

NIX

An HDF5-based data file format

Achilleas Koutsou

December 18, 2017



- ▶ Development and free distribution of tools for handling and neurophysiological data.
- ▶ All tools developed within the G-Node are open source and freely available.
- ▶ Main projects:
 - ▶ **NIX**: Manage data and metadata together in an open, versatile format.
 - ▶ odML: Collect and manage all information about your experiment.
 - ▶ GIN: Secure data storage, easy collaboration and publication.

Data, results, and metadata

Data:

- ▶ Voltage traces, EEG recordings
- ▶ Subject location across time
- ▶ Subject preference to selection task

Analysis results:

- ▶ Spike times
- ▶ Total time the subject spent in an area
- ▶ Frequency of correct response over time

Data, results, and metadata

Data:

- ▶ Voltage traces, EEG recordings
- ▶ Subject location across time
- ▶ Subject preference to selection task

Analysis results:

- ▶ Spike times
- ▶ Total time the subject spent in an area
- ▶ Frequency of correct response over time

Metadata:

- ▶ Recording equipment
- ▶ Subject age, gender, physical attributes
- ▶ Date, time, experimenter

Problems

The problem with (not necessarily) old datasets:

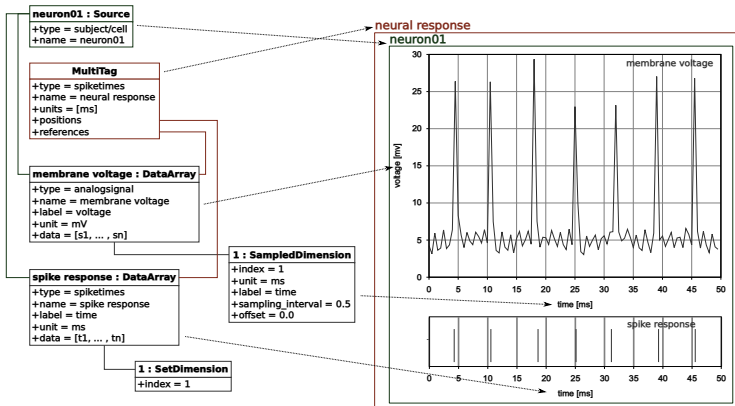
- ▶ Where did this dataset come from?
- ▶ When was it created?
- ▶ What were the simulation parameters when I generated these numbers?
- ▶ Who ran the experiment?
- ▶ What's the meaning of the second row of numbers?

Main features

- ▶ Open data format
- ▶ Store data, analysis results, and metadata conveniently in the same file
- ▶ Descriptive associations between data, analysis results and metadata

NIX

Object hierarchy schema



Libraries available for multiple languages

C++ core library and reference implementation.

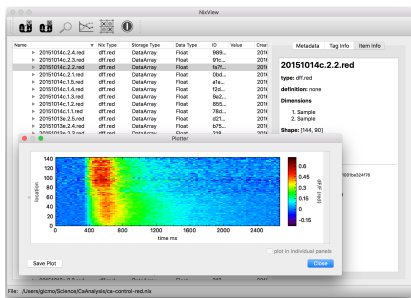
Python bindings for core lib as well as complete reimplementations.

Matlab bindings for core lib.

Java bindings for core lib.

Neo IO allows interoperability with Neo, an API for organising electrophysiological data.

NixView — Cross-platform GUI viewer



- Convenient exploration of data and metadata.
- Exports data to CSV.
- Plotting of data.

NIX

Python code

```
import nixio as nix
import numpy as np

data = np.load("bunch-o-numbers.npz")["data"]

# create new nix file (overwrite truncates existing file)
nixfile = nix.File.open("mydata.nix", nix.FileMode.Overwrite)
# block: top level data grouping
block = nixfile.create_block("DataBlock", "experiment")
# data array: data storage object
darray = block.create_data_array("raw-01", "raw data", data=data)
time = darray.append_sampled_dimension(0.1)
time.unit = "ms"
darray.append_set_dimension()
```

NIX

Python code

```
# identify regions of interest with start position and length
regionsstart = np.array([13, 45, 63])
regionslength = np.array([2, 2, 2])

roipos = block.create_data_array("roi-positions", "positions",
                                data=regionsstart)
roiext = block.create_data_array("roi-extents", "extents",
                                data=regionslength)
mtag = block.create_multi_tag("roi", "regions of interest", roipos)
mtag.extents = roiext
mtag.units = "ms"
mtag.references.append(darray)

nixfile.close()
```

