D5.7.1 Representations of optical imaging data in the Neo data object model

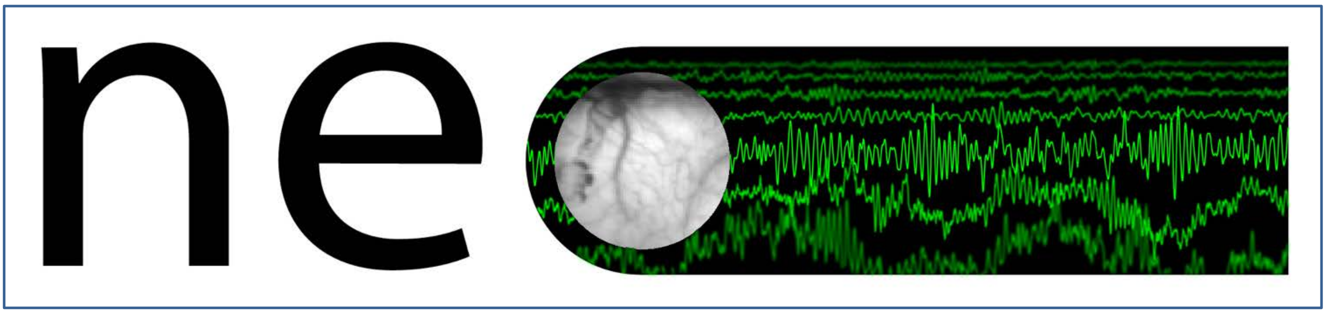

Figure 1: Modified Neo logo incorporating calcium imaging data produced by the WaveScalES collaboration within the Human Brain Project

Project Number:	785907	Project Title:	Human Brain Project SGA2

Document Title:	Representations of optical imaging data in the Neo data object model
Document Filename:	D5.7.1 (D34.1, D78) SGA2 M18 ACCEPTED 200730.docx
Deliverable Number:	SGA2 D5.7.1 (D34.1, D78)
Deliverable Type:	Other *(software release)*
Work Packages:	WP5.7
Dissemination Level:	PU = Public
Planned Delivery Date:	SGA2 M18 / 30 Sep 2019
Actual Delivery Date:	SGA2 M19 / 3 Oct 2019; Accepted: 30 Jul 2020
Author(s):	Andrew DAVISON, CNRS (P10)
Compiled by:	Andrew DAVISON, CNRS (P10)
Contributor(s):	Andrew DAVISON, CNRS (P10) conception, design, testing Corentin FRAGNAUD, CNRS (P10) design, implementation, testing
SciTechCoord Review:	Martin TELEFONT, EPFL (P1)
Editorial Review:	Annemieke MICHELS, EPFL (P1)
Description in GA:	This is a release of the Neo library that contains support for the representations of optical imaging data. This feature will facilitate the analysis of activity data obtained from optical imaging experiments using tools built on the Neo data model. Successful representation of a dataset in the Neo object model, and the subsequent application of an analysis function of the Elephant framework on that data object, will be shown (T5.7.1-2)
Abstract:	We report on a new release of the Neo library that contains support for optical imaging data. This feature will facilitate the analysis of activity data obtained from optical imaging experiments using tools built on the Neo data model. This report shows the successful representation of a dataset in the Neo object model, and the subsequent application of an analysis function of the Elephant framework on that data object.
Keywords:	optical imaging, fluorescence microscopy, calcium imaging, two-photon imaging, file formats, Python, workflows
Target Users/Readers:	neuroscientists

Table of Contents

Table of Figures

1. Introduction

While recording of electrical signals through intracellular and extracellular electrodes was for many years the dominant method of investigating the physiology of neurons and neuronal circuits, optical signals recorded with microscope-attached video cameras, including intrinsic signal imaging and fluorescence imaging using calcium- or voltage-sensitive dyes, is now very widely used.

As with electrophysiology, extensive, multi-stage analysis of the signals obtained through optophysiology is typically used to extract signal from noise and to isolate the components of interest. The Neo Python library (Garcia *et al*., 2014) was conceived and developed to simplify and accelerate development of such analyses for electrophysiology data, by providing (i) a common in-memory data model and (ii) a set of input/output (IO) modules for the plethora of file formats, both proprietary and open, used for electrophysiological recordings. While Python libraries exist for reading specific file formats (e.g. pynwb) or for handling a specific use case such as spike sorting (e.g. Spyking Circus), Neo is the only actively-maintained, general purpose, file-format-independent library for handling electrophysiology data in Python.

