# DataAcquisition

## A user guide

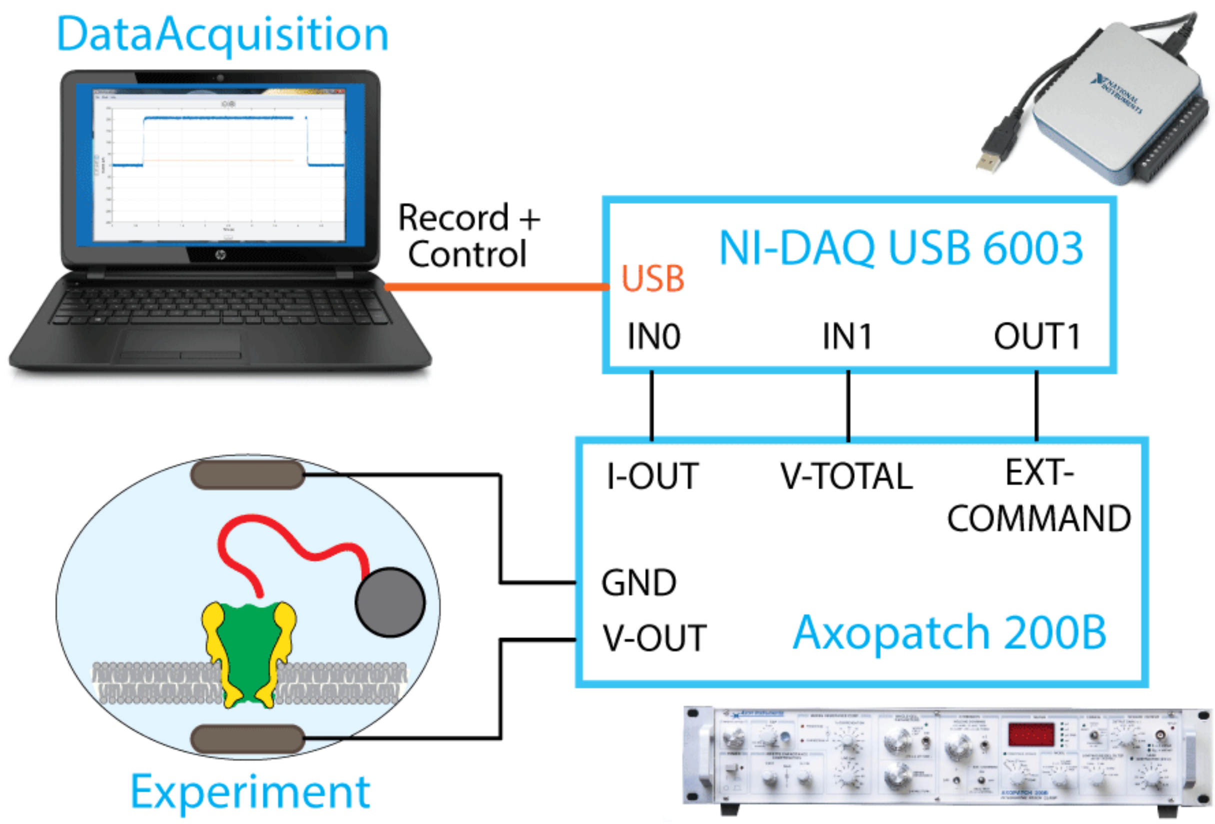
### Stephen Fleming

---

# Contents

## What is DataAcquisition?

DataAcquisition is an open-source graphical user interface for viewing and recording data from a National Instruments DAQ in real time. The program runs in MATLAB. Specifically developed for patch-clamp electrophysiology recordings, DataAcquisition together with the NI DAQ USB-6003 provides an alternative to Axon's pCLAMP software together with the DigiData analog-to-digital converter.



## Why

Other software platforms for recording data from a DAQ are usually either expensive or complicated. This is a minimalist approach with a simple interface for viewing and saving data. There are a few features tailored to electrophysiology recordings, such as noise plots, current-versus-voltage curves, and capacitive current measurements.

This software allows for data to be acquired directly into Matlab from a National Instruments DAQ USB-6003, which is relatively inexpensive, but still produces high-quality, low-noise recordings.

This program was developed for and field tested by Harvard's Fall 2016 Freshman Seminar lab course on nanopore DNA sequencing.

