

SpikeGLX pipeline

#####

READ ME FIRST:

The pipeline has been modified to add a few things:

1. Drift map - Generates a drift map for the session. Can be used to get a sense of the sessions recording quality (drift, stability etc.)
2. Drift metric – Appends an additional drift metric to the metrics.csv. An estimate of drift based on spike rate.

The installation of individual dependencies/software can be referred to using the links provided. The software highlighted in red saying “Clone from Server” should be cloned/downloaded from the lab’s Server and not the source websites since they are our custom versions.

Location: Li-Lab\Pipelines\Pipelines_Ephys\Kilosort_spike_sorting_NP\PreProcessing\setups

#####

The following repository has all the instructions needed to run the Spike GLX pipeline. https://github.com/jenniferColonell/ecephys_spike_sorting. Refer to the “Usage” section for running scripts. It is well documented at https://github.com/jenniferColonell/ecephys_spike_sorting.

I have compiled all the steps in this document but detailed information about each step can be found at the links provided.

Downloads/Installs required:

If you already have any of these dependencies, you can skip the download/install step for it.

1. **Python (v3.8.10) (Clone from Server)**

Since the code has been tested with v3.8.10, it is better to install the same version of Python. (<https://www.python.org/downloads/release/python-3810/>). Make sure to select “Add to PATH” during installation. If not, then it can be added later (similar to Figure 1)
Download the installer and run it.

2. **MATLAB (>=v2020b) (<https://www.mathworks.com/>).**

Sign in to your MathWorks account and follow installation.

You need 4 toolboxes for Kilosort to run.

- a. Parallel Computing Toolbox
- b. Signal Processing Toolbox
- c. Statistics and Machine Learning Toolbox
- d. GUI Layout Toolbox

These can be installed by launching MATLAB → Home → Add-Ons → Get Add-Ons. Search for the toolbox and install.

3. **CUDA Toolkit (v11.xx)** (<https://developer.nvidia.com/cuda-11-7-0-download-archive> --for v11.7)
Download and install CUDA Toolkit v11.xx following on-screen prompts. Once installed, it needs to be added to environment variables. To do this:
- Open Windows command prompt as administrator
 - Type `"rundll32.exe sysdm.cpl,EditEnvironmentVariables"` and press Enter
 - Click New
 - Edit variable name to: CUDA_PATH (or CUDA_PATH_VX_X if you have multiple versions)
Edit variable value to: *<Installation directory>*
 - Click OK. An example is shown in Figure 1.

You may have multiple toolkit versions available. You can use the following link to swap active versions. (<https://github.com/bycloudai/SwapCudaVersionWindows>). Keep in mind that the toolkit version used by a MATLAB version is fixed even if you swap to other versions in the system variables hierarchy. (See <https://www.mathworks.com/matlabcentral/answers/243613-matlab-link-to-new-cuda-toolkit>). Check the toolkit version compatible with your MATLAB version here (<https://www.mathworks.com/help/releases/R2021b/parallel-computing/gpu-support-by-release.html>)