The goal of the work reported here was to add support for optical imaging data to the Neo data object model, with the following requirements: (1) the user should be able to load image sequence data, stored either as a text file or as a folder of TIFF images, into the Python interpreter, such that the data are represented in a structure behaving like a 3D NumPy array but with additional metadata representing pixel size and the frame/sampling rate; (2) the user should be able to extract time series signals from regions of interest within the image sequence, to store these signals as Neo AnalogSignal objects, and to store a representation of the relationship between the original image data and the derived time series within the Neo container structure.

This has been done, and a new version of Neo including these extensions has been released (v0.8.0). This document summarises the new classes added to Neo, and gives an example of how these classes can be used together with the Elephant data analysis toolkit to perform an analysis of calcium imaging data. More details are available online in the release notes.

2. Implementation

This new release of Neo extends the core data structures with a new data object, ImageSequence, and a metadata/grouping object, RegionOfInterest. The former is represented by a single class, the latter by several subclasses for different shaped regions, e.g. CircularRegionOfInterest, RectangularRegionOfInterest.

As is done for the other Neo data objects, ImageSequence subclasses the NumPy nD-array class (via the Quantity array class from the quantities package). This means it can be used in the same way as a plain NumPy array in downstream analysis code, but also contains essential metadata about the data (for example the pixel size and the frame/sampling rate) and can also contain additional, user-defined metadata.

RegionOfInterest objects contain information about the shape of the region of interest (e.g. for a circular region, the image coordinates of the centre of the region, and its radius). A RegionOfInterest object can be applied to an image sequence (via the signal_from_region method of the ImageSequence object) to generate a time series (the average image brightness within the Region of Interest (ROI) as a function of time), which is returned as a Neo AnalogSignal object. The RegionOfInterest object also serves to store the link between the AnalogSignal and the ImageSequence from which it was derived, in this case.

ImageSequence and RegionOfInterest objects are integrated into the Neo container hierarchy similarly to other Neo objects: ImageSequences are contained within Segments (in the attribute imagesequences), and RegionOfInterests are contained within Blocks.

In addition to the core data structures, Neo provides a collection of IO modules for reading data in different file formats. For image sequence files, two new IO modules have been developed for this release: AsciiImageIO and TiffIO. IO modules for other file formats are under development.

3. Example usage

To illustrate how Neo's optical imaging data representations can be used in a real-world data analysis workflow, and to validate the implementation, we present here a demonstration of loading wide-field fluorescence microscopy data recorded from a transgenic mouse model expressing a calcium indicator (GCaMP6f) in excitatory neurons (Celotto *et al.*, 2018), and calculating the power spectrum from a region of interest.

```
import os
import matplotlib.pyplot as plt
import numpy as np
from quantities import Hz, mm, dimensionless
from neo.core import CircularRegionOfInterest
from neo.io import TiffIO
import elephant as el

data_path = os.path.expanduser("~/Data/WaveScalES/LENS/170110_mouse2_deep/t1")

# loading data
data = TiffIO(data_path, units=dimensionless, sampling_rate=25 * Hz,
              spatial_scale=0.05 * mm).read()
images = data[0].segments[0].imagesequences[0]
images /= images.max()
plt.subplot(2, 2, 1)
plt.imshow(images[100], cmap='gray')
plt.title("Original image (frame 100)")

# preprocessing
background = np.mean(images, axis=0)
preprocessed_images = images - background
plt.subplot(2, 2, 2)
plt.imshow(preprocessed_images[100], cmap='gray')
plt.title("Subtracted background (frame 100)")

# defining ROI and extracting signal
roi = CircularRegionOfInterest(x=50, y=50, radius=10)
circle = plt.Circle(roi.centre, roi.radius, color='b', fill=False)
ax = plt.gca()
ax.add_artist(circle)

central_signal = preprocessed_images.signal_from_region(roi)[0]
plt.subplot(2, 2, 3)
plt.plot(central_signal.times, central_signal, lw=0.8)
plt.title("Mean signal from ROI")
plt.xlabel("Time [s]")

# calculating power spectrum
freqs, psd = el.spectral.welch_psd(central_signal,
                                   fs=central_signal.sampling_rate,
                                   freq_res=0.1 * Hz,
                                   overlap=0.8)
plt.subplot(2, 2, 4)
plt.plot(freqs, np.mean(psd, axis=0), lw=0.8)
plt.title("Average power spectrum")
plt.xlabel("frequency [Hz]")
plt.ylabel("Fourier signal")

plt.tight_layout()
plt.show()  # see Figure 2
```

