Neuron
NeuroView

Neurodata Without Borders: Creating a Common Data Format for Neurophysiology

Jeffery L. Teeters,[1] Keith Godfrey,[2] Rob Young,[2] Chinh Dang,[2] Claudia Friedsam,[3] Barry Wark,[3] Hiroki Asari,[4] Simon Peron,[5] Nuo Li,[5] Adrien Peyrache,[6] Gennady Denisov,[5] Joshua H. Siegle,[2] Shawn R. Olsen,[2] Christopher Martin,[7] Miyoung Chun,[7] Shreejoy Tripathy,[8] Timothy J. Blanche,[1] Kenneth Harris,[9,10] György Buzsáki,[6] Christof Koch,[2] Markus Meister,[4] Karel Svoboda,[5] and Friedrich T. Sommer[1,*]
[1]Redwood Center for Theoretical Neuroscience & Helen Wills Neuroscience Institute, University of California, Berkeley, Berkeley, CA 94720, USA
[2]Allen Institute for Brain Science, 615 Westlake Avenue North, Seattle, WA 98109, USA
[3]Physion LLC, 1 Broadway, 14th Floor, Cambridge, MA 02141, USA
[4]Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, CA 91125, USA
[5]Janelia Research Campus, 19700 Helix Drive, Ashburn, VA 20147, USA
[6]School of Medicine, NYU Neuroscience Institute, New York University, East River Science Park, 450 East 29th Street, New York, NY 10016, USA
[7]The Kavli Foundation, 1801 Solar Drive, Suite 250, Oxnard, CA 93030, USA
[8]Centre for High-Throughput Biology, University of British Columbia, 2329 West Mall, Vancouver, BC V6T 1Z4, Canada
[9]UCL Institute of Neurology, University College London, London WC1N 3BG, UK
[10]UCL Department of Neuroscience, Physiology and Pharmacology, London WC1E 6DE, UK
*Correspondence: fsommer@berkeley.edu
http://dx.doi.org/10.1016/j.neuron.2015.10.025

The Neurodata Without Borders (NWB) initiative promotes data standardization in neuroscience to increase research reproducibility and opportunities. In the first NWB pilot project, neurophysiologists and software developers produced a common data format for recordings and metadata of cellular electrophysiology and optical imaging experiments. The format specification, application programming interfaces, and sample datasets have been released.

Background

Progress in science is increasingly driven by sharing data. Astronomy, genomics, and, more recently, image-based cell biology have adopted standards that facilitate data sharing. Large collaborative projects such as the genome projects pool data with the same format into massive databases, permitting mega- and meta-analyses (respectively, pooled analysis of raw data and pooled analysis of published results; Costafreda, 2009), and the development and use of common tools for analysis and modeling. In neuroscience, concerted efforts have emerged only recently to enable and leverage large-scale data sharing, such as those related to neuroimaging (Poldrack and Gorgolewski, 2014). Further, communities working on particular systems, such as the fly and the worm, have established standards for sharing reagents and data (http://www.wormbase.org, http://flybase.org). But neurophysiology research is still mostly done in laboratories that pursue diverse questions about different organisms using a great variety of individually tailored tools. The output is mainly traditional research papers, with the original data rarely accessible. While there have been some efforts to make neurophysiological data available online under more or less standardized conditions (e.g., http://neurodatabase.org, http://brainliner.jp, http://www.g-node.org, http://www.neuroelectro.org, http://www.carmen.org.uk, https://www.ieeg.org), most data is distributed in the native format of individual labs (Gardner et al., 2001; Herz et al., 2008; Teeters et al., 2008). Progress has been made toward crafting a common description of raw neurophysiology data (Neuroshare, http://neuroshare.sourceforge.net; Neo, http://neuralensemble.org/neo; CARMEN NDF, http://www.carmen.org.uk; INCF task force document, http://tinyurl.com/INCF-ephys-req-v0-72), but there is still no widely adopted standard, let alone a single format that can accommodate all the metadata needed to conduct meaningful analyses. As a consequence, the time and effort required for data discovery and analysis are unnecessarily high. Further, the lack of a common format has made comparison across techniques and laboratories difficult and replication of specific experiments almost impossible, significantly slowing overall progress in the field.

Neurodata Without Borders (NWB) is a broad initiative to standardize neuroscience data and to remove barriers to data sharing among neuroscientists (http://nwb.org). Here we describe the NWB: Neurophysiology pilot project, the first effort of this initiative. In this project, experimental and computational neuroscientists collaborated with developers over a year to produce a unified data format for cell-based neurophysiology data. We will describe the evolution of this focused and highly collaborative project. Further, we discuss how the resulting format was influenced by previous approaches and how it could help unify neurophysiology data and impact the future of neuroscience.

Approach

A particular challenge in developing formats for neurophysiology is that neural signals are often impossible to interpret without access to the complex metadata that accompanies each experiment. This includes information about stimulus properties, the configuration of the recording hardware, and—in the case of in vivo

Table 1. Systems to Store and Process Neurophysiology Data that Were Presented at the First NWB: Neurophysiology Project Meeting

System	Summary	Presenter and References
odML	Method to organize and store metadata	Thomas Wachtler, LMU Munich (Grewe et al., 2011; Sobolev et al., 2014)
Neo	Python object model for representing electrophysiology data and workflows	Michael Denker, Juelich (Garcia et al., 2014; M. Denker et al., 2011, Front. Neuroinform., abstract)
NIX (HDF5)	Simple data model for storing neuroscience data	Christian Kellner, LMU Munich (A. Stoewer et al., 2014, Front. Neuroinform., abstract)
LBNL Brain (HDF5)	Data format specified via JSON. "Managed objects" and "relationship attributes" specify semantic components	Oliver Ruebel, LBNL (Rübel et al., 2015)
Orca (HDF5)	Format developed at the Allen Institute for neurophysiology data	Keith Godfrey, Allen Institute
KWIK (HDF5)	Format used in Klusta Suite, an open-source spike sorting software	Kenneth Harris, UCL (Kadir et al., 2014; Rossant et al., 2015)
EEGBase	Portal for managing EEG data using a relational and NoSQL database	Vaclav Papez, University of West Bohemia (Mouček et al., 2014)
MEF	Format for electrophysiology data; has compression, encryption, and redundancy	Matt Stead, Mayo Clinic (Brinkmann et al., 2009)
NeuroElectro	Mining published literature for physiological properties of cell types	Shreejoy Tripathy, UBC (Tripathy et al., 2015)
Thunder and Lightning	Tools and formats for large-scale exploratory data analysis	Jeremy Freeman, Janelia Farm (Freeman, 2015)
Open Ephys	Initiative to develop open-source tools for electrophysiology	Joshua Siegle, Allen Institute (Siegle et al., 2015)