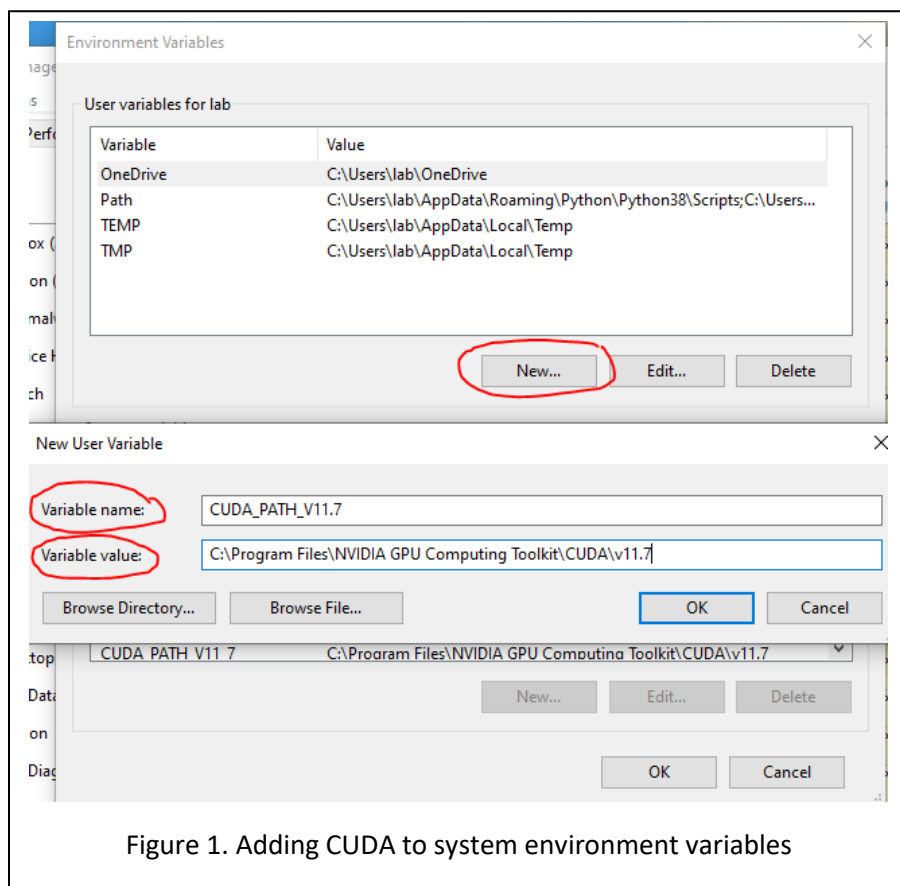


Figure 1. Adding CUDA to system environment variables

NOTE: You can compile CUDA MEX files using mexcuda without installing a CUDA Toolkit as (since R2019b) Parallel Computing Toolbox ships the correct version of the CUDA Toolkit to do this. (See <https://www.mathworks.com/matlabcentral/answers/1793600-minimum-maximum-cuda-toolkit-requirement-for-r2022a>)

I have tested this on my personal laptop and it is true, the Toolkit is not required to compile CUDA MEX files. This needs more extensive testing and as of now, for the sake of the pipeline, it would be better to install the Toolkit. In the future, this step might not be required.

4. Visual Studio (>=v2019) (Optional: Clone from Server)

You may need to check that the version of Visual Studio you install is compatible with your version of CUDA. (<https://visualstudio.microsoft.com/vs/older-downloads/>).

Kilosort requires a CPU compiler (on Windows machines) and Visual Studio is a freely available one. When installing Visual Studio:

- Make sure to select C++ during installation (Figure 2). It is not selected by default. This can also be done later by launching the installer and clicking 'Modify'.
- If you had previously used a different CPU compiler in MATLAB or this is a new installation of a CPU compiler (Visual Studio in our case), you must switch to the CUDA-compatible compiler by typing "*mex -setup C++*" in the MATLAB command window.