# Usage

Open Matlab (tested for Matlab 2016 and 2017 releases). Navigate to the folder on your computer where DataAcquisition is stored (the folder that contains this file).

In the Matlab command line, type:

```
>> DataAcquisition;
```

or

```
>> d = DataAcquisition();
```

Optionally in the function call, you can input Channels, Alphas, OutputAlpha, and/or SampleFrequency as a list of name, value pairs:

`'Channels'` should be paired with a vector (maximum of four elements) containing the integers 0 through 3 that specifies which channels are inputs. e.g. [0, 1] Note: specify the scalings for all inputs and outputs if you specify any.

`'Alphas'` should be paired with a vector of scale factors to apply to analog inputs (to convert measured values to either pA or mV). Note: you must specify the scale factors for all inputs if you specify any.

`'OutputAlpha'` should be paired with a numeric scale factor to be applied to the analog output (to convert values in mV to voltage output in the range [-10,10] Volts).

`'SampleFrequency'` should be paired with a numeric value that specifies the frequency at which data are sampled. Default is 25kHz. Note: due to hardware limitations, the upper limit for the NI DAQ USB-6003 is 100kHz divided by the number of input channels.

These values need not be set when calling the function. They can be set later from within the graphical user interface using File > Configure DAQ.

## Note on file format

DataAcquisition by default saves data in the Dataacquisition Binary Format, a DBF file. This can be opened and used for analysis in Matlab using Tamas Szalay's PoreView package with DBF file handling, found here.

DBF files can be converted to other file formats for use in other data analysis pipelines. Supported file types are CSV and HDF4. In the DataAcquisition graphical interface, navigate to File > Convert Data File. After selecting a DBF file for conversion, select either CSV or HDF for the desired output file format. HDF files are better, and the same size as DBF files. CSV files are easy to open in Excel or Notepad, but the file size is much much larger.

To get started creating your own analysis pipeline in Matlab, try using Matlab's hdftool utility to import the data from an HDF file. In Matlab's command line, type:

```
>> hdftool('filename');
```

# Example Usage

You should have your DAQ plugged into the computer and the relevant National Instruments software installed (including NIDAQmx, which installs automatically). The name of the device should be 'Dev1', which is the default.

To launch DataAcquisition, open MATLAB, navigate to the folder containing DataAcquisition.m, and type the following in the MATLAB command line:

```
>> d = DataAcquisition();
```

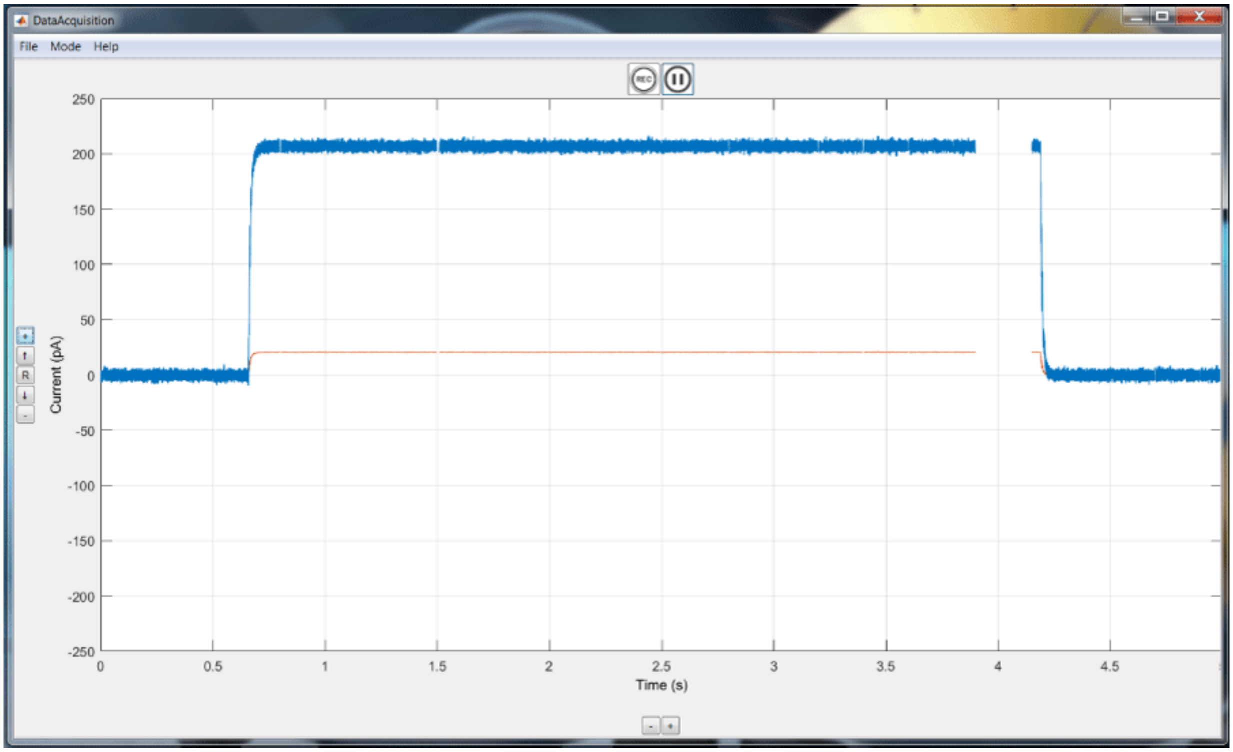
The command above uses default values. Or you can specify everything:

```
>> d = DataAcquisition('Channels',[0,1], 'Alphas',[100,100], 'OutputAlpha',10,
'SampleFrequency',25000);
```