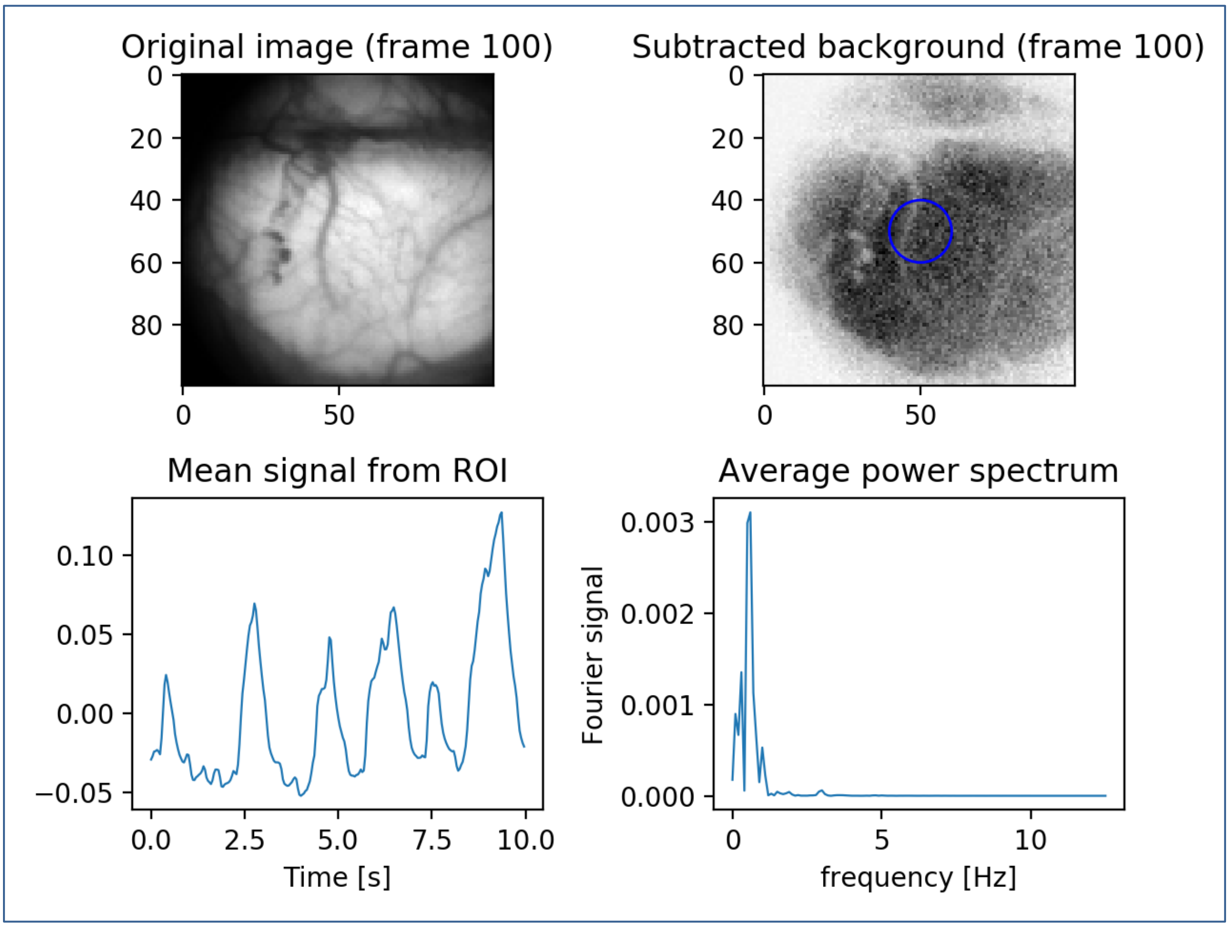

Figure 2: Power spectrum of region of interest generated by Neo-Elephant pipeline from calcium imaging data

4. Future work

In future work, we will develop IO modules for other optical imaging file formats (collaboration with Thomas DENEUX, CNRS, Gif-sur-Yvette, France) according to user demand. In addition we will add support for writing image sequence data to standard, open file formats (in particular NIX and NWB). Neo as a whole is moving close to the release of version 1.0, at which point the API will be frozen to ensure backwards compatibility. Before this, a refactoring of the Neo grouping structures to make them easier to use and understand, together with a major improvement to the documentation, is planned.

5. References

Marco Celotto, Chiara De Luca, Paolo Muratore, Francesco Resta, Anna Letizia Allegra Mascaro, Francesco Saverio Pavone, Giulia De Bonis, Pier Stanislao Paolucci (2018) Analysis and Model of Cortical Slow Waves Acquired with Optical Techniques. arXiv:1811.11687 [q-bio.NC]

Garcia S., Guarino D., Jaillet F., Jennings T.R., Pröpper R., Rautenberg P.L., Rodgers C., Sobolev A., Wachtler T., Yger P. and Davison A.P. (2014) Neo: an object model for handling electrophysiology data in multiple formats. *Frontiers in Neuroinformatics* 8:10 doi:10.3389/fninf.2014.00010