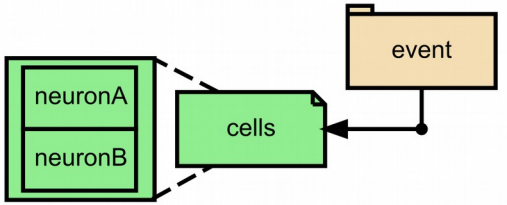
A



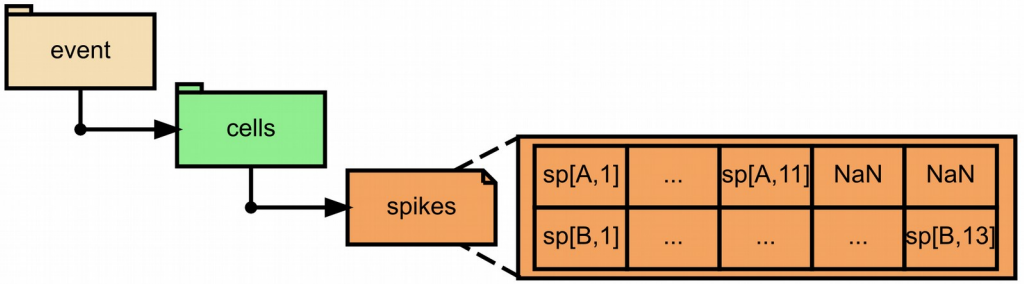


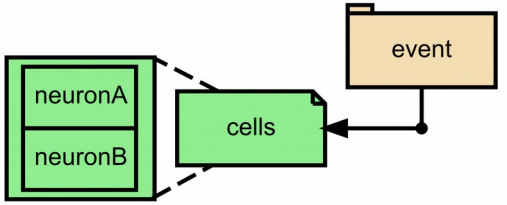
B

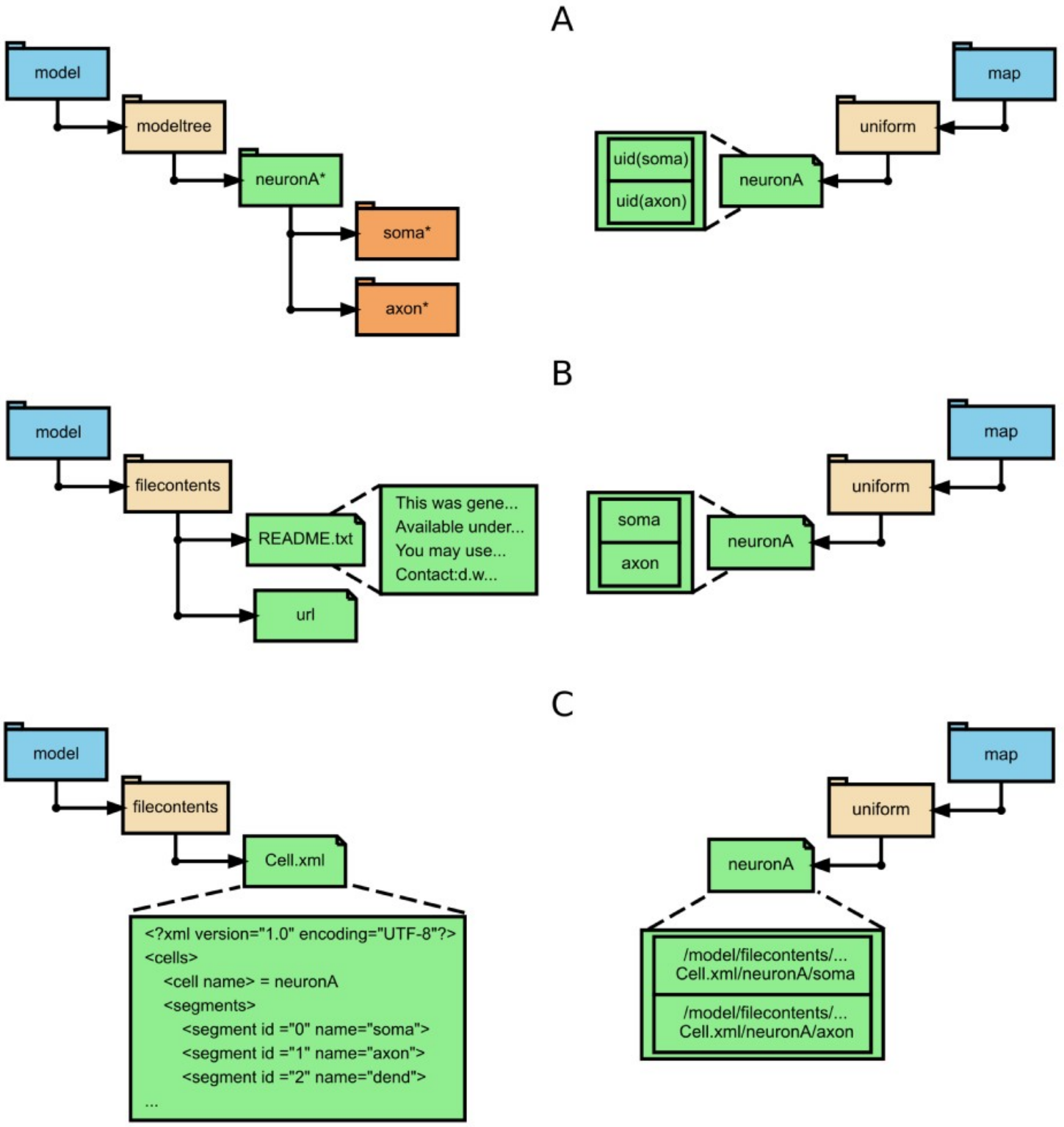




C






A
model
modeltree
neuronA*
soma*
axon*
map
uniform
uid(soma)
uid(axon)
neuronA
B
model
filecontents
README.txt
url
This was gene...
Available under...
You may use...
Contact:d.w...
map
uniform
soma
axon
neuronA
C
model
filecontents
Cell.xml
<?xml version="1.0" encoding="UTF-8"?>
<cells>
<cell name> = neuronA
<segments>

...
map
uniform
neuronA
/model/filecontents/...
Cell.xml/neuronA/soma
/model/filecontents/...
Cell.xml/neuronA/axon

Cheers!