Note that these parameters can also be changed using the graphical interface, once it has launched (File > Configure DAQ).

## Normal acquisition mode

View and record current and voltage as a function of time. Access this mode via Mode > Normal acquisition.




View data in real time, as you would on an oscilloscope, using the Play/Pause button.

The blue line is current, in picoamperes [pA], and the red line is voltage, in millivolts [mV]. Although the y-axis label is Current (pA), the voltage signal is numerically on the same scale, but is in mV.

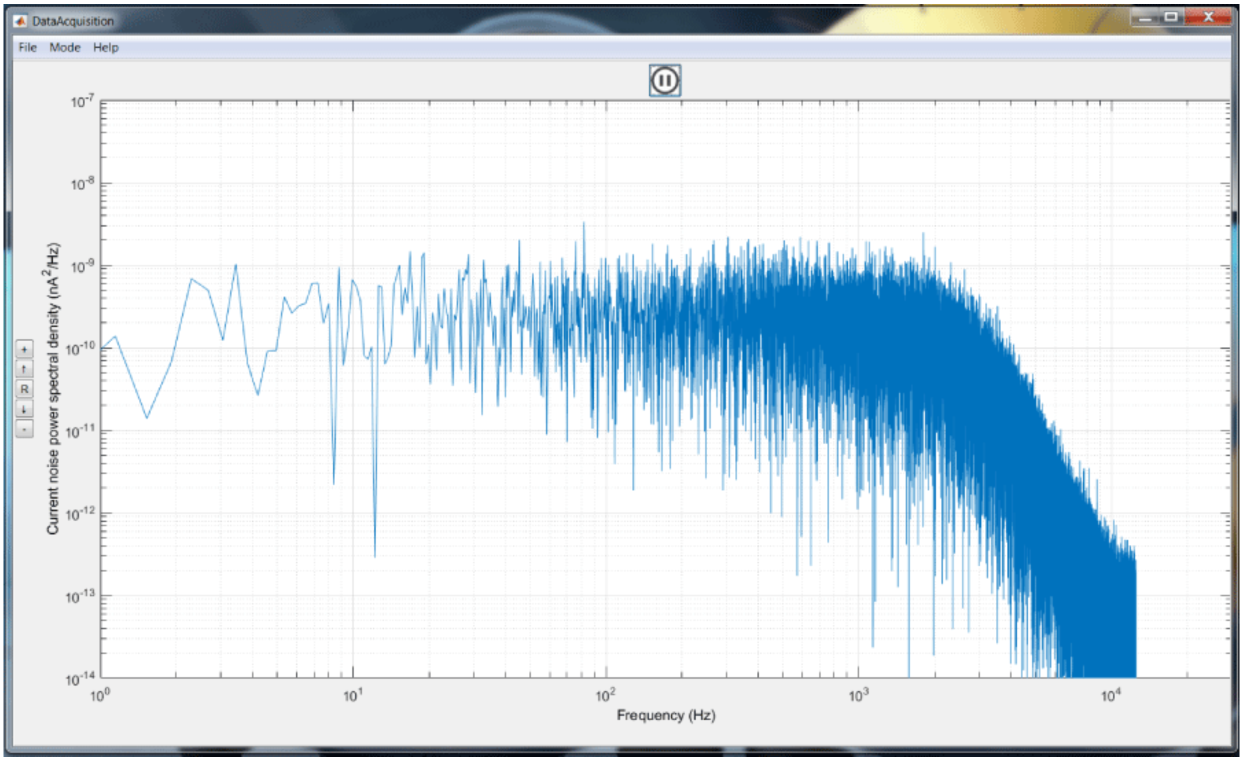
Buttons on the left and bottom of the window allow you to zoom in and out. The 'R' button at the left returns the y-axis to its original scaling.

