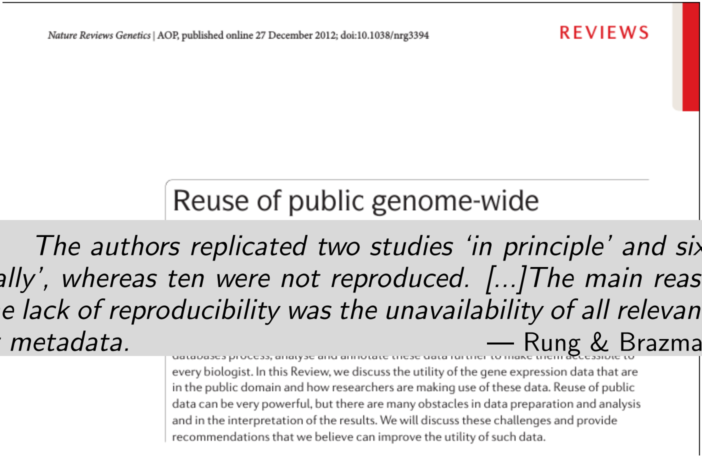
INCF virtual training weeks 2021
– NIX Neo workshop –

Michael Denker, Jan Grewe, Julia Sprenger, & Michael Sonntag

German Neuroinformatics Node

August 30, 2021 – September 01, 2021

Reproducibility crises



Nature Reviews Genetics | AOP, published online 27 December 2012; doi:10.1038/nrg3394

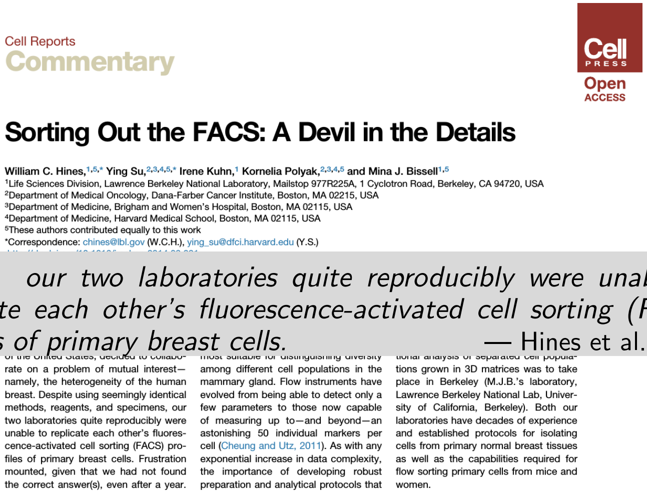
REVIEWS

Reuse of public genome-wide

every biologist. In this Review, we discuss the utility of the gene expression data that are in the public domain and how researchers are making use of these data. Reuse of public data can be very powerful, but there are many obstacles in data preparation and analysis and in the interpretation of the results. We will discuss these challenges and provide recommendations that we believe can improve the utility of such data.

> *The authors replicated two studies 'in principle' and six 'partially', whereas ten were not reproduced. [...]The main reason for the lack of reproducibility was the unavailability of all relevant data or metadata.*
>
> — Rung & Brazma, 2012

Reproducibility crises



Cell Reports
Commentary

Cell PRESS Open ACCESS

Sorting Out the FACS: A Devil in the Details

William C. Hines,[1,5,*] Ying Su,[2,3,4,5,*] Irene Kuhn,[1] Kornelia Polyak,[2,3,4,5] and Mina J. Bissell[1,5]
[1]Life Sciences Division, Lawrence Berkeley National Laboratory, Mailstop 977R225A, 1 Cyclotron Road, Berkeley, CA 94720, USA
[2]Department of Medical Oncology, Dana-Farber Cancer Institute, Boston, MA 02215, USA
[3]Department of Medicine, Brigham and Women's Hospital, Boston, MA 02115, USA
[4]Department of Medicine, Harvard Medical School, Boston, MA 02115, USA
[5]These authors contributed equally to this work
*Correspondence: chines@lbl.gov (W.C.H.), ying_su@dfci.harvard.edu (Y.S.)