Systems to store and process neurophysiology data that were presented at the first NWB: Neurophysiology project meeting. Left column: system name and label (HDF5) for the systems that use HDF5. Right column: presenter name and references. Slides for many of the presentations are available at: http://crcns.org/NWB/hackathon-1.

experiments—any number of variables describing a subject's behavioral state. While it is possible to store such data inside any generic data container, there are two main challenges to making data easy to interpret and share. The first is to express all the different pieces of data and the essential interrelationships between them, such as the relative timing between stimuli and neural signals. Anticipating all possible experiments or *use cases* is infeasible because of the constantly evolving experimental paradigms and improving instrumentation. The second challenge is to develop a storage scheme, which enables users to access similar data elements in a common, compatible way. Many use cases share common data elements, for example, a recording technique. To date, this commonality has not been exploited, preventing methods that can access data from one lab to work on data from another. Imagine the difficulties in borrowing a computer or piano, if keyboards lacked a standard. The goal of a common neurophysiology format is to advance to a situation comparable to standardized keyboards, which made pianos, typewriters, and computers sharable resources.

The approach of the NWB: Neurophysiology pilot project was to:

- Tackle a challenging but manageable multitude of use cases.
- Employ an approach driven by the domain problem rather than by computer science methods, but be aware of the relevant existing solutions.
- Provide a formal definition of format properties, enabling extensions to new use cases.
- Finish the project within one year.

This approach was formulated at a meeting in Chicago organized by the Kavli Foundation, attended by Maryann Martone (UCSD), Sean Hill (EPFL, INCF), and Robert Wells (GE) and by some of the authors. The project started in July 2014 with a team of two full-time software developers, one full-time neuroinformaticist/computer scientist, and part-time collaborators from Caltech, Janelia Farm, NYU, UC Berkeley, and the Allen Institute for Brain Science. In addition, various outside experts contributed significantly who attended one or both of the project meetings at Janelia Farm. Meeting 1 took place in November 2014 and Meeting 2 in May 2015, and the project ended in July 2015 with the release of its products (http://github.com/NeurodataWithoutBorders; summary in Supplemental Information, section A).

Existing Methods for Neurophysiology Data

The team started by defining requirements for the data format and surveying existing neurophysiology formats (http://crcns.org/files/data/nwb/nwb_hackathon1.pdf). Based on this information, experts were invited to Meeting 1 to brief the team about existing efforts related to neurophysiology data formats, summarized in Table 1. (A summary of this meeting is at https://incf.org/activities/projects/neurodata-without-borders-meeting-report.)

In addition to the items of Table 1, the team reviewed other sources, including the requirements document of the INCF electrophysiology task force, which enumerates the basic data structures required for sharing neurophysiology data (http://tinyurl.com/INCF-ephys-req-v0-72).

Use Cases, Data Model, and Goals

Central to the development of the data format was a diverse set of use cases,

each one presented and discussed at Meeting 1. These use cases included rodent experiments with different behavioral paradigms and recording techniques from published studies; for details, see Supplemental Information, section B. The development team interacted with the use case experts to compile the data and metadata requirements of all use cases in the so-called "what" document. This document was started at Meeting 1, with input from many of the authors, Thomas Cleland (Cornell), and Matt Stead (Mayo Clinic).

The "what" document is organized into sections called modules. Each module contained pseudocode, describing the data, metadata, and their relationships for a particular aspect of the experiment. For example, there are modules for different recording techniques, such as whole-cell intracellular recording or optical imaging, and for different experimental paradigms, such as sensory stimulation or behavior. In the course of the project, this information was translated into a data model, the "NWB data model." Excerpts from the "what" document are given in Supplemental Information, section B.

With the data model established, the creation of the data format required mapping entities of the data model to locations within a file. The team identified three main design goals for the format: (1) inclusion of all entities of the NWB data model, (2) easy usage of the format on all major computer platforms, and (3) easy readability of the data files without requiring a special API.

The team chose HDF5 (http://www.hdfgroup.org/HDF5) as the data container for the format because its features seemed well aligned with the goals 2 and 3. First, it is a well-supported and mature standard that is available on Mac, Windows, and Linux and includes a graphical utility (HDFView), which allows easy browsing of HDF5 files. Second, HDF5 allows the hierarchical organization of data, similar to a file system within a file. "HDF5 groups" correspond to the directories, and "HDF5 datasets" store arbitrary array-type data and correspond to files. Third, the linking feature of HDF5 enables data stored in one location to be transparently accessed from multiple locations in the hierarchy, even when the data is external to the file. Finally, the ongoing accessibility of HDF-stored data is the mission of the HDF Group, a nonprofit that is the steward of the technology.

The NWB Format Prototype

The Allen Institute Orca format was selected as a starting point for the NWB prototype format because of its close match to the design goals 1 and 3. The NWB data model was incorporated and improvements were made, some suggested by project collaborators who had tested the Orca format. Written documentation was created to convey the format features and technical specification.