Record and view data using the REC button. The REC button will turn red when recording is active. Recording can be stopped by either pressing the red REC button again, or by pressing the Play/Pause button.

Each time you record, a new data file will automatically be created in the directory you have specified. You can change this directory via File > Choose Save Directory. File names are generated automatically, with a date prefix and increasing numeric order.

## Noise mode

View a plot of the noise power spectral density, updated in real time. Access this mode via Mode > Noise plot.

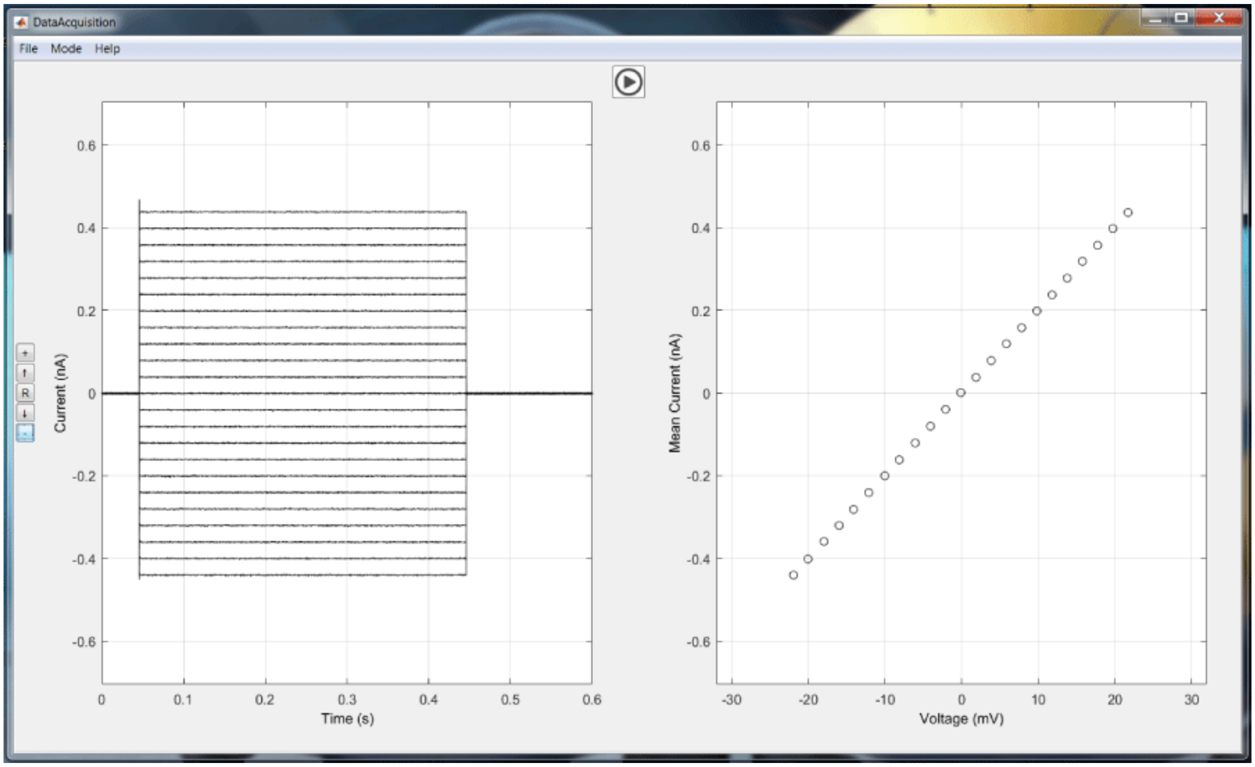



Current noise power spectral density (in square nanoamperes per Hertz) is plotted on a logarithmic scale as a function of frequency on a logarithmic scale, from 1Hz up to 30kHz. The plot is automatically updated every 2 seconds.

