NIX - Comprehensive Storage of Neuroscience Data and Metadata

Christian Kellner
LMU Munich, G-Node
kellner@bio.lmu.de

NWB Hackathon

G-Node

Origins

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

- Generalization of NEO (http://www.neuralensemble.org/neo)
- Started in 2012 at the INCF Hackathon

- Main developers:
  - Adrian Stoewer, Programmer, G-Node
  - Jan Grewe, Neuroscientist, Uni Tübingen
  - Christian Kellner, Neuroscientist & Programmer, LMU Munich
  - Balint Morvai & Andrey Sobolev, Programmers, G-Node

G-Node

Overview

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

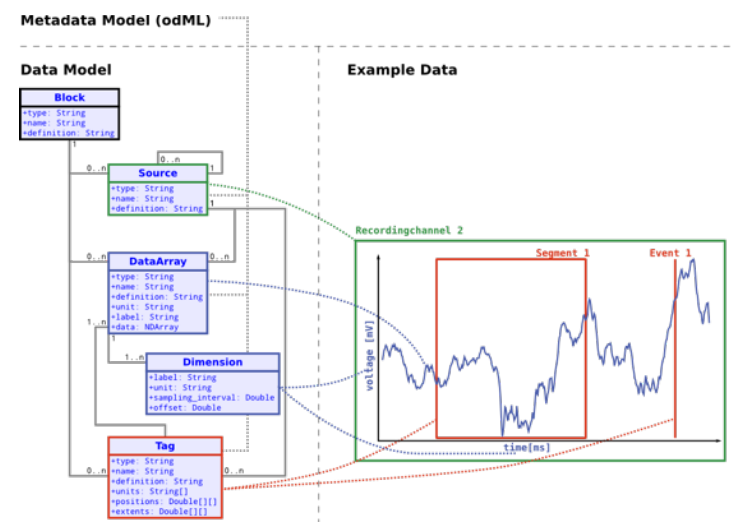
- Data Model
- HDF5 Data Format
- C++ Core Library / API
- Language Bindings
- High-Level API
- Tools

G-Node

Data Model

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

- Flexible model to store **raw** data, **derived** data, **relations** of the data (e.g. waveforms of spikes, ROI)
- Core component: **DataArray** with dimensions, units (SI)
- Full **metadata** integration (odML [1], RDF/ontologies)
- Store all necessary information to create a basic scientifically correct plot, including labels & units
- fulfils INCF Ephys Data Sharing Task Force requirements for storing electrophysiology data