of the United States, decided to collaborate on a problem of mutual interest—namely, the heterogeneity of the human breast. Despite using seemingly identical methods, reagents, and specimens, our two laboratories quite reproducibly were unable to replicate each other's fluorescence-activated cell sorting (FACS) profiles of primary breast cells. Frustration mounted, given that we had not found the correct answer(s), even after a year.

most suitable for distinguishing diversity among different cell populations in the mammary gland. Flow instruments have evolved from being able to detect only a few parameters to those now capable of measuring up to—and beyond—an astonishing 50 individual markers per cell (Cheung and Utz, 2011). As with any exponential increase in data complexity, the importance of developing robust preparation and analytical protocols that

tional analysis of separated cell populations grown in 3D matrices was to take place in Berkeley (M.J.B.'s laboratory, Lawrence Berkeley National Lab, University of California, Berkeley). Both our laboratories have decades of experience and established protocols for isolating cells from primary normal breast tissues as well as the capabilities required for flow sorting primary cells from mice and women.

“ *... our two laboratories quite reproducibly were unable to replicate each other's fluorescence-activated cell sorting (FACS) profiles of primary breast cells.* — Hines et al., 2014

Reproducibility crisis

"Challenges in irreproducible research", Nature special, 06.07.2018, ISSN 1476-4687 https://www.nature.com/collections/prbfkwmwvz

> *"More than 70% of researchers have tried and failed to reproduce another scientist's experiments, and more than half have failed to reproduce their own experiments. Those are some of the telling figures that emerged from Nature's survey of 1,576 researchers who took a brief online questionnaire on reproducibility in research."*

Reproducibility crisis

What factors contribute to irreproducible research?

- Selective reporting
- Pressure to publish
- Low statistical power or poor analysis
- Not replicated enough in original lab
- Insufficient oversight/mentoring
- Methods & code unavailable
- Poor experimental design
- Raw data not available from original lab
- Fraud
- Insufficient peer review
- Problems with reproduction efforts
- Technical expertise required for reproduction
- Variability of standard reagents
- Bad luck
- Insufficient metadata

Reproducibility crisis

What factors contribute to irreproducible research?

- Selective reporting
- Pressure to publish
- Low statistical power or poor analysis
- Not replicated enough in original lab
- Insufficient oversight/mentoring
- Methods & code unavailable
- Poor experimental design
- Raw data not available from original lab
- Fraud
- Insufficient peer review
- Problems with reproduction efforts
- Technical expertise required for reproduction
- Variability of standard reagents
- Bad luck
- Insufficient metadata

Reproducibility crisis also in the Neurosciences

There is a lack of standardization

- Earlier (roughly 2007/2008) this was also identified as one problem in neuroscience.

Reproducibility crisis also in the Neurosciences

There is a lack of standardization

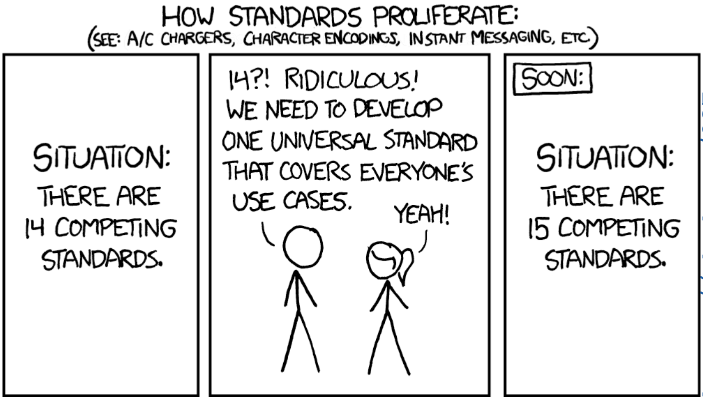
- Earlier (roughly 2007/2008) this was also identified as one problem in neuroscience.
- We need standards and tools to help doing good neuroscience.