This mode can be used for diagnostics. A spike near 50-60Hz indicates that there is power line noise, and the experiment is insufficiently shielded from the environment. Noise that increases at low frequencies may indicate a leakage or other problem. A falloff in noise at high frequencies indicates the use of a hardware filter in the experimental setup (here part of the current amplifier).

# IV curve mode

Record current at various voltages by applying a set of voltages for specified times. Access this mode via Mode > IV curve.




The DAQ's analog output AO0 must be connected to an external command input of the current amplifier for this mode to work. (Also check to be sure the scale factor for the output is set correctly. This will depend on your hardware.)

Pressing Play brings up a dialog box that can be used to programmatically sweep through a series of applied voltages. Every time an IV curve is run, the data are automatically saved as a DBF file.

In the left panel, current versus time is plotted at each voltage, with sweeps overlaid on top of each other.

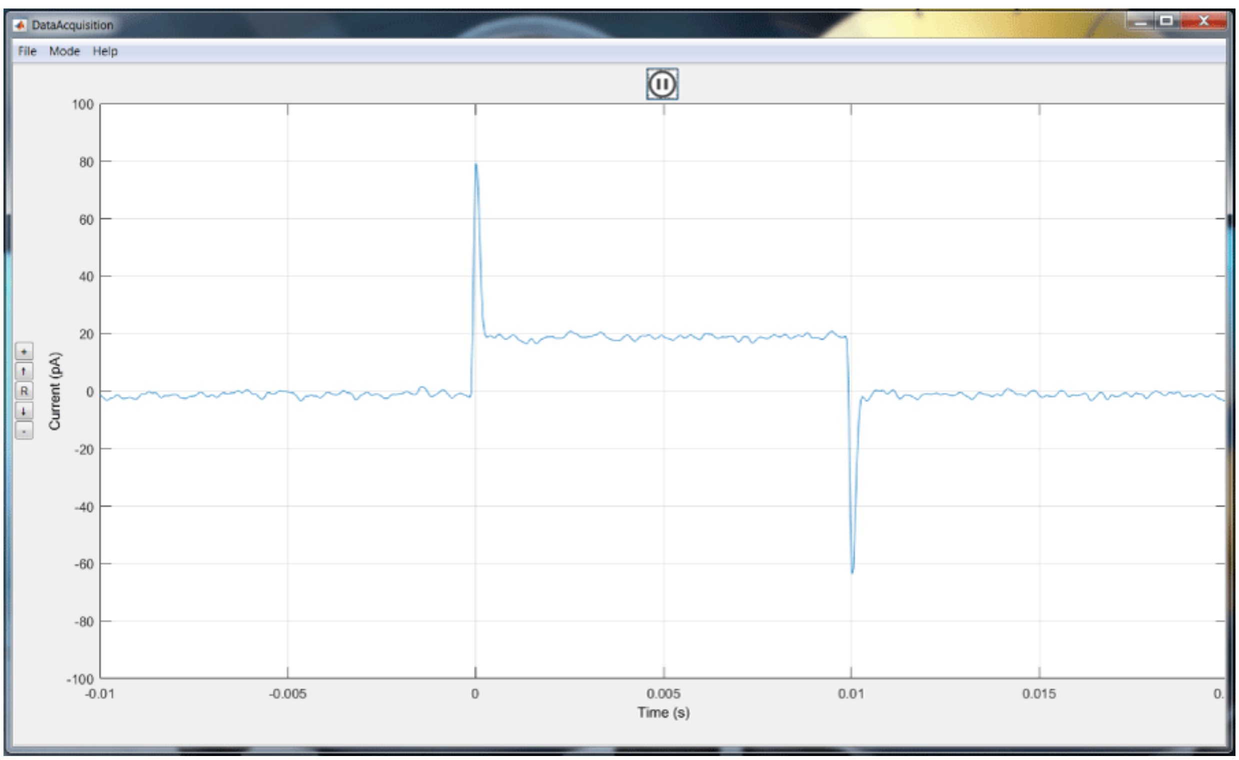
In the right panel, mean current is plotted as a function of applied voltage. The linear relationship is characteristic of a resistor.

