Considerations for developing a standard for storing electrophysiology data in HDF5

Jeff Teeters, Jan Benda, Andrew P. Davison, Stephen Eglen, Stephan Gerhard, Richard C. Gerkin, Jan Grewe, Kenneth Harris, Tom Jackson, Roman Mouček, Robert Pröpper, Hyrum L. Sessions, Leslie S. Smith, Andrey Sobolev, Friedrich Sommer, Adrian Stoewer, Thomas Wachtler

incf Program on Standards for Data Sharing

Motivation:

HDF5 is increasingly being used for storing neuroscience electrophysiology data. A standard how data is stored could greatly facilitate data sharing.

Requirements

A standard must specify methods for storing data and metadata.

Data types

Basic data types included in Neuroshare:

- *Time Series* - continuous recording.
- *Time Series Segment* - short sections of recorded data, usually encompassing spike waveforms.
- *Neural Event* - times of neural events, e.g. spikes, optionally associated with sorted units (neurons).
- *Experimental Event* - list of times and values; used to describe stimuli or other experimental events.

Other data types:

- *Images, image stacks* - for imaging data
- *Feature vectors* - for spike sorting

Metadata

There are many types of metadata. Examples:

- Experimental context, i.e.: what species, lab, brain region, etc.
- Data type (spikes, waveforms, analog) and units.
- Annotations, such as cell type, firing rate.
- Relationships between the data. (e.g. two recordings are from the same electrode).
- Stimulus information, and other experimental events.
- Workflow specifying how data was derived.

Some issues

Object model & API vs. File Format Standard.

An object model with an API's has the advantage of being independent of any file format and allowing usage of different back end stores. However, it requires that an API library be written for every platform.

A File Format Standard uses the specification of the format as the standard, so no additional API is needed. However, multiple storage methods or formats are not possible, and users must write code to use the HDF5 API. Which is more appropriate? Can both approaches be used at the same time?

Simplicity of data structures

If the data structures used to store numeric data were easily understandable to humans (via HDFView) the standard would be easier to use. But, would this preclude designs that have other advantages, such as easier machine processing?

Flexibility for data organization

Should the standard allow using custom HDF5 groups, and custom names for nodes to organize the data? Such flexibility would allow meaningful organization of data by users. Can this be done in a way that allows software tools to easily locate the data and metadata in the files?

Level of metadata completeness

Minimal metadata allows identifying the data types and units. Additional metadata (or conventions) are needed to specify relations between different data, context of the experiment, annotations, and workflows. For data sets with multiple files, a machine-readable inventory describing all files in the data set would be useful.

Which of these metadata should be part of a standard? Is it possible to start with minimal metadata, then later and add other metadata on top of that?

Metadata format

Should a standard specify a common format for all metadata? Would such a common format make it difficult to store some metadata, for example metadata best stored in a relational database? Could using a relational database (SQLite) be part of the standard?

Interoperability and data integration

How can a standard be designed to facilitate integration of data from different data sets? This requires that both data and metadata be machine processible. How should standard ontologies be incorporated into a standard?

Conclusion: a standard must:

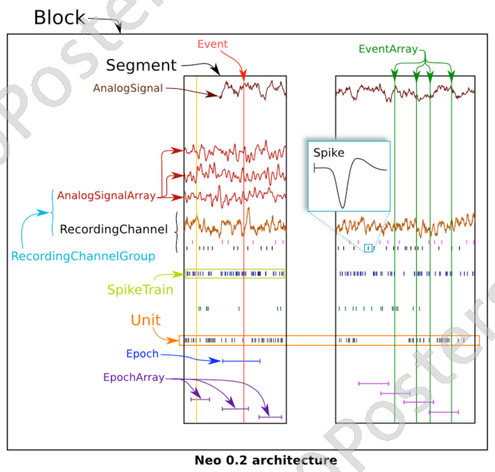
1. Allow storing basic data types.
2. Be flexible to allow incorporate many types of metadata; including metadata that can't be anticipated.
3. Also allow using standard metadata schema and ontologies to make data machine processable.

Existing HDF5 formats

NEO - H5IO

http://packages.python.org/neo/

NEO provides a data model and API (in Python) for electrophysiology data which allows interfacing with many different file formats, including HDF5.



Neo 0.2 architecture

Using NEO, time series segments are stored in HDF5 as two arrays for each channel in path like:
/Block_0/segments/Segment_0/spiketrains/SpikeTrain0/{times, waveforms}
times - 1D array (data set) of spike times.
waveforms - 2D array of spike wave forms.