Reproducibility crisis also in the Neurosciences

There is a lack of standardization

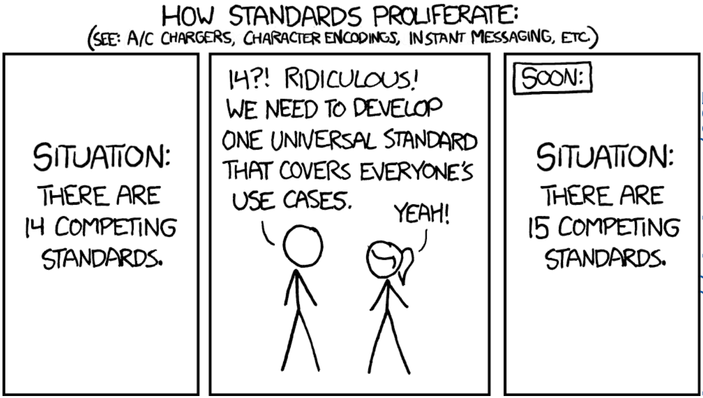
- Earlier (roughly 2007/2008) this was also identified as one problem in neuroscience.
- We need standards and tools to help doing good neuroscience.
- The INCF task force on **Standardization in electrophysiology** was the starting point of the NIX project.

Reproducibility crisis also in the Neurosciences


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

https://xkcd.com/927

Reproducibility crisis also in the Neurosciences


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

https://xkcd.com/927

... but at least some of these are **open-source**!

NIX

Neuroscience **I**nformation E**x**change

Originates from the neuroscience ...
but is not limited to it.

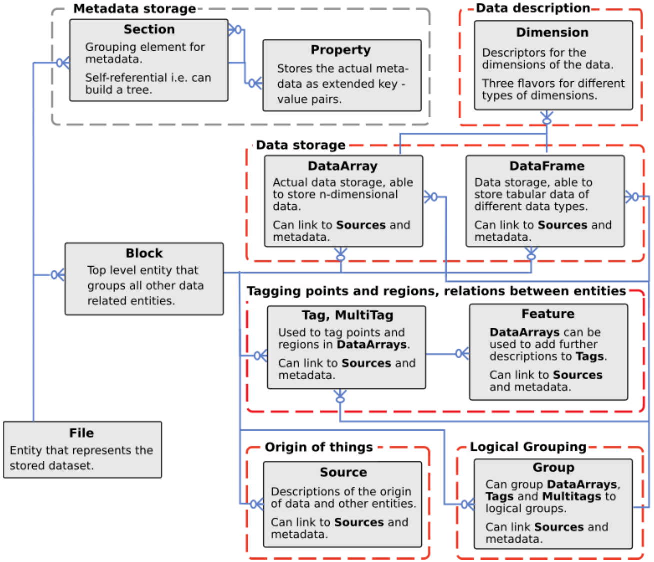
NIX concept

NIX aims at:

1. One data model for storing data and metadata within the same container.
2. Flexible enough to support a high variety of data types.
3. Store n-dimensional data.
4. Allow in-depth annotations.
5. Data entities are self describing.
6. Support standardization in data and metadata storing.

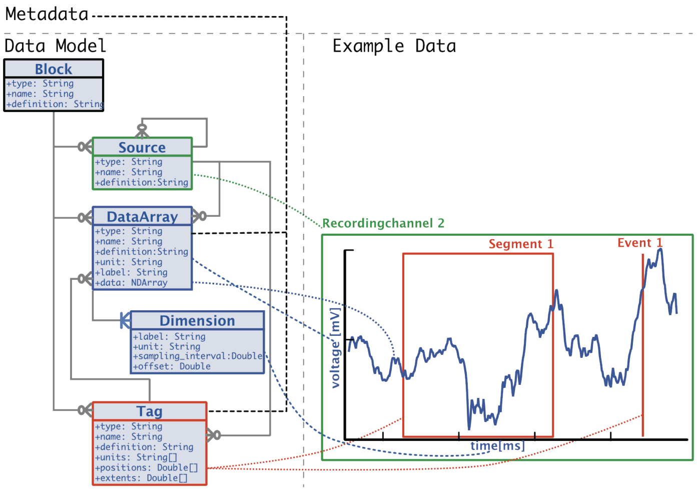
7. Of course, open source.