The NWB format prototype covering most of the use cases was delivered in March 2015 and tested by the experimental team members. In addition, tool developers who attended Meeting 1 provided feedback. Some of the feedback expressed concerns about the methods used to specify and implement the format. Because the consistency between implemented features and documentation could not be checked automatically, the documentation did not completely describe the implementation, which is a frequent problem when a software specification is evolving. Also, the tool developers expressed reservations about adopting a standard that left anything to interpretation. A related problem was that any changes to the format required modifying the code implementing the API. This would have made extensions to the format difficult to manage, especially if there were multiple labs creating extensions.

Incorporation of a Specification Language

To overcome the shortcomings of the format prototype, an API was developed based on a specification language in which the features of the format are described in a JSON-like syntax that is both human and machine readable. Defining the format with a specification language was somewhat inspired by the NeXus scientific format (http://www.nexusformat.org). Other examples of APIs that are based on a specification language include swagger (http://swagger.io) and API Blueprint (https://apiblueprint.org).

The specification file (for the NWB format: nwb_core.py) serves as the single definitive source for the format specification. It contains two sections, one defining the structures (arrays, metadata, and relationships) of the data model and another specifying where in the HDF5 file the structures are stored. Examples for how elements of the NWB data model are expressed with the specification language are given in Supplemental Information, section C.

Calls to the API for creating a file are automatically checked to ensure that the file conforms to the specification. Further, it is easy to change or extend the format because only the specification file must be modified and not the API software. This also facilitates the creation of APIs for multiple programming languages. So far, a Python and MATLAB write API have been implemented. Code examples for how to use the APIs in the different programming languages are provided in Supplemental Information, section D.

The specification language incorporates a namespace mechanism (similar to XML namespaces) allowing extensions to the format to be independently created and shared between labs. Such extensions could be centralized using online version control systems like GitHub (e.g., http://github.com/NeurodataWithoutBorders), and popular extensions could be considered for inclusion in the standard.

Summary of Current Format Features

It is important to emphasize that the current release of the NWB format offers a possible starting point for unifying neurophysiology data, not a final solution. The purpose of the release is to engage with the broader community. Although the pilot project has ended, the NWB initiative will continue to support improvements and extensions of the format suggested by users. Characteristic features of the current alpha version of the NWB format are:

- A general time series class with subclasses for many specific types of data. Each time series has labels (HDF5 attributes) that identify its structure and content, and each subclass contains the metadata required to interpret the data within it. Tools that are written to operate

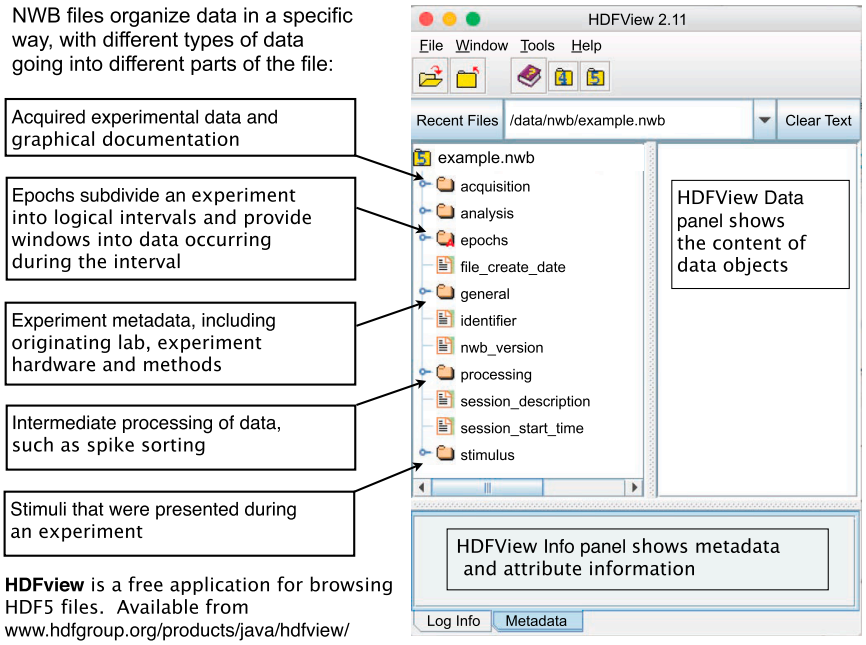

Figure 1. Layout of an NWB File as Shown when Opened with HDFView

on data of a class will also function on data of its subclasses.

- Processed data that are derived from acquired data, such as the results of spike sorting or image segmentation, are also stored with labels that identify structure and content. These labels allow software tools to quickly determine whether the file contains the necessary data for a specific analysis or for subsequent processing.
- Files are organized by different kinds of data. For instance: recorded data, stimuli, and data resulting from an analysis are kept separate, which enhances human readability; see Figure 1.
- Mechanism for linking information about intervals directly to the time series data for which the information applies. For example, recordings can be stored contiguously and trial structure can be added using this mechanism.
- Compatibility with HDFView; see Figure 1.
- Format features expressed in the specification language are human- and machine-readable.
- Easy extensibility to new use cases through the specification language.

The current release includes the NWB format specification and basic application programming interfaces for writing data files in Python and MATLAB, and samples of use-case datasets translated into the new data format (see Supplemental Information, section A, for overview).

Discussion

Project Evolution and Relationship to Existing Neurophysiology Data Formats

The NWB: Neurophysiology pilot project was unusual in many regards. First, the time horizon of one year was brief, given the considerable challenge of developing a data format, but it kept the team focused on a tangible outcome. Second, the project involved a close collaboration between software developers and many domain experts (neuroscientists). While this collaboration sometimes made it difficult to arrive at a consensus, it was critical to the solution we found. Third, a unique feature of the project was the breadth of the initial domain it targeted, a challenging combination of datasets from different laboratories and institutions. The varied use cases included whole-cell and extracellular electrophysiology, as well as optical imaging.