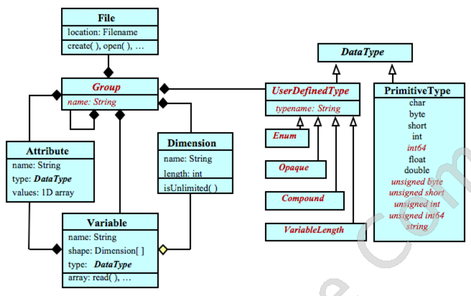
klusta-team Kwik-format

github.com/klusta-team/kwiklib/wiki/Kwik-format
Intended for particular software used to process multi-electrode extracellular recordings, with each electrode (shank) having multiple recording sites.
Uses multiple HDF5 files, named KWIK and KWD.
KWIK - Single HDF5 file; stores spike sorting data for each shank in a single HDF5 table with path: /shanks/shankX/*spikes*. X is the shank number. *spikes* is a 2D array, each row has: spike time, features, mask, cluster number (after spike sorting).
{.raw,.low.,hi}.KWD - (three HDF5 files). Contain raw, low and high pass filter time series data. Format is a single 2D array for each.
Uses JSON to store metadata. Has predetermined metadata structures for electrode properties.

netCDF4

http://www.unidata.ucar.edu/software/netcdf

* Data model and API for storing and sharing array-oriented scientific data. Uses HDF5.
* The content of a netCDF file can be described using an XML document called a "Data Convention", which is based on NcML schema.
* Extensive community support, many software tools available. Used by PLATO project (RIKEN Brain Science Institute) to store experimental data of large-scale modeling. (http://dx.doi.org/10.1016/j.neunet.2011.06.011).



netCDF4 data model.

NeXus File Format.

http://www.nexusformat.org/

* Designed for storing particle physics data. But could be used for other domains.
* Either HDF5 or XML can be used to store data.
* Uses a high level language, NeXus Definition Language (NXDL), to define structures which can be stored.
* The definitions for different structures are stored in a central repository.
* Tools exist for validating data files using the stored definitions.
* A community process is in place for approving which new structures can be contributed, allowing expansion of the data types stored.

BrainLiner

brainliner.jp is a web portal for sharing brain and behavioral data for neuroscience. It uses HDF5 format for some data and has online visualization.

Ovation

http://physionconsulting.com/web/Ovation.html
Ovation is a commercial system for managing laboratory scientific data. It includes an open source data model which could be useful for developing a standard. Data can be exported to HDF5.

NeuroHDF

http://neurohdf.readthedocs.org/en/latest/
Has conventions for storing neuroscience data in HDF5. Specifications for electrophysiology data are not yet in place, but could be added.

Proposed formats

epHDF

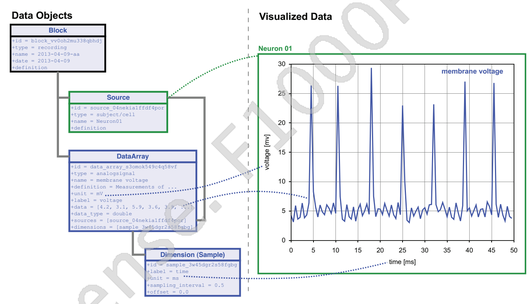
Provides a standard method for storing the basic electrophysiology types in HDF5 along with machine readable metadata needed to interpret the data. The standard does not constrain the location of data within the HDF5 file and allows adding new data types and metadata in a structured manner. This flexibility enables constructing new conventions as needed while still maintaining the capability to interpret the basic electrophysiology data types. For details see poster P40.

Pandora

Uses a two-tiered approach for storing neuroscience data and associated metadata in HDF5 files.

1) The data model is able to describe all sorts of multidimensional scientific data. It includes units, data types, axis labels and other properties necessary to understand and use the data for analysis, or to automatically create a plot. Additional features include elements for the definition of regions and points of interest, enabling tagging data with additional information about events, stimulus conditions or hardware settings. These features enable storing annotated datasets based on common data-types like time series, waveforms, spike trains, images and image stacks. Flexibility for storing metadata is provided by using the odML data model.
2) The schema definition for HDF5 files represents all elements of the data model. It describes in detail where certain information and data is stored and how numerical data and relationships between data objects are represented.

The main elements of the data model are shown below for storing a time series of a recorded membrane potential. Block elements group data elements that belong to a certain experiment or experimental trial. Sources describe the provenance of data elements, like a corresponding neuron or a recording channel. DataArrays and their Dimension descriptors describe the recorded data.
For further details see the Poster by Stoewer et al (P54).