NIX Data Model



Metadata storage
Section
Grouping element for metadata.
Self-referential i.e. can build a tree.
Property
Stores the actual meta-data as extended key - value pairs.
Data description
Dimension
Descriptors for the dimensions of the data.
Three flavors for different types of dimensions.
Data storage
DataArray
Actual data storage, able to store n-dimensional data.
Can link to **Sources** and metadata.
DataFrame
Data storage, able to store tabular data of different data types.
Can link to **Sources** and metadata.
Block
Top level entity that groups all other data related entities.
Tagging points and regions, relations between entities
Tag, MultiTag
Used to tag points and regions in **DataArrays**.
Can link to **Sources** and metadata.
Feature
DataArrays can be used to add further descriptions to **Tags**.
Can link to **Sources** and metadata.
File
Entity that represents the stored dataset.
Origin of things
Source
Descriptions of the origin of data and other entities.
Can link to **Sources** and metadata.
Logical Grouping
Group
Can group **DataArrays**, **Tags** and **Multitags** to logical groups.
Can link **Sources** and metadata.

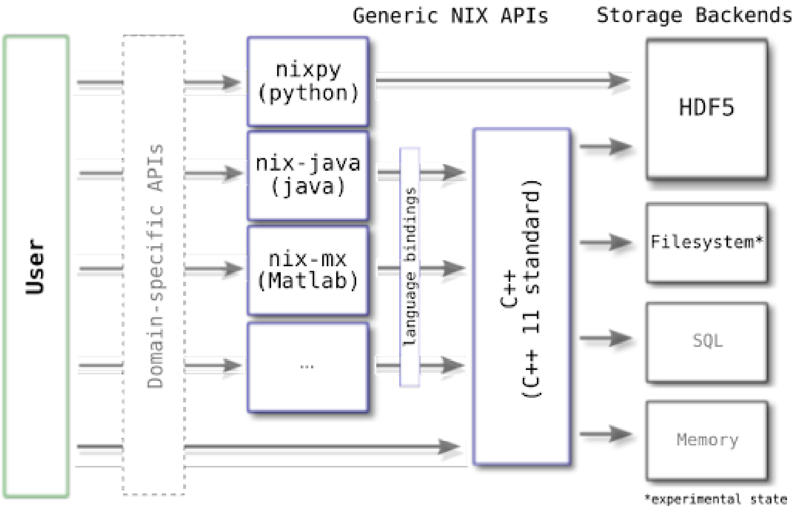
NIX Data Model



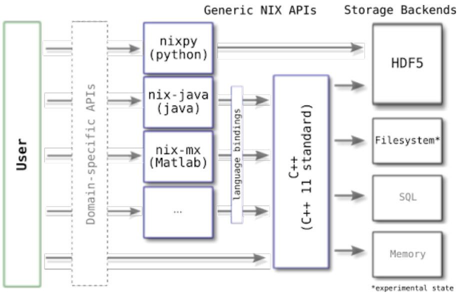
NIX Data Model

- The NIX data model is not a file format.
- It can be implemented in many backends.
- So far, we use the HDF5 file format as storage backend.