The NWB format includes the description of the data model using the specification language and a method for mapping the data into files. This connection of a data model to a storage method is in common with Neo (Garcia et al., 2014), NIX (A. Stoewer et al., 2014, Front. Neuroinform., abstract), SignalML (Durka and Ircha, 2004), and the NeXus format in particle physics (http://www.nexusformat.org). The NWB format differs from these systems by its detailed data model, which was designed with the representative set of NWB use cases in mind. Therefore, it can determine with less ambiguity how data elements of these use cases should be stored, as compared to formats with more generic data models or developed for other domains. Because the NWB data model is defined in the specification language, the format is also flexible to accommodate new use cases. Further, the separation between data model and storage method also can enable options for multiple back-end stores, like in SignalML and NeXus. The Neuroshare API has successfully leveraged this principle for accessing electrophysiology data in different vendor formats.

The team considered the possibility of building the NWB format directly onto more established systems, as thoroughly as the one-year time horizon permitted. Since the project started with the development of a specific data model, the use of any other system would have required adding an additional layer for translating between data models. For example, the NIX format, one of the best developed at the time, would have been able to handle almost all of the NWB data model. But an additional layer, required for mapping the NWB data model to the more generic NIX data model, would have added to the complexity of the solution. Further, once the data is expressed in the more generic data model, it would have been hard to group data according to the more specific NWB data model. This would have likely resulted in HDF5 files that were more difficult to understand using HDFView.

Aside from the described differences, the current NWB format was strongly influenced by other existing systems. The NWB specification language has similarities to elements of the LBNL Brain format (Rübel et al., 2015), and the definition of dimensions in the specification language was influenced by the NIX format. In addition, our design was informed by

the INCF requirements document (http://tinyurl.com/INCF-ephys-req-v0-72), and the format's high-level design was influenced by the KWIK format (Kadir et al., 2014; Rossant et al., 2015).

Potential Avenues to Unify Neurophysiology Datasets

The goal of this NWB: Neurophysiology pilot project was to derive a common description of experimental cellular datasets from different experiments and labs. We hope that widespread adoption of such a description will improve reproducibility of neuroscience research while at the same time opening new research avenues. Due to the rapid advance of experimental neuroscience techniques even within the short duration of this project, the notion of such a common description was a moving target. At Meeting 1, the experimentalists in the project considered it important that the data organization within files be common among datasets so that the data can be interpreted even without an API. A large fraction of attendees at Meeting 2 agreed that the efficiency of data processing might impose other important constraints on how the data should be stored. Thus, a stronger emphasis was put on the organization of the data at the level of the data model. Related to this, the attendees supported the addition of a specification language to formally describe the data model and to make it extensible to new use cases.

As a result of the project dynamics, the current NWB format offers two potential avenues toward a common description of datasets. One is a convention for how the data are arranged in the HDF5 file. The other, perhaps more powerful, approach is through generating a read/write API that can work with other formats if they are compatible with the NWB data model or extensions of it. Such a translation between formats was pioneered by the Neuroshare API, but restricted to essentially only the recording data. Leveraging the separation between data model and storage method, an enhanced version of specification language could be developed to describe other formats that store data and metadata of experiments. Since the NWB data model can describe many types of neurophysiology experiments (and also can be easily extended), it could constitute a quite general conduit for interoperability between data formats. Thus, data model and specification language of the NWB format could be used in methods for unifying data in different data formats without the need to reformat any data.

Potential Impact of a Unified Data Format on Scientific Progress

The aim of NWB for a unified description of neurophysiology data was also pursued by prior efforts (Gardner et al., 2001, 2008; Gibson et al., 2009; Grewe et al., 2011; Y. Le Franc et al., 2014, Front. Neuroinform., abstract; J.L. Teeters et al., 2013, Front. Neuroinform., abstract). While a unified data format may seem like a technical advance of little relevance for scientific progress, it is surprising how transformative a well-executed format can be. Astronomy provides a concrete and instructive example of how a data format can profoundly change the culture of a field (McCray, 2014). Throughout the 20th century, an astronomer would likely describe his or her expertise by reference to the wavelength of light used by their observational tool of choice: e.g., an "optical" or "radio astronomer." Today, astronomers are able to study a particular question by seamlessly combining data from many different telescopes at many different wavelengths (Abt, 1993). This shift from tools to questions is largely due to the fact that astronomy data is available in one format, known as FITS (the Flexible Image Transport System). Thus, the presence of a unified data format has fundamentally changed the culture in astronomy. Astronomers now introduce themselves by the actual subjects they study—e.g. as a "stellar" or "galactic astronomer."

The history of the FITS format in astronomy might give us a glimpse of the possible effects of unifying neuroscience data: FITS required careful consideration of the unique needs and use cases brought forward by different groups in order to be truly inclusive. Then, even once the format was agreed upon in 1979, many years of outreach and education were required to ensure adoption by the entire community. To this day, a working group of the International Astronomical Union carefully considers any additions to the format and also reviews and promulgates recommended practices. Finally, if a format is done well, its use can spread well beyond its imagined purposes, so perhaps it shouldn't be a surprise that, in 2010, the Vatican Library announced that it would scan rare manuscripts using the FITS format.

The most immediate impact of a common data format to neuroscience would be facilitation of data sharing and creating opportunities for the development of open-source analysis tools. Currently, most tools for data analysis are developed for a specific format and cannot be easily applied to data in other formats.

SUPPLEMENTAL INFORMATION

Supplemental Information includes Supplemental Data on NWB and two figures and can be found with this article online at http://dx.doi.org/10.1016/j.neuron.2015.10.025.

ACKNOWLEDGMENTS

The Kavli Foundation, General Electric, Howard Hughes Medical Institute, the Allen Institute for Brain Science, the National Science Foundation (grant 0855272), and the International Neuroinformatics Coordinating Facility provided the financial support for conducting the NWB: Neurophysiology pilot project. Janelia Research Campus administered and hosted the two project meetings. The project was heavily dependent on exchanges of the project team with external experts who provided critical input. We thank Jim Berg, Aleena Garner, and Kenji Mitzuseki for sharing experimental data; Jack Waters for help with defining the data model for optophysiology; David Feng and Lydia Ng for support with technical issues of HDF5; and Anton Arkhipov, Tsai-Wen Chen, Saskia De Vries, Severine Durand, Nathan Gouwens, and Zengcai Guo for reviewing the prototype version of the NWB format.

