OSSI proposal: Kilosort by Marius Pachitariu

Neuropixels electrodes, developed at Janelia, have single-handedly transformed the field of electrophysiology. Almost overnight, our community went from mostly using single-channel electrodes to mostly using 384-channel Neuropixel electrodes. Instead of painstakingly recording one neuron at a time like before, researchers can now record hundreds or thousands of neurons simultaneously. We are already witnessing the acceleration in research provided by Neuropixels, with Janelia at the forefront of its science as well.

Software overview

Kilosort [paper][code] is the main software used to analyze data from Neuropixels probes. Its main task is to take hundreds of GB of electrophysiology data and output a set of neurons with their spike times. The pipeline includes several distinct algorithmic steps, such as data registration, clustering, spike extraction and quality control.

I have developed, maintained and updated Kilosort alongside the Neuropixels probes, in close collaborations with the people developing and testing the probes: at UCL when I was a postdoc, then in collaborations with Janelia / Allen / Moser lab at Kavli / International Brain Lab at UCL and Champalimaud. Other updates to Kilosort have stemmed from collaborations outside the Neuropixels consortium (Deisseroth lab at Stanford, Sternson lab at Janelia). There have been four major versions of Kilosort (1 / 2 / 2.5 / 3) which I developed at intervals of 1-2 years.

The hardware design of the probes themselves has been strongly influenced by spike sorting results from Kilosort. For example, Neuropixels 2.0 has vertically-aligned columns of channels rather than the checkerboard design of 1.0, because we realized the vertical columns are much better for spike sorting. In turn, Kilosort 2.5 was optimized to perform well on such geometries. There is thus already a strong synergy between the software and hardware side of this highly successful Janelia effort. However, while the hardware development has been supported by multiple multi-million dollar grants (from HHMI, from Welcome, from Allen, from Champalimaud, etc), the software side has received very little support. So, while teams of dozens of people work to develop, test and maintain the hardware, it is primarily just me working to develop and maintain Kilosort.

Significance

We estimate that Kilosort is used by more than 500 labs. This includes most of the labs using Neuropixels probes, as well as labs using other types of probes or even tetrodes. 85% of respondents use Kilosort, as per a poll in the Neuropixels slack. The original Kilosort paper is still just a preprint, but it has received almost 500 citations, compared to the 1,000 for the Neuropixels paper in Nature. We don’t have a way to track downloads or unique installations, but the repo gets 500-1000 unique visitors / 2 weeks.

Request

I have been the primary developer for Kilosort since the start. The main extra help I got was last year when 3 programmers helped me translate the code from Matlab to Python. These programmers finished the conversion, but we still haven't found the time to fully test it, so the Python codebase has remained somewhat abandoned (not quite, a team at IBL was able to adapt it for their own consortium and we have some other early adopters too).

The sheer scope of maintaining the software is well above the free time I have, on-the-side from running a full neuroscience lab myself. For example, there are now 157 open issues on github, which I have not had time to respond to since the beginning of summer due to other commitments such as running the Neuromatch summer schools. Furthermore, whenever I do find some time I prefer working on algorithmic improvements to the software. Therefore, I have been able to release a new Matlab version of Kilosort this year (3.0), but the Python conversion is now further lagging behind.

We are requesting support for (1) maintenance; (2) a full Python release of the latest Matlab version with some workshops to help the community transition; (3) custom pipelines, developed closely with Janelia labs and then released broadly to the community. I am especially excited about this last one, because it's a great way to find existing weaknesses of the algorithm and try to improve on them. The modular nature of the pipeline allows for many possible extensions.

I am hoping to find a neuroscientist-coder, or coder who wants to learn some neuroscience, and work with them to pass on some of the specialized knowledge needed to maintain the software, as well as to communicate effectively with the labs that need help or need new features. I believe this software project would be a solid pillar to support Janelia's neuroengineering and cognitive neuroscience efforts. Like all collaborations between labs and tool developers at Janelia, the benefits will be directly passed on to the entire community. Since we already have a large user base, the benefits of this work will be felt nearly instantly.