## Capacitance compensation mode

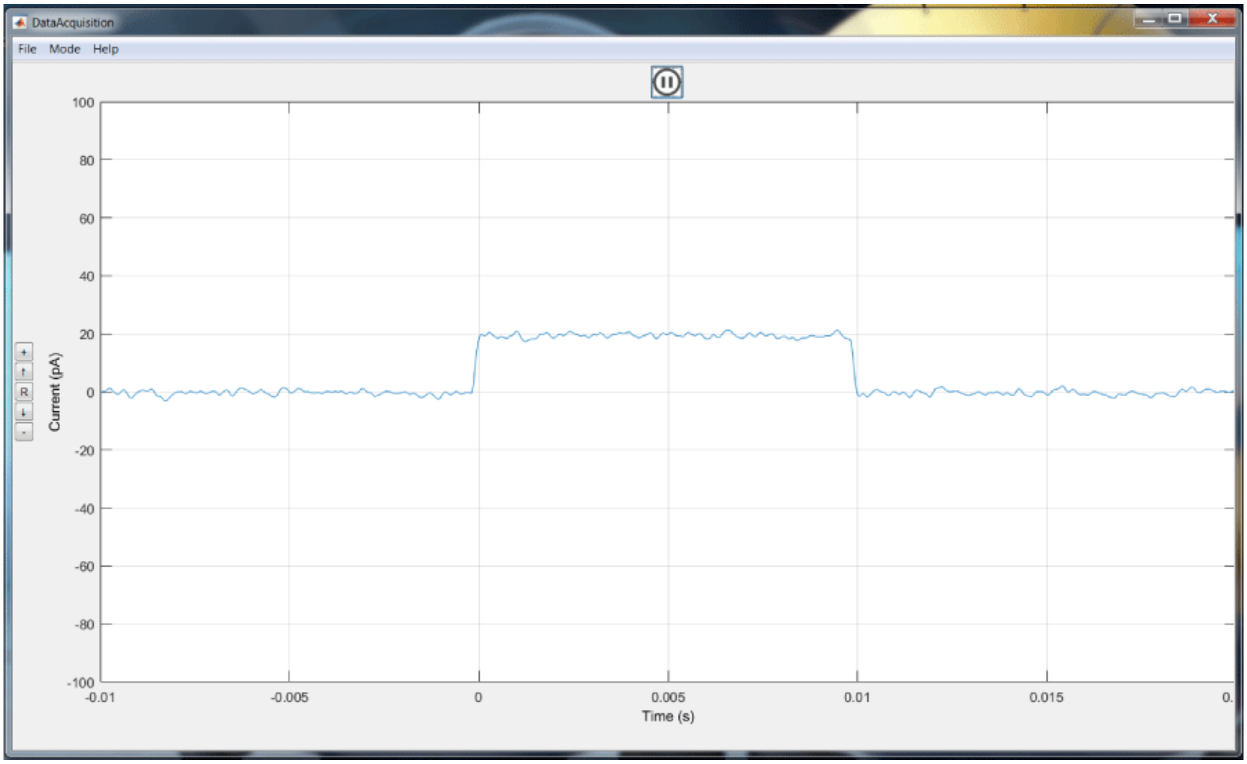
Display the current-versus-time response to a 10ms square-wave of 5mV, in order to look at the capacitive current spikes. Access this mode via Mode > Membrane seal test. "Membrane seal test" refers to similar functionality in other electrophysiology recording software.

Capacitive current spikes can be compensated for on most current amplifiers used for electrophysiology by means of a "capacitance compensation" knob. Using capacitance compensation mode makes this process easy by immediately showing the results of attempts at compensation.

Before compensation (capacitive current spikes visible):



After compensation using the "capacitance compensation" knob on the current amplifier (capacitive current spikes gone):




## Recommendations on Experimental Setup

The ideal experimental setup is to have BNC cables attached to the current amplifier, and then to have BNC splitters attached to those cables, which split the BNC into signal and ground. These wires can be plugged into the DAQ separately (e.g. 0+ for current signal, and 0- for current ground), implementing differential sensing mode. Connect the output of the current amplifier to DAQ channel 0, connect the membrane voltage output of the amplifier to DAQ channel 1, and connect the external command input on the current amplifier to DAQ AO0 (signal) and AO_ground (ground).

## Who

Stephen Fleming, PhD candidate at the Golovchenko Lab in the physics department (part of the Harvard Nanopore Group) at Harvard University, 2016.