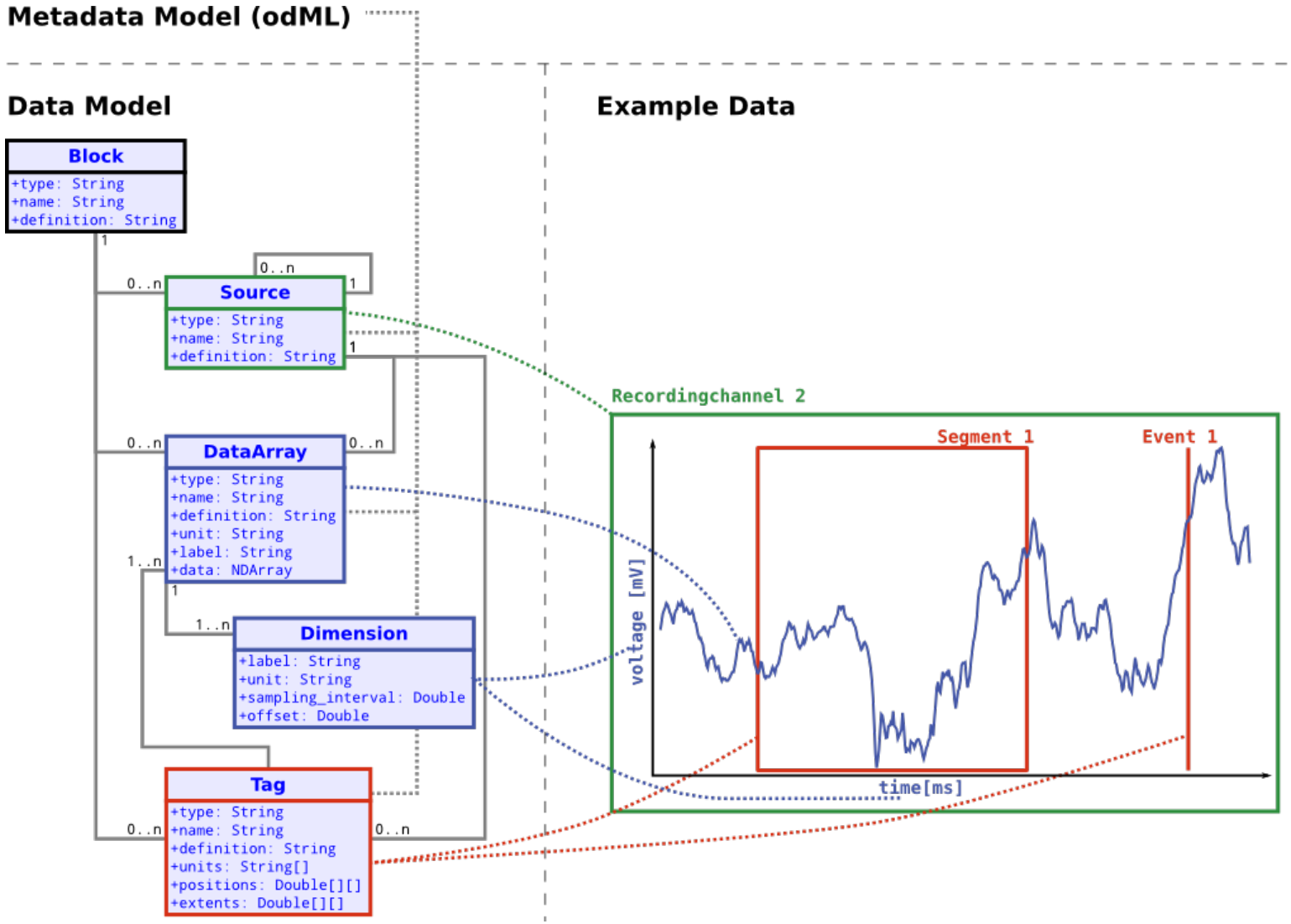
https://github.com/G-Node/nix/wiki/The-Model