REFERENCES

Abt, A.A. (1993). Publications of the Astronomical Society of the Pacific *105*, 437–439.

Brinkmann, B.H., Bower, M.R., Stengel, K.A., Worrell, G.A., and Stead, M. (2009). J. Neurosci. Methods *180*, 185–192.

Costafreda, S.G. (2009). Front. Neuroinform. *3*, 33.

Durka, P.J., and Ircha, D. (2004). Comput. Methods Programs Biomed. *76*, 253–259.

Freeman, J. (2015). Curr. Opin. Neurobiol. *32*, 156–163.

Garcia, S., Guarino, D., Jaillet, F., Jennings, T., Pröpper, R., Rautenberg, P.L., Rodgers, C.C., Sobolev, A., Wachtler, T., Yger, P., and Davison, A.P. (2014). Front. Neuroinform. *8*, 10.

Gardner, D., Knuth, K.H., Abato, M., Erde, S.M., White, T., DeBellis, R., and Gardner, E.P. (2001). J. Am. Med. Inform. Assoc. *8*, 17–33.

Gardner, D., Goldberg, D.H., Grafstein, B., Robert, A., and Gardner, E.P. (2008). Neuroinformatics *6*, 161–174.

Gibson, F., Overton, P., Smulders, T., Schultz, S., Eglen, S., Ingram, C., Panzeri, S., Bream, P., Whittington, M., Sernagor, E., et al. (2009). Nature Precedings. http://precedings.nature.com/documents/1720/version/1.

Grewe, J., Wachtler, T., and Benda, J. (2011). Front. Neuroinform. *5*, 16.

Herz, A.V.M., Meier, R., Nawrot, M.P., Schiegel, W., and Zito, T. (2008). Neural Netw. *21*, 1070–1075.

Kadir, S.N., Goodman, D.F., and Harris, K.D. (2014). Neural Comput. *26*, 2379–2394.

McCray, W.P. (2014). Technology and Culture *55*, 908–944.

Mouček, R., Ježek, P., Vařeka, L., Rondík, T., Brůha, P., Papež, V., Mautner, P., Novotný, J., Prokop, T., and Stěbeták, J. (2014). Front. Neuroinform. *8*, 20.

Poldrack, R.A., and Gorgolewski, K.J. (2014). Nat. Neurosci. *17*, 1510–1517.

Rossant, C., Kadir, S.N., Goodman, D.F.M., Schulman, J., Belluscio, M., Buzsaki, G., and Harris, K.D. (2015). bioRxiv. http://dx.doi.org/10.1101/015198.

Rübel, O., Prabhat, Denes, P., Conant, D., Chang, E., and Bouchard, K. (2015). bioRxiv. http://dx.doi.org/10.1101/024521.

Siegle, J.H., Hale, G.J., Newman, J.P., and Voigts, J. (2015). Curr. Opin. Neurobiol. *32*, 53–59.

Sobolev, A., Stoewer, A., Leonhardt, A., Rautenberg, P.L., Kellner, C.J., Garbers, C., and Wachtler, T. (2014). Front. Neuroinform. *8*, 32.

Teeters, J.L., Harris, K.D., Millman, K.J., Olshausen, B.A., and Sommer, F.T. (2008). Neuroinformatics *6*, 47–55.

Tripathy, S.J., Burton, S.D., Geramita, M., Gerkin, R.C., and Urban, N.N. (2015). J. Neurophysiol. *113*, 3474–3489.

Neuron

Supplemental Information

Neurodata Without Borders: Creating a Common Data Format for Neurophysiology

Jeffery L. Teeters, Keith Godfrey, Rob Young, Chinh Dang, Claudia Friedsam, Barry Wark, Hiroki Asari, Simon Peron, Nuo Li, Adrien Peyrache, Gennady Denisov, Joshua H. Siegle, Shawn R. Olsen, Christopher Martin, Miyoung Chun, Shreejoy Tripathy, Timothy J. Blanche, Kenneth Harris, György Buzsáki, Christof Koch, Markus Meister, Karel Svoboda, and Friedrich T. Sommer

SUPPLEMENT

A. Releases of the NWB:Neurophysiology pilot project
The July 2015 release of the NWB initiative includes code and documentation for the specification language, APIs, and tools. These are available at: https://github.com/NeurodataWithoutBorders.

A1. Format Specifications

Description of features of the NWB format and its specification language.
https://github.com/NeurodataWithoutBorders/specification

- **Format Specification** (PDF) - Specification of the format in English text.
- **NWB Schema** - JSON file that uses specification language to define the NWB format. (Created from nwb_core.py used with the Python API).
- **Specification Language Documentation** (PDF) –Detailed reference for the NWB specification language.

A2. Application Programming Interfaces (API)

Python and MATLAB application programming interfaces (API) are provided for data publishers. Both implementations refer to a schema file written in the specification language.

- **Write API** (Python) - Github repository for the Python Write API's source code. https://github.com/NeurodataWithoutBorders/api-python
- **Write API (MATLAB) -** Github repository for the MATLAB Write API's source code and unit tests. https://github.com/NeurodataWithoutBorders/api-matlab

A3. NWB Tools

The following is a list of tools that are available to work with the NWB format.

- **File Diff Tool** (Python) - Script that compares two files in the NWB format. https://github.com/NeurodataWithoutBorders/diff
- **Transform MATLAB Data File to HDF5** (Python) - Recursively transforms data in the MATLAB format to a more easily processed HDF5 format. https://github.com/NeurodataWithoutBorders/mat2h5

A4. Additional Allen Institute NWB Resources

The Allen Institute for Brain Science provides a software development kit for the Allen Cell Types Database and a write API for the NWB format written in Python. The API contains a few features not yet included in the official version and is not currently based on the specification language.

