**Installation instructions**

Before you start:

1. Make sure your computer is up-to-date, and has compatible (not necessarily the newest) CUDA drivers for your Nvidia GPU. If you don’t have an Nvidia GPU, you’ll have to buy one, because multiple Acquipix functions, as well as the external libraries it uses (Kilosort, Psychtoolbox) make use of CUDA-specific GPU-accelerated computing.
2. You will want a fast SSD for data buffering for various pre-processing operations. You can run everything without an SSD, but it will be noticeably slower.

Install the external libraries for the recording computer:

1. If you want to run visual experiments on this computer, download Psychtoolbox and follow the installation instructions: <http://psychtoolbox.org/download>
2. If you want to record Neuropixels on this computer, install SpikeGLX: <https://github.com/billkarsh/SpikeGLX>
3. If you want to perform pupil-tracking specifically, or generally want to have accurate & automatic synchronization between your recorded videos, stimulation and Neuropixels data, you can you install this Acquipix-independent module on the computer where you’re recording the video’s: <https://github.com/JorritMontijn/EyeTracker>

Install the external libraries for preprocessing your data:

1. Download the Kilosort, Npy-matlab and spikes repositories and follow their installation instructions: <https://github.com/MouseLand/Kilosort/>, <https://github.com/kwikteam/npy-matlab> and <https://github.com/cortex-lab/spikes>. To compile Kilosort’s GPU functions, you’ll first need to install a compiler. You can for example use visual studio, but note that only specific versions will work in combination with specific versions of matlab: <https://visualstudio.microsoft.com/vs/older-downloads/>. In all cases, make sure you get the ***Community*** version, and make sure you also install the ***C++ compiler***.
2. Compiling Kilosort’s CUDA functions with ***Visual Studio Community 2015*** with Update 3 seems to work with Matlab R2019b, but the version compatibility seems somewhat arcane and random. If your matlab is R2019b or later, try VSC2015 first. If it doesn’t work, you can try a different version (i.e., VSC2013).
3. Rename and edit the config and chanmap files to match your preferred settings and copy them to a folder outside the git repositories if you want to make sure they don’t get overwritten accidentally

Install Acquipix and its dependencies:

1. Download the Acquipix repository at <https://github.com/JorritMontijn/Acquipix>
2. Download the GeneralAnalysis and MNCP repositories from <https://github.com/JorritMontijn/>
3. If you wish to automatically compute a responsiveness metric for your putative single cells, download <https://github.com/JorritMontijn/ZETA>

To run an experiment:

1. Start SpikeGLX and begin a new acquisition
2. Start “RunExperiment” in MATLAB
3. Select your stimulation and parameters & start the experiment (spikeglx will start recording automatically)

Optional steps for pre-processing:

1. Pre-process your eye-tracking data
2. Pre-process your probe coordinates

To pre-process your spiking data in five mouse clicks:

1. Start “RunRecordingProcessor” in MATLAB and compile your data library
2. Select the recording you wish to preprocess
3. Cluster your data by clicking the button in the GUI
4. Combine data from multiple sources by clicking another button using the GUI
5. Optionally check the results, make some changes if necessary & then export with click #5

**User guide**

**Background**

Multi-stream data recorded while performing a visual or optogenetic experiment, can be difficult and laborious to process. If your data includes stimulation, multi-channel spiking, LFPs, running speed, and pupil tracking, you will likely already have to deal with 5 independent data streams that need to be synchronized (fig. 1). Moreover, keeping track of all these different files can be a pain. That’s why we created the Acquipix repository, which does all the synchronizing and synthesizing for you. All our code is modular, open source and fully customizable, but we made sure you can also run everything out-of-the-box using only graphical user interfaces without having to write a single line of code.