[1] Grewe et al (2011) Frontiers in Neuroinformatics 5:16

G-Node

Data Model

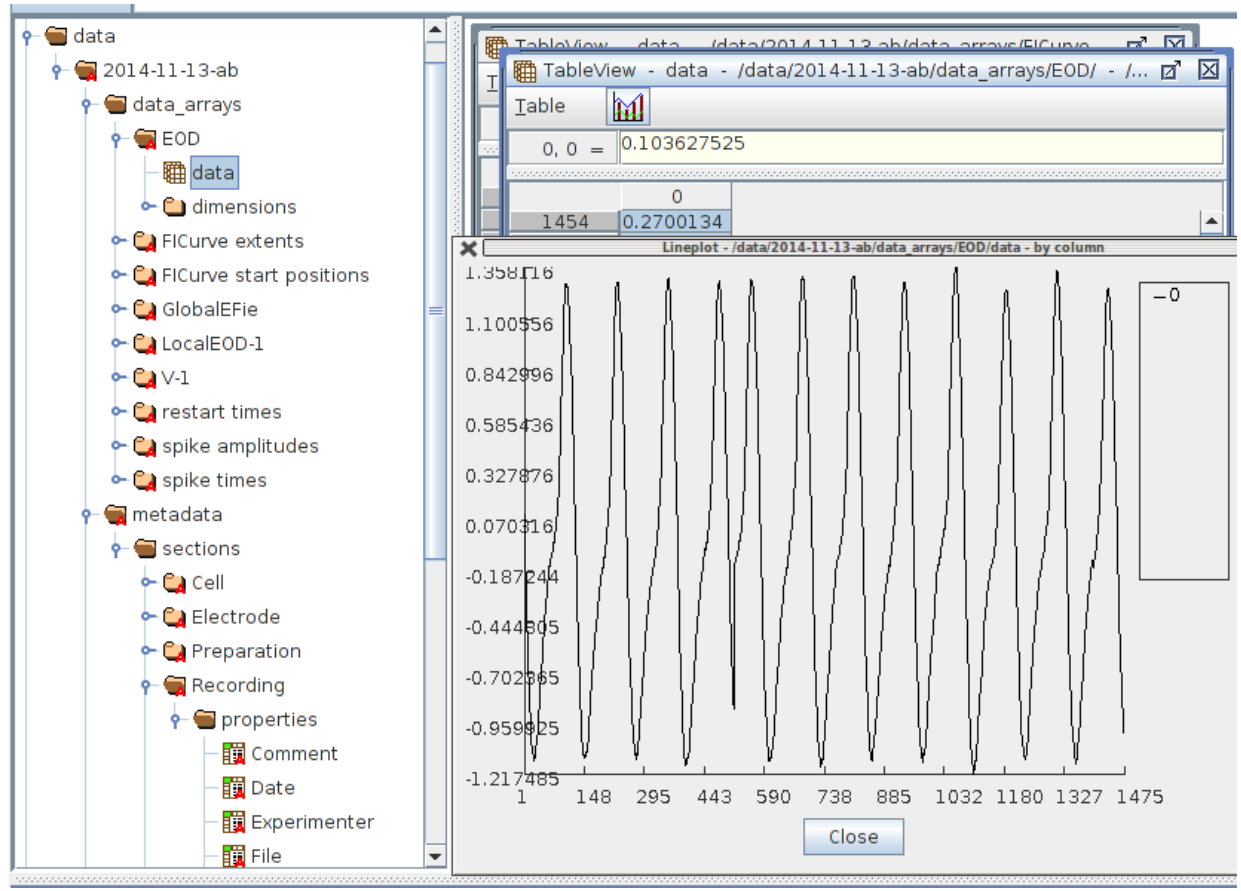
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



https://github.com/G-Node/nix/wiki/The-Model

File Format

- HDF5
- Structure reflects data model
- Easy to understand
  - human readable names
  - hdf5 links
  - mtime, ctime
  - version information

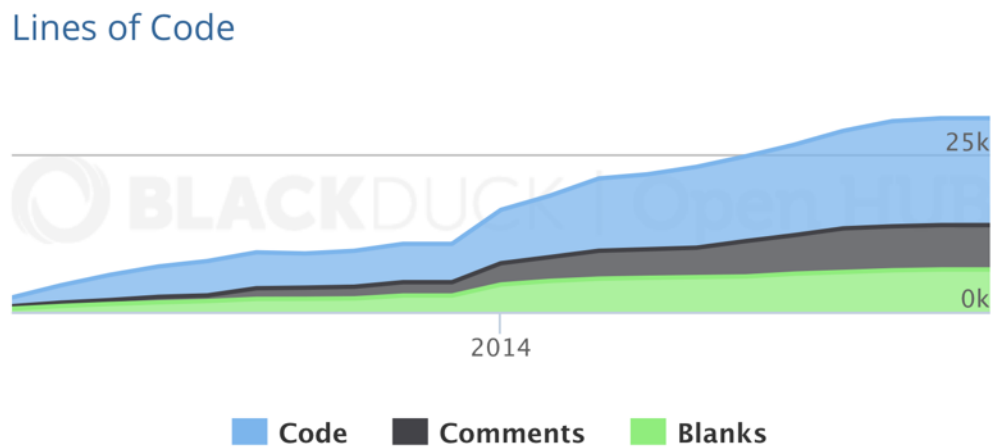


HDFView: can view and plot data in NIX file

G-Node

NIX Library / API

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



- Core component
- Convenient IO API, utility functions
- Multiple backends
- C++ (modern C++11)
- 1,854 commits
- 17,069 lines of actual code
- Well documented
- Unit tested + Continous integration
- Linux, MacOS X & Windows support
- Debian packages in a PPA, Windows binaries
- Command line tool (browse file + verification), benchmark tool
- BSD 4-clause license