- **Allen SDK** (Python) - Software development kit for reading experimental data and running models from The Allen Cell Types Database. http://alleninstitute.github.io/AllenSDK/
- **Python Write API for the NWB Format** (Python) - Object oriented API for writing to a NWB file. https://github.com/AllenInstitute/nwb-api

A5. Exemplar Datasets

For all of the use cases, sample datasets in the new format are available at CRCNS.org (https://crcns.org/NWB/exemplar-data-sets). In the following list, the id in parenthesis (e.g. "hc-3") indicates the name of the data set at CRCNS.org containing the original data, i.e. before conversion to the NWB format, and the link is to example files of the data in the NWB format.

- **Rat Hippocampus (hc-3)** – Multi-unit recordings from different rat hippocampal regions while the animals were performing several behavioral tasks. Data from the Buzsáki lab. https://portal.nersc.gov/project/crcns/download/nwb-1/hc-3
- **Mouse Visual Cortex (pvc-6)** – In vitro intracellular recording and staining of a single neuron in the visual cortex of a mouse. Data from the Allen Institute for Brain Science. https://portal.nersc.gov/project/crcns/download/nwb-1/allenInst
- **Mouse Premotor Cortex (alm-1)** – Extracellular recordings from neurons in the anterior lateral

motor cortex of adult mice performing a tactile decision behavior. Data from the Svoboda lab. https://portal.nersc.gov/project/crcns/download/nwb-1/alm-1

- **Mouse Vibrissal S1 (ssc-1)** – Calcium imaging from vibrissal somato-sensory cortex 1 in mice performing a pole localization task. Data from the Svoboda lab. https://portal.nersc.gov/project/crcns/download/nwb-1/ssc-1
- **Mouse Retina (ret-1)** – Multi-electrode recording on ex-vivo retina, with visual stimulus. Data from the Meister lab. https://portal.nersc.gov/project/crcns/download/nwb-1/ret-1

B. The use cases in the NWB pilot project

The use cases considered in the project included rodent experiments with different behavioral paradigms and recording techniques from published studies. The studies from the Buzsáki lab included multi-electrode recordings in hippocampus and entorhinal cortex in rats exploring mazes (Pastalkova et al., 2008; Mizuseki et al. 2009; 2011; 2014; Mizuseki & Buzsáki, 2013). The studies from the Svoboda lab included extracellular recordings from ALM neurons of adult mice performing a tactile decision behavior (Li et al., 2015), and calcium imaging data from vibrissal S1 in mice performing a pole localization task (Peron et al., 2015). The studies from the Meister lab included single-unit neural responses recorded from isolated retina from mice using a 61-electrode array in response to various visual stimuli (Lefebvre et al., 2008; Zhang et al., 2014). The use cases from the Allen Institute for Brain Science included slice physiology using whole-cell recordings (Berg, 2014), and in-vivo multi-electrode recordings during visual stimulation in cat visual cortex (Blanche, 2005). A list with more detailed information about each of the use cases is at http://crcns.org/NWB/Data_sets.

C. Examples of how data structures are defined in the specification language

In the "what" document the information required to describe a certain type of experiment was formulated in a pseudocode, intended to be independent of any particular storage method. A few examples from the "what" document are given here along with corresponding expressions in the specification language.

C1. Simple pieces of metadata

The following metadata in the "what" document describe properties of the experimental animal or subject. These descriptions consisted of a key-value pair:

```
Species - Text (use biology-wide standard)
Genotype - Text (use biology-wide standard)
Sex - Text (M/F)
```

They are expressed in the specification file (nwb_core.py) by:

```
 "subject/": {
        "species": { "data_type": "text",
               "description": "Species of subject"},
        "genotype": { "data_type": "text",
               "description": "Genotype of subject"},
        "sex": {
               "data_type": "text",
               "description": "Gender of subject"}, ...
    },
```

C2. Array structure containing data and related metadata

In the "what" document, data recorded from electrodes and associated metadata is described by the following pseudocode. The "for statement" (similar to that statement in programming languages) indicates that descriptions inside the loop-block apply to each instance in a collection of data items of the same type:

```
# Recordings from electrodes
For p in probes
       probe source – text (manufacturer, part number)
       probe location
       For g in channel groups
              Estimated tip location
              ground – text
              For c in channels
                     Channel location – numeric: x, y, z (µm)
                     Raw data array index
                     Impedance – numeric, Ohm
```

The section in the pseudocode describing the raw data, electrode locations and impedances are expressed in the specification file (nwb_core.py) by:

```
"<ElectricalSeries>/": {
       "description": "Acquired voltage data from extracellular recordings.",
       "merge": ["<timestamps>/"],
       "attributes": {"ancestry": "TimeSeries, ElectricalSeries" }},
       "data": {
              "description": "Recorded voltage data",
              "dimensions": ["timeIndex", "channelIndex"], # specifies 2-d array
              "data_type": "number",
              "unit": "volt"},
       "electrode_idx": {
              "description": "Indices to electrodes in electrode_map",
              "dimensions": ["channelIndex"],
              "data_type": "int",
              "references":
   "/general/extracellular_ephys/electrode_map.electrode_number"},
       },
"extracellular_ephys/": {
        "electrode_map": {
               "description": "Physical location of electrode, x,y,z in meters",
               "dimensions": ["electrode_number","xyz"], # specifies 2-D array
               "data_type": "number",
               "xyz" : { # definition of dimension xyz
                      "type": "struct",
                      "components": [
                             { "alias": "x", "unit": "meter" },
                             { "alias": "y", "unit": "meter" },
                             { "alias": "z", "unit": "meter" } ] }
        },
       "impedance": {
              "description": "Impedance of electrodes in electrode_map",
              "dimensions": ["electrode_number"], # specifies 1-D array
              "data_type": "text"
        },
}
```

D. Writing and reading NWB files

This section provides some additional information and code for how to create and read NWB files.