NIX Implementations

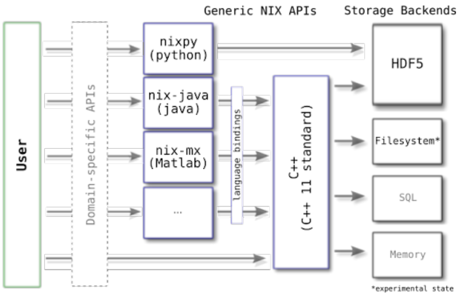


Generic APIs



- There are two native APIs: nixpy (python) and nixio (C++) and two that use language bindings to write NIX files to HDF5 files.
- In principle, we can support several different backends.

Domain specific APIs

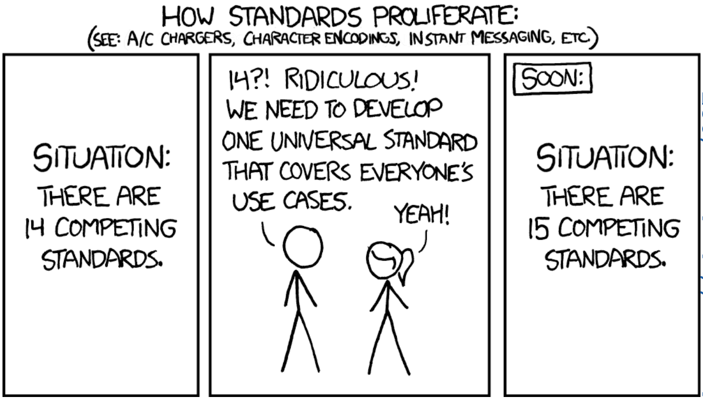


- Domain specific APIs use the generic ones to ease access to data from one domain, e.g. Electrophysiology.
- They provide a way to adapt and standardize storing data.
- This does not hinder to read the data with the generic libraries.

⇒ Neo-nixio stores Neo electrophysiology data using NIX as storage backend.

⇒ Storing to NIX can be embedded into the recording software `http://relacs.sourceforge.net`.

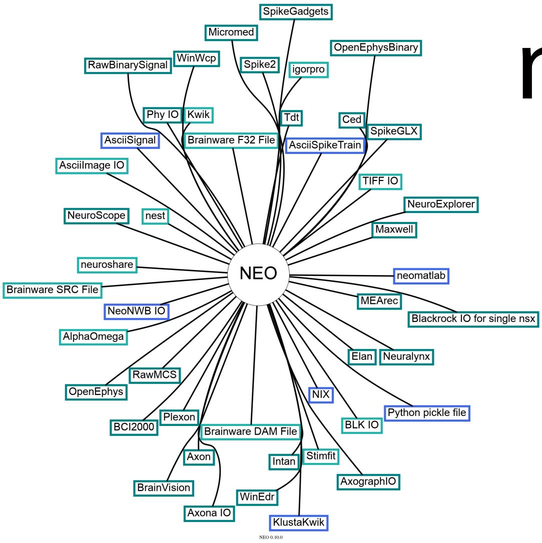
Reproducibility crisis also in the Neurosciences


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

https://xkcd.com/927

...and the other 40+ ephys data standards?

Neo

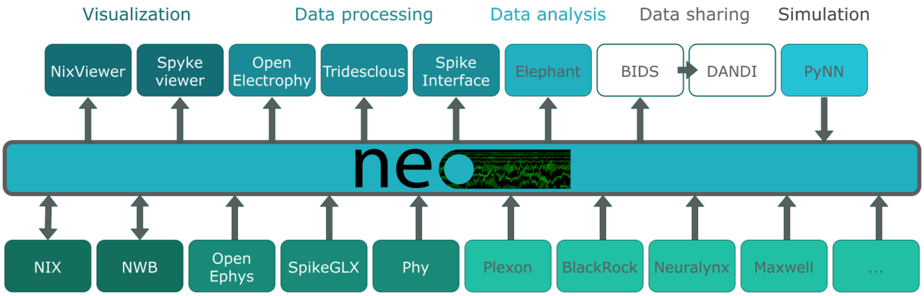


- interfaces to 40+ ephys proprietary & open formats in various versions
- provides a standardized ephys data **representation**, it is not a file format.
- reads all formats, but provides writing capability only for selected open formats.