https://github.com/G-Node/nix

G-Node

Benchmark tool

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

```
% ./nix-bench
Performing generators tests...
Performing disk IO tests...
Performing read tests...
Performing write tests...
Performing read tests...
Performing read (poly) tests...
 === Reports ===
Double@{ 2048 1 }, G, 85.087 MB/s, 1.1152e+07 N/s
Double@{ 1 2048 }, G, 84.584 MB/s, 1.1087e+07 N/s
Double@{ 2048 1 }, O, 70.219 MB/s, 9.2037e+06 N/s
Double@{ 1 2048 }, O, 69.87 MB/s,  9.158e+06  N/s
Double@{ 2048 1 }, I, 4687.5 MB/s, 6.144e+08  N/s
Double@{ 1 2048 }, I, 5408.7 MB/s, 7.0892e+08 N/s
Double@{ 2048 1 }, W, 29.039 MB/s, 3.8062e+06 N/s
Double@{ 1 2048 }, W, 41.74 MB/s,  5.471e+06  N/s
Double@{ 2048 1 }, R, 73.545 MB/s, 9.6396e+06 N/s
Double@{ 1 2048 }, R, 137.09 MB/s, 1.7969e+07 N/s
Double@{ 2048 1 }, P, 56.155 MB/s, 7.3604e+06 N/s
Double@{ 1 2048 }, P, 89.255 MB/s, 1.1699e+07 N/s

[…]
```

G-Node

Python Bindings

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

- Boost.Python based
- Pythonic, IO interfaces modelled after h5py
- IO directly to NumPy arrays
- Unit tested + Continous integration
- Linux, MacOS X & Windows support
- Helper for plotting
- Large collection of tutorials
- BSD 4-clause license

https://github.com/G-Node/nixpy

http://g-node.github.io/nixpy/tutorial.html

G-Node

Python Bindings

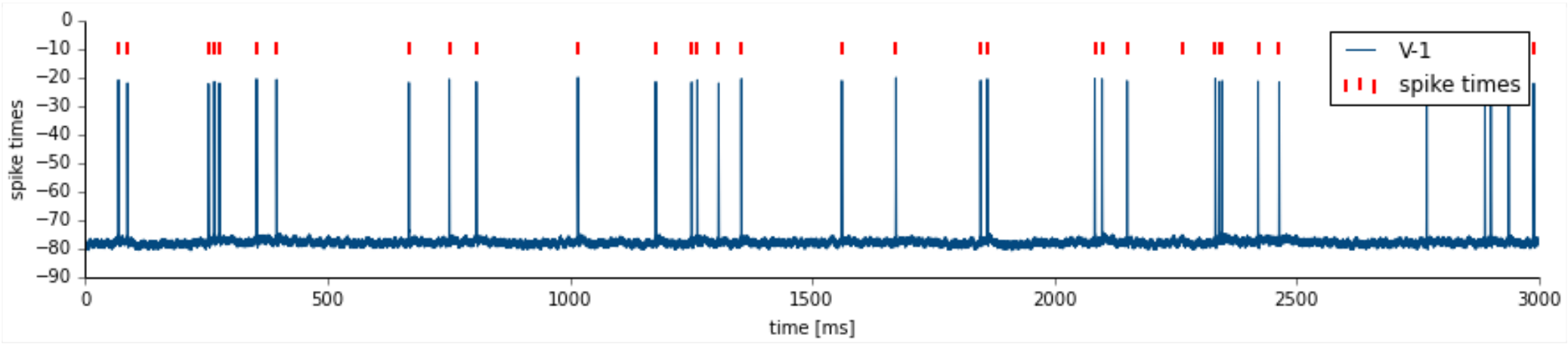
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

```
import nix
from utils.plotting import Plotter

# open file
nix_file = nix.File.open('data.nix.h5')

# open data arrays of first block
b = nix_file.blocks[0]
voltage = b.data_arrays['V-1']
spikes = b.data_arrays['spikes']

plotter = Plotter(width=1280, height=800, lines=4)
plotter.add(voltage, 0)
plotter.add(spikes, color='red')
plotter.plot()
```