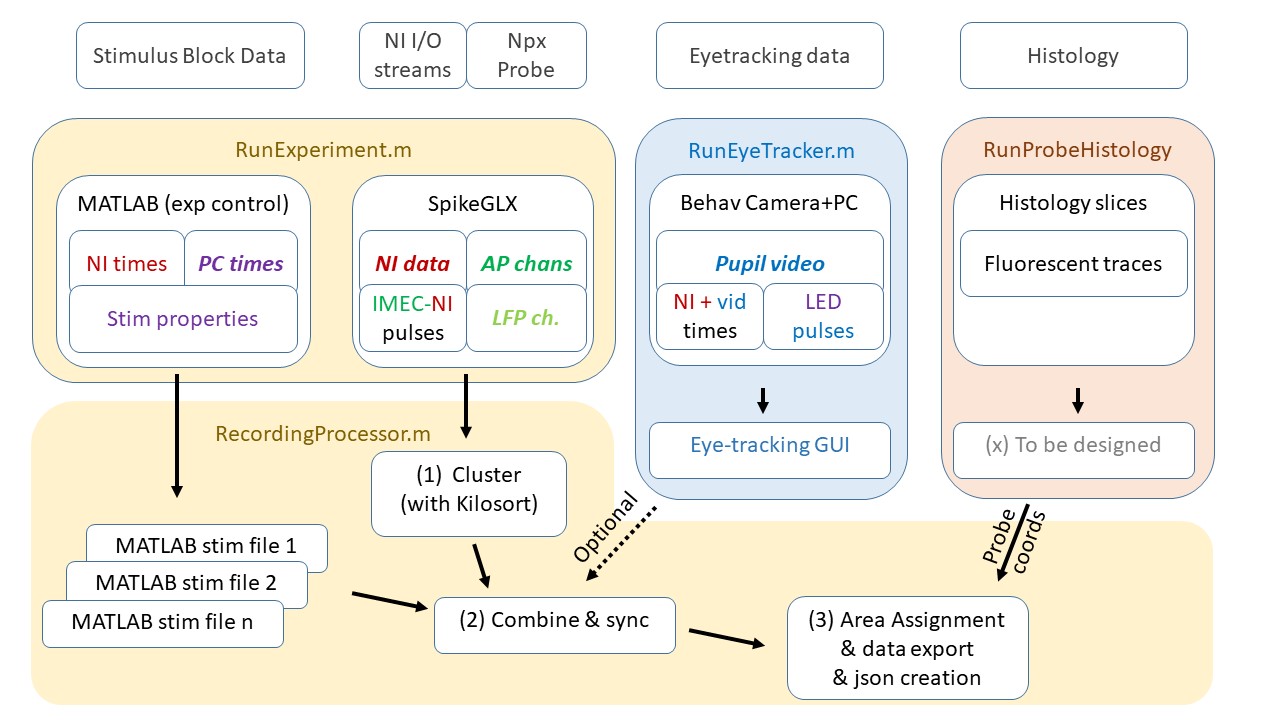


Figure 1. Technical data flowchart showing a single recording. First column from left (*MATLAB (exp control)*): each recording consists of one or more stimulus blocks that contain onset/offset times and stimulus data that are generated using RunExperiment.m and its dependency functions. The stimulation scripts automatically record the time stamps of the national instruments (NI) I/O card and internal stimulus times (PC times) generated by Psychtoolbox. Second column (*SpikeGLX*): although the NI I/O streams and IMEC neuropixels (Npx) data may appear to be monolithic, they are entirely separate data streams with independent clock times. SpikeGLX produces data streams at three different sampling frequencies: the NI I/O data, the AP channels, and the downsampled LFP channels. To synchronize these data streams, you need common pulses; i.e., connect the IMEC pulse generator to an NI I/O stream. Third column (*RunEyeTracker*): optionally, you can use the eye-tracker program to perform online eye-tracking and automatically synchronize your pupil tracking movie to the rest of your data. While recording, the eye tracker program periodically queries SpikeGLX for NI data stream time stamps and saves a log file containing synchronous video time stamps and NI time stamps. Moreover, you can define an ROI for luminance-based high-resolution synchronization (LED pulses). Last column (*RunProbeHistology*): by registering your histology slices to the Allen Brain Atlas, you can extract your probe’s location during recording. The RecordingProcessor allows you to easily cluster your spikes with Kilosort (1), combine and synchronize data from multiple sources (2), and automatically extract which brain area each cell was located, using the ABA API (3).

**Running an experiment**

To run an experiment, run matlab and execute “RunExperiment”

**Preprocessing your data**

To start preprocessing, run matlab and execute “RecordingProcessor”

**Troubleshooting**

Question (“actually, it’s more of a comment”): *It doesn’t work*

Answer: Restart your PC

Q: *I downloaded everything, but it says files are missing*

A: Double check you have added all folders to the path in Matlab, you have the required Matlab toolboxes installed (Curve Fitting, Parallel Computing), and you’re using a supported matlab version: R2019b is tested and works; anything earlier than R2016b will fail for sure; other versions might work. You may also need additional toolboxes, such as Image Processing and Acquisition to perform eye-tracking. If it’s still not working after you’ve tried the above, google the filename and reinstall its source repository. If it still fails, create a report here: <https://github.com/JorritMontijn/Acquipix/issues>.

Q: *I cannot compile Kilosort’s GPU code*

A: First try installing VSC2015 (make sure you have the Visual Studio ***Community*** version) and make sure you install the C++ compiler. Then ***restart your PC*** and try again. If it doesn’t work: uninstall and try VSC2013. If neither of them work, look for help here: <https://github.com/MouseLand/Kilosort/>

Q: *I cannot run any GPU code in matlab (i.e., gpuArray() fails)*

A: Make sure that you have the correct CUDA drivers installed for your GPU. Note that if you’re using anything other than a (modern) Nvidia GPU, you cannot run CUDA.

Q: *I found a bug*

A: Great! Or at least, it’s great that you found it, not that it’s there. If you’ve fixed it, you can make a pull request, otherwise you can create a bug report here: <https://github.com/JorritMontijn/Acquipix/issues>. Please copy/paste the matlab error message and as much detail as you can about what you were doing when it happened. If I cannot recreate the issue, I probably won’t be able to fix it.