Semantic Web-based approach

The goal of the semantic web is to allow anyone to publish data on the web which is semantically related to other data that is online. Achieving this requires standards for how data are semantically identified and for expressing relationships between data. Since a standards for data sharing must also define methods for specifying machine-processable properties of data and relationships, perhaps the technologies used for the semantic web could be used to develop a standard format for data sharing.
How this could be implemented:

1. A shared data set (collection of files being shared as one unit) is assigned a URI.
2. In HDF5 each group or dataset is assigned a unique identifier that can be referenced in RDF (semantic web Resource Description Framework).
3. Also define a convention for referencing parts of a dataset.
4. The above features allow making arbitrary statements about data and relations between data.
5. Define ontologies for electrophysiology data types and relationships.
6. A "SemanticSitemap" (www.w3.org/wiki/SemanticSitemap) could be used to organize all data and metadata files in a shared data set.
7. Shared data could be stored on a repository that has a SPARQL endpoint (which allows doing searches in RDF data). Thus the very RDF data used to describe data sets would also be available for searching the data sets.

Combining SQLite and HDF5

A standard for storing data in HDF5 should store metadata needed to interpret the data in the same file as the data. However, for purposes of searching for data of interest, sometimes it may be useful to store some metadata separately in a relational data base. An example of this is a hippocampus data set which will soon be available at CRCNS.org (hc3, see next section). There are over 400 files in the data set. To allow effectively searching for data of interest, a SQLite database was created which contained tables of metadata (http://sqlite.org/). SQLite provides an easy way to implement a relational database containing metadata that could be downloaded as part of a large dataset.

Use cases (test data sets).

Test case data sets will be used to evaluate possible standards.

Blackrock Microsystems sample file

Blackrock Microsystems has provided a sample data file containing data from multiple channels in HDF5 format. The file has data from 144 channels. Included are continuous time series signal, and time series segments which contain spike times and waveforms, and sorted unit IDs and two experimental event channels. This sample data file has been converted to epHDF in a way that largely preserved the original file organization and reduces the storage size. See poster P40 for details.

CRCNS.org pvc data sets

Three data sets from primary visual cortex, hosted at CRCNS.org have been used to test HDF5 formats. The pvc-1 data set contains time series segments, and also the corresponding visual stimulus which are jpg movie frames. The pvc-2 dataset has single unit neural event. Stimuli are 12x12 gray scale movie frames and 1D bars. The pvc-3 dataset has spike times for ten neurons. Stimuli are 64 x 64 gray scale move frames and moving bars generated by Vision Egg software. A challenge in developing a format for these this data set is providing metadata that synchronizes the stimulus with the neural responses.

CRCNS.org hc-2 and hc-2 data sets

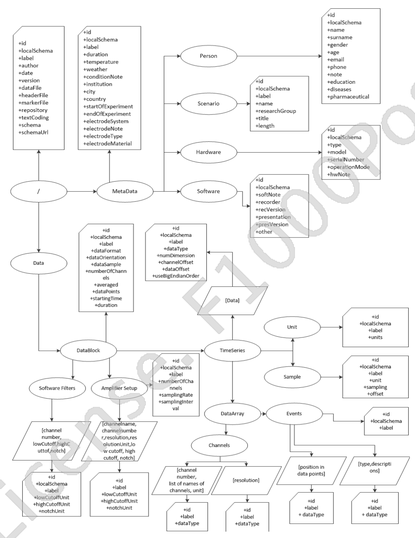
These data sets are the hippocampus of freely moving rats. They contain time series and time series segment data from multiple electrodes (shanks). In addition, there are feature vectors used for spike sorting. A challenge in developing a format for these data sets are that there are many recording sessions and to allow searching for sessions of interest, metadata describing the sessions and units are stored in data tables separately from the data. This data is of the same type as used for the Klusta-Team Kwik-format, (described left) but are not stored in HDF5.

Eglen MEA recording format

A team led by Stephen Eglen is creating a repository of MEA (multielectrode array) recordings from retina. The neural data consists of spike times from each electrode. The data is stored in HDF5 using three arrays: one with electrode positions (epos), another with the number of spikes detected on each electrode (sCounts), then the third with times of spikes from all electrodes concatenated (spikes). Metadata in included to allow the the data to be loaded into the R package for processing.

Hdf5Manager

Hdf5Manager is a software tool for annotating and storing EEG data measured in the neuroinformatics laboratory at the University of West Bohemia in HDF5. It was developed as part of masters thesis by Jan Rericha. The data are recorded by Brain Vision Recorder software into its proprietary data format. Much metadata is also included with the data. The data model is shown below.



Conclusion

There are a range of issues to consider for developing a standard for storing electrophysiology data in HDF5. The minimum requirements are storage of basic electrophysiology data types and metadata needed to interpret them. Additional features could be included in a standard that can provide benefits, but may have costs. It might be possible to use an existing standard such as NeXus or NetCDF4 for storing electrophysiology data in HDF5. Or a new standard could be developed based on epHDF, Pandora, the Semantic Web, or using a combination of these approaches.

Acknowledgments: This work was conducted within the Electrophysiology Task Force of the INCF Program on Standards for Data Sharing. Funding for J.L. Teeters and F.T. Sommer provided through NSF grant 0855272.