D1. Writing NWB files

The Python and MATLAB write APIs for the NWB format contain two main functions: "`make_group`" which makes groups in the HDF5 file and "`set_dataset`" which create HDF5 datasets. In addition there is a function "`set_attr`" for setting attributes. The specific arguments in the calls to these functions (i.e. which groups, datasets and attributes can be created) are determined by the definition of the NWB format written in the specification language. The following example calls demonstrate the usage of the NWB APIs. The examples are taken from scripts creating the NWB file for the "alm-1" dataset at CRCNS.org,

contributed by the Svoboda lab. They illustrate how to write text metadata (the species) and a time series recording of numeric data. The numeric data is called "`lick_trace`" and contains the signal from a photodiode, which detects licking movements. The codes for the Python and MATLAB APIs are very similar.

Python

```
# open NWB file for writing
import nwb_file
f = nwb_file.nwb_file(output_file_name, start_time)

# Store species metadata
s = f.make_group("subject")
s.set_dataset("species", "Mus musculus")

# Create group for lick_trace timeseries
g = f.make_group("<TimeSeries>", "lick_trace",
  path="/acquisition/timeseries")

# Store datasets for timeseries
d = g.set_dataset("data", lick_trace_data)
t1 = g.set_dataset("timestamps", lick_trace_timestamps)
g.set_dataset("num_samples", len(lick_trace_data))
```

MATLAB

```
% open file for writing
f = nwb_file(output_file_name, start_time);

% Store species metadata
s = f.make_group("subject");
s.set_dataset("species", "Mus musculu");

# Store datasets for lick_trace timeseries
g = f.make_group("<TimeSeries>", "lick_trace", "path", ...
  "/acquisition/timeseries");
g.set_dataset("data", lick_trace);
t1 = g.set_dataset("timestamps", timestamps');
g.set_dataset('num_samples', int64(length(lick_trace)));
```

D2. Reading NWB files

Currently the software for the NWB format does not include a read API. A read API is not necessarily required because the format was designed so that data can be easily read using direct calls to HDF5 library functions. Examples of these calls are given below for both Python and MATLAB.

Python

The following example is taken from a script reading the hc-3 dataset at CRCNS.org, contributed by the Buzsáki lab. The script plots data from extracellular recordings in hippocampal areas of rats while they are moving in a confined space. The resulting plots show the position of the animal when the specific neuron generated a spike, color-coded by the approximate direction of movement at the time of the spike. To load the data for plotting for a particular unit, the following code is used:

```
import h5py
# open NWB file
infile = "ec013.156.nwb"
f = h5py.File(infile, "r")
```

```
# function to find all interfaces of a specific type
def get_interfaces(f, itype):
       ilist = []
       proc = f["processing"]
       for k in proc.keys():
       if itype in proc[k].attrs["interfaces"]:
       ilist.append(k)
       return ilist

# find place modules
ilist = get_interfaces(f, "Position")
if not ilist:
  print "Error -- cannot find Position data"
  sys.exit(1)
# use the first; make path to position data
place_mod = "processing/" + ilist[0] + "/Position/position"

# load position time series
pos = f[place_mod]["data"].value
# load times corresponding to each position
post = f[place_mod]["timestamps"].value

# find modules containing UnitTimes interfaces
unit_mods = get_interfaces(f, "UnitTimes")

# loop through all modules containing unit times
for unit_mod in unit_mods:
  # group containing unit times
  grp = f["processing"][unit_mod]["UnitTimes"]

  # loop through each unit
  unit_list = grp["unit_list"].value
  for unit in unit_list:

    # Load spike times for unit
    unit_t = grp[unit]["times"]

    # Now can make plot for unit
```

Once the data is loaded, the plot shown below can be generated. (The code to generate the plot is not shown).

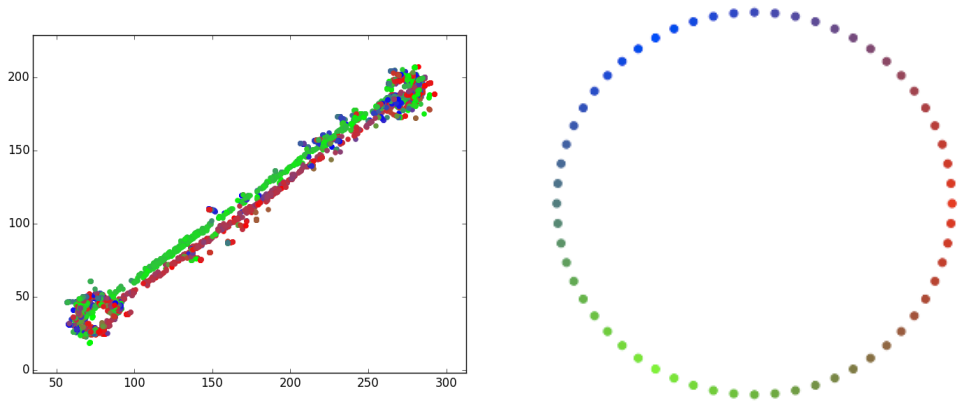

Figure S1: (Left) Plots of spikes from one neuron displayed at the point in the environment that the animal traversed when the spike occurred. Colors indicate the approximate direction of movement at the time of the spike using the color code shown on the right, with movement from center outward. (Data is from hc-3 dataset in CRCNS.org, which was contributed by the Buzsáki lab).