NIX

HDF5 structure

```
/ Group
/data Group
/data/DataBlock Group
/data/DataBlock/data_arrays Group
/data/DataBlock/data_arrays/raw-01 Group
/data/DataBlock/data_arrays/raw-01/data Dataset {1000/Inf, 2/Inf}
/data/DataBlock/data_arrays/raw-01/dimensions Group
/data/DataBlock/data_arrays/raw-01/dimensions/1 Group
/data/DataBlock/data_arrays/raw-01/dimensions/2 Group
/data/DataBlock/data_arrays/roi-extents Group
/data/DataBlock/data_arrays/roi-extents/data Dataset {3/Inf}
/data/DataBlock/data_arrays/roi-positions Group
/data/DataBlock/data_arrays/roi-positions/data Dataset {3/Inf}
/data/DataBlock/multi_tags Group
/data/DataBlock/multi_tags/roi Group
/data/DataBlock/multi_tags/roi/extents Group, same as /data/DataBlock/data_arrays/raw-01/dimensions/1
/data/DataBlock/multi_tags/roi/positions Group, same as /data/DataBlock/data_arrays/raw-01/dimensions/2
/data/DataBlock/multi_tags/roi/references Group
/data/DataBlock/multi_tags/roi/references/b55fc44a-059b-4064-b864-8359150c097a Dataset {2/Inf}
/data/DataBlock/multi_tags/roi/units Dataset {2/Inf}
/metadata Group
```

NIX

Data in NixView

The screenshot shows the NixView application window. The main panel displays a hierarchical tree of data. The 'rol' item is selected, and its details are shown in the right-hand panel.

Name	Nix Type	Storage Type	Data Type	ID	Value	Created at
Metadata			n.a.			
Data			n.a.			
DataBlock	experiment	Block	n.a.	0d4e03f0-c247-48bd-a050-05c33d940162		201712181
rol-positions	positions	DataRow	Int64	ee32c400-0490-4ffd-904e-31eacc90c156		201712181
rol-extends	extends	DataRow	Int64	4fb10981-139d-4ea6-9594-5f2848aa2c6f		201712181
rol	regions of interest	MultiTag	n.a.	b1302418-0179-474e-9dc1-fa49c606e589		201712181
raw-01	raw data	DataRow	Double	b55fc44a-059b-4064-b864-8359150c097a		201712181

rol

type: regions of interest

definition: none

Metadata: none

References

- raw-01[raw data]

Features

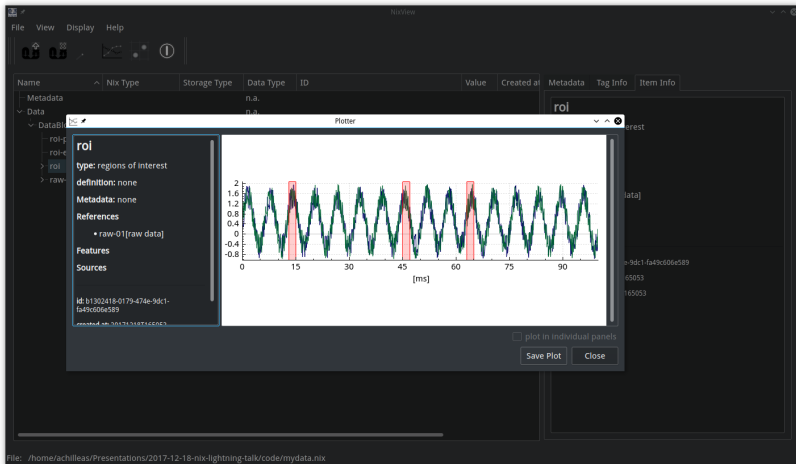
Sources

id: b1302418-0179-474e-9dc1-fa49c606e589

created at: 20171218165053

updated at: 20171218165053

File: /home/achilleas/Presentations/2017-12-18-nix-lightning-talk/code/mydata.nix



Resources

- ▶ NIX info and documentation:
`https://github.com/G-Node/nix/wiki`
- ▶ NIX source: `https://github.com/G-Node/nix/`
- ▶ NIX Python source: `https://github.com/G-Node/nixpy/`