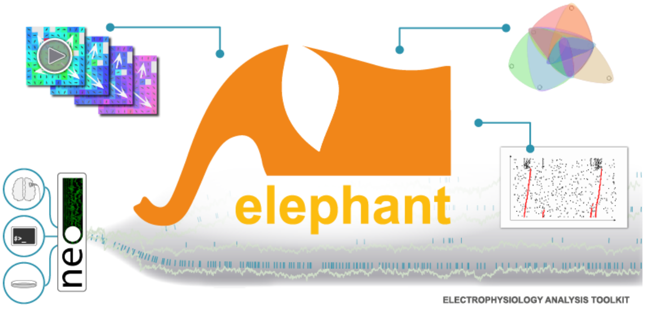
Neo

A basis for

- format conversion
- custom data analysis
- diverse tools dealing with ephys data



Elephant



Electrophysiology Analysis Toolkit

- analysis of brain activity (spike trains, local field potentials,. . .)
- builds on the Neo framework (input, output)
- focus on population measures from experiment and simulation

Viziphant

Quick visualizations

- of Neo representations of electrophysiology data
- of Elephant analysis results

NIX-Neo Workshop

In this workshop we will ...

- introduce the ideas and concepts of the NIX data model.
- show how various data types can be represented with it.
- show how to use and find data in NIX files.
- become familiar with the Neo representation of electrophysiology data.
- learn how to used Elephant and Viziphant for data analysis and visualizations.
- use a lot of hands-on exercises.

NIX-Neo Workshop

Day 1: NIX Basics

- Introduction
- File & data handling
- Tagging
- Annotations

Day 2: Deeper dive

- Adding *Features*
- Finding stuff
- Advanced features

Day 3: Neo&Elephant

- Introduction
- Working with Neo data
- Elephant toolbox

NIX-Neo Workshop - Requirements

- We run the workshop using the python version of the NIX library (called nixio while the project is called nixpy...) and you should have some basic knowledge of the python programming language.
- You need a running python environment ($\geq$ 3.6).
- Installed libraries:
  - jupyter
  - matplotlib
  - nixio
  - neo
  - elephant
  - viziphant
- Alternatively we can use Binder (see the README file in the course repository). `https://gin.g-node.org/INCF-workshop-2021/NIX-Neo-workshop`

NIX-Neo Workshop - Requirements

Installing the requirements locally

```
>pip3 install -r requirements.txt
```

NIX-Neo Workshop - Requirements

Installing the requirements locally

```
>pip3 install -r requirements.txt
```

Please make sure you have the latest `nixio` version.

We will rely on nixio version 1.5.1 In case of doubt run:

```
pip3 install nixio --upgrade
```

NIX-Neo Workshop - Resources

Course material:

- https://gin.g-node.org/INCF-workshop-2021/NIX-Neo-workshop

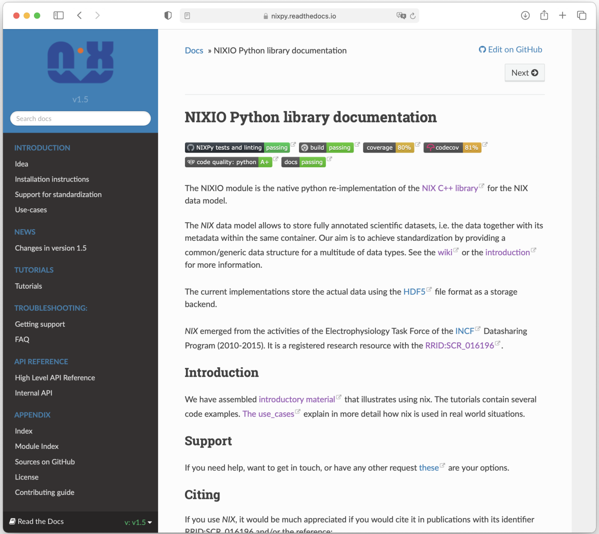
Source code:

- https://github.com/g-node/nix
- https://github.com/g-node/nixpy
- https://github.com/g-node/nix-mx

Documentation & tutorials:

- https://nixpy.readthedocs.io/
- https://nixio.readthedocs.io/
- https://neuralensemble.org/neo/
- https://python-elephant.org

NIXPY documentation



NIXIO Python library documentation

The NIXIO module is the native python re-implementation of the NIX C++ library for the NIX data model.

The *NIX* data model allows to store fully annotated scientific datasets, i.e. the data together with its metadata within the same container. Our aim is to achieve standardization by providing a common/generic data structure for a multitude of data types. See the wiki or the introduction for more information.

The current implementations store the actual data using the HDF5 file format as a storage backend.

NIX emerged from the activities of the Electrophysiology Task Force of the INCF Datasharing Program (2010-2015). It is a registered research resource with the RRID:SCR_016196.

Introduction

We have assembled introductory material that illustrates using nix. The tutorials contain several code examples. The use_cases explain in more detail how nix is used in real world situations.

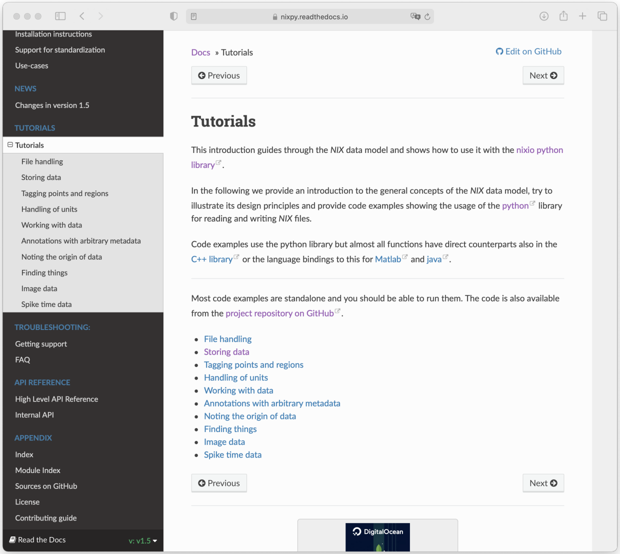
Support

If you need help, want to get in touch, or have any other request these are your options.

Citing

If you use *NIX*, it would be much appreciated if you would cite it in publications with its identifier RRID:SCR_016196 and/or the reference:

NIXPY documentation



Installation instructions
Support for standardization
Use-cases

NEWS
Changes in version 1.5

TUTORIALS
⊟ Tutorials
File handling
Storing data
Tagging points and regions
Handling of units
Working with data
Annotations with arbitrary metadata
Noting the origin of data
Finding things
Image data
Spike time data

TROUBLESHOOTING:
Getting support
FAQ

API REFERENCE
High Level API Reference
Internal API

APPENDIX
Index
Module Index
Sources on GitHub
License
Contributing guide

Read the Docs v: v1.5

Docs » Tutorials — Edit on GitHub

Previous | Next

Tutorials

This introduction guides through the *NIX* data model and shows how to use it with the nixio python library.

In the following we provide an introduction to the general concepts of the *NIX* data model, try to illustrate its design principles and provide code examples showing the usage of the python library for reading and writing *NIX* files.

Code examples use the python library but almost all functions have direct counterparts also in the C++ library or the language bindings to this for Matlab and java.

Most code examples are standalone and you should be able to run them. The code is also available from the project repository on GitHub.

- File handling
- Storing data
- Tagging points and regions
- Handling of units
- Working with data
- Annotations with arbitrary metadata
- Noting the origin of data
- Finding things
- Image data
- Spike time data

Previous | Next