MATLAB

The following sample code is from a script for the alm-1 dataset at CRCNS.org, contributed by the Svoboda lab. The script creates plots showing the response during multiple trials recorded from anterior lateral motor cortex (ALM) neurons of adult mice, for two behaviors (a left or right movement) and also a histogram of the responses. To load the data for plotting a particular unit, the following code is used:

```
% open NWB file
infile = 'NL_example20140905_ANM219037_20131117.nwb';
fid = H5F.open(infile);

% get list of units (this depends on the name of the module being "Units")
unitList = h5read(infile, '/processing/Units/UnitTimes/unit_list');
% get number of trials
epochGroup = H5G.open(fid, '/epochs');
numTrials = H5G.get_info(epochGroup).nlinks;

% get properties of each trial
HitR = zeros(1, numTrials);
HitL = zeros(1, numTrials);
t_TrialStart = zeros(numTrials,1);
for i = 1:numTrials
       trialPath = sprintf('/epochs/Trial_%03i', i);
       trialTypes = h5read(infile, strcat(trialPath, '/tags'));
       trialTypes = deblank(trialTypes);
       if any(ismember(trialTypes, 'HitR'))
              HitR(1,i) = 1;
       end
       if any(ismember(trialTypes, 'HitL'))
              HitL(1,i) = 1;
       end
       startTime = h5read(borgFilePath, strcat(trialPath,'/start_time'));
       t_TrialStart(i,1) = startTime;
end

% Loop through all units (to generate plot for each one)
for i_unit = 1:numel(unitList)
       currUnit = unitList{i_unit}

       % Load unit spike times
       dataPath = strcat('/processing/Units/UnitTimes/', currUnit);
       times = h5read(infile, strcat(dataPath, '/times'));

       % Make plot for unit (code not shown)
end
```

The plot of trial responses and histograms for one unit is shown in Figure S2.

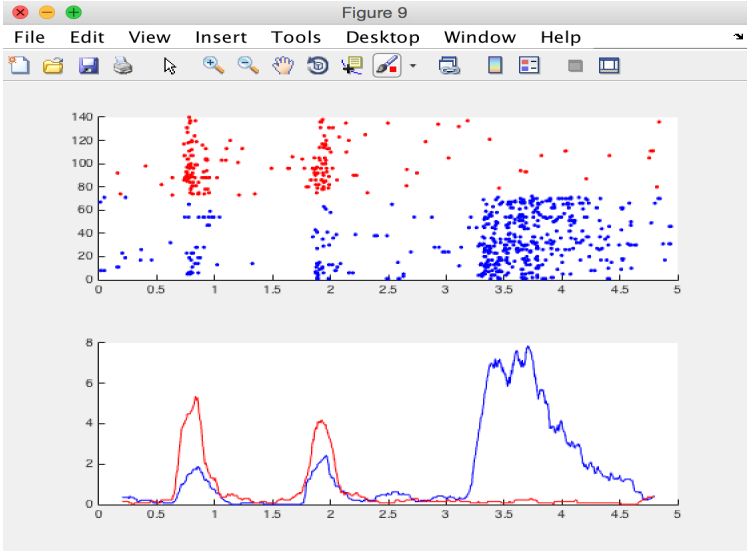

Figure S2: Trial raster plots (upper plot) and PSTHs (lower plot) for one neuron (red = trials with left movement, blue = trials with right movement). From alm-1 dataset in CRCNS.org, contributed by Svoboda lab.

SUPPLEMENTAL REFERENCES

Berg, J. (2014) In vitro whole-cell patch clamp recordings from visual cortex neurons in the adult mouse. CRCNS.org. http://dx.doi.org/10.6080/K0H12ZXD

Blanche T. J., Spacek M. A., Hetke J. F., Swindale N. V. (2005) Polytrodes: high density silicon electrode arrays for large scale multiunit recording. J. Neurophys. 93 (5): 2987-3000. doi: 10.1152/jn.01023.2004 PMID: 15548620

Lefebvre, J.L., Zhang, Y., Meister, M., Wang, X., and Sanes, J.R. (2008) Gamma-Protocadherins regulate neuronal survival but are dispensable for circuit formation in retina. Development 135:4141-4151. doi: 10.1242/dev.027912 PMID: 19029044

Li N., Chen T. W., Guo Z. V., Gerfen C. R., Svoboda K., (2015). A motor cortex circuit for motor planning and movement. *Nature* 519(7541):51-56. doi: 10.1038/nature14178 PMID: 25731172

Mizuseki K, Diba K, Pastalkova E, Teeters J, Sirota A, Buzsáki G. (2014) Neurosharing: large-scale data sets (spike, LFP) recorded from the hippocampal-entorhinal system in behaving rats. F1000Res. 3:98. doi: 10.12688/f1000research.3895.2 PMID: 25075302

Mizuseki K, Buzsáki G. (2013) Preconfigured, skewed distribution of firing rates in the hippocampus and entorhinal cortex. Cell Rep. 4(5):1010-21. doi: 10.1016/j.celrep.2013.07.039 PMID: 23994479

Mizuseki K, Diba K, Pastalkova E, Buzsáki G. (2011) Hippocampal CA1 pyramidal cells form functionally distinct sublayers. Nature Neuroscience 14(9):1174-81. doi: 10.1038/nn.2894 PMID: 21822270

Mizuseki K, Sirota A, Pastalkova E, Buzsáki G. (2009) Theta oscillations provide temporal windows for local circuit computation in the entorhinal-hippocampal loop. Neuron 64(2):267-80. doi: 10.1016/j.neuron.2009.08.037 PMID: 19874793

Pastalkova E, Itskov V, Amarasingham A, Buzsáki G. (2008) Internally generated cell assembly sequences in the rat hippocampus. Science. Sep 5;321(5894):1322-7. doi: 10.1126/science.1159775 PMID: 18772431

Peron, S., Freeman, J., Iyer, V., Guo C., Svoboda, K. (2015) A Cellular Resolution Map of Barrel Cortex Activity during Tactile Behavior. Neuron May 6; Vol 86, Issue 3, 783–799. doi: 10.1016/j.neuron.2015.03.027 PMID: 25913859

Zhang, Y.F., Asari, H., Meister, M. (2014); Multi-electrode recordings from retinal ganglion cells. CRCNS.org. http://dx.doi.org/10.6080/K0RF5RZT