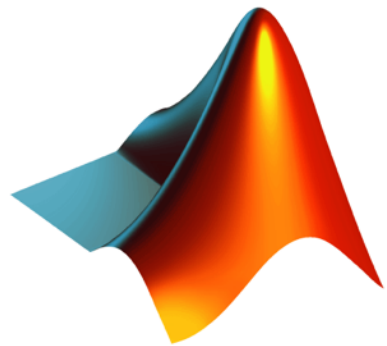


G-Node

Matlab Bindings

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

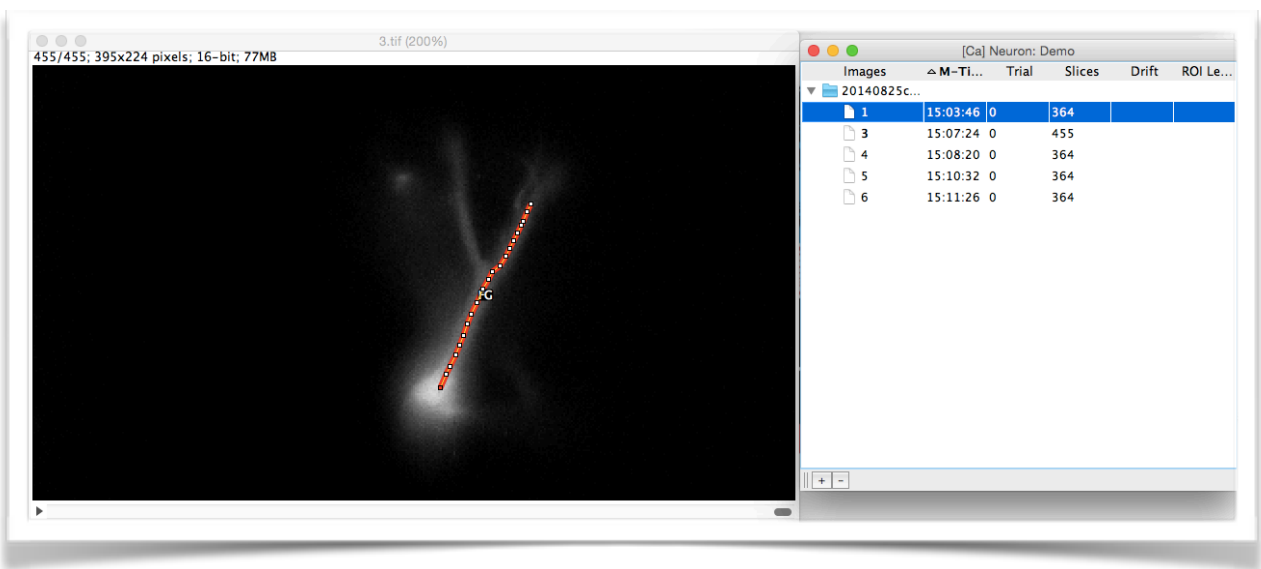
- Using Matlab's C interface
- IO directly to Matlab arrays
- Early development stage
- High priority
- BSD 4-clause license



https://github.com/G-Node/nix-mx

Java Bindings

- SWIG based
- Proof-of-Concept stage
- High priority
- ImageJ plugin
- BSD 4-clause license



https://github.com/G-Node/nix-java

G-Node

High-Level API

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

- User facing API
- Implementation at C++ level with bindings
- High-Level API for

  Electrophysiology (work in progress, Jan Grewe),

  Simulation (NEST),

  Images / Image Stacks + Video Data

G-Node

Usage

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

- RELACS (http://relacs.sourceforge.net), fully integrated for data & metadata
- Benda lab, Uni Tübingen: Ephys (field-potentials, spike data), stimulation, behaviour (video tracking)
- Felmy lab, LMU Munich: Ephys (patch clamp) + Ca-imaging
- Wachtler lab, LMU Munich: Ephys (ERG, HD-MEA), simulation; eye-tracking (in development)
- Leibold lab, LMU Munich: Ephys (tetrode), behavior
- Grün Lab, Research Center Jülich (evaluation)
- NEO (http://neuralensemble.org/neo, planned)

G-Node

Outlook

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

- RDF/ontologies integration for metadata
- Provenance
- Julia bindings

G-Node

Demos

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

IPython Notebooks:

- Basic usage, 'automatic' plotting
- RELACS data: ephys + behaviour + metadata
- NEST Simulations
- NWB datasets pvc-6, pvc-7