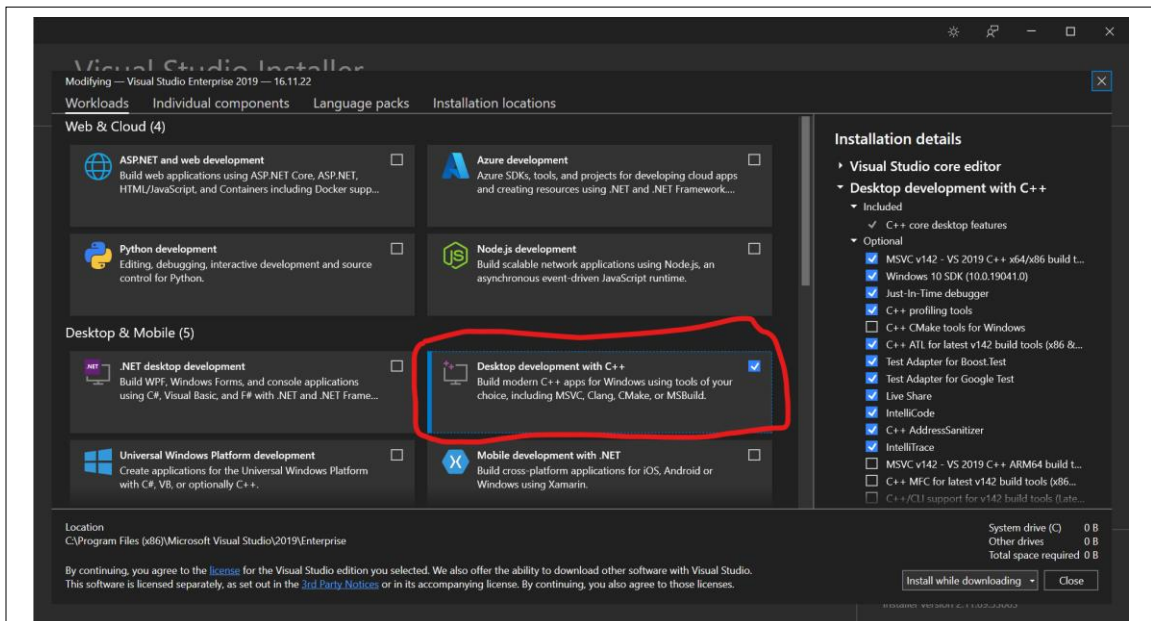


Figure 2. Select C++ compiling dependencies during Visual Studio installation.

5. **pipenv (needs python >=3.8)**

To install open Command Prompt **as administrator** and type “*\$ pip install --user pipenv*”.

After installing pipenv, close the command prompt and re-open it but **not as administrator** for the following steps.

Also add the following directories to PATH:

C:\Users\labadmin\AppData\Local\Programs\Python\Python38

C:\Users\labadmin\AppData\Local\Programs\Python\Python38\Scripts.

6. **Ecephys spike sorting repo (Clone from Server/Github)**

Download the code from github (unzip if necessary) and keep it in a convenient location.

Navigate to the ecephys spike sorting directory in the command window and run the following:

```
$ set PIPENV_VENV_IN_PROJECT=1
```

```
$ pipenv install
```

```
$ pipenv shell
```

```
(.venv) $ pip uninstall setuptools
```

```
(.venv) $ pip install setuptools==59.8.0
```

```
(.venv) $ pip install .
```

7. Next, check if MATLAB can recognize the GPU correctly. Open MATLAB (v2020b or later) and run `gpuDevice()` in the command window. If the drivers are up to date and compatible with MATLAB, you will see an output similar to Figure 3. If you get an error, try:
 - a. Quit MATLAB and update the drivers for the GPU card -- this can be done with the Device Manager in Windows 10, and will also happen automatically if you update the CUDA Toolkit.
 - b. Restart MATLAB and enter `gpuDevice()` to ensure it is recognized.

Once this step is successful, close Windows command prompt and re-open **as administrator** to run the following:

```
$ pipenv shell
```

```
(.venv) $ cd <matlabroot>\extern\engines\python
```

```
(.venv) $ python setup.py install
```

NOTE: <matlabroot> will be the path to your MATLAB installation folder.

For example: C:\Program Files\MATLAB\R2022a

```
>> gpuDevice()

ans =

    CUDADevice with properties:

        Name: 'NVIDIA RTX A4000'
        Index: 1
        ComputeCapability: '8.6'
        SupportsDouble: 1
        DriverVersion: 11.7000
        ToolkitVersion: 11.2000
        MaxThreadsPerBlock: 1024
        MaxShmemPerBlock: 49152
        MaxThreadBlockSize: [1024 1024 64]
        MaxGridSize: [2.1475e+09 65535 65535]
        SIMDWidth: 32
        TotalMemory: 1.7171e+10
        AvailableMemory: 1.5830e+10
        MultiprocessorCount: 48
        ClockRateKHz: 1560000
        ComputeMode: 'Default'
        GPUOverlapsTransfers: 1
        KernelExecutionTimeout: 1
        CanMapHostMemory: 1
        DeviceSupported: 1
        DeviceAvailable: 1
        DeviceSelected: 1
```

Figure 3. Command Window output for `gpuDevice()`

8. SpikeGLX, CatGT, TPrime & C_Waves (Clone from Server)

Download the versions specified for SpikeGLX, CatGT, TPrime and C_Waves using the link below (unzip if necessary) and keep them in a convenient location.

<http://billkarsh.github.io/SpikeGLX/#command-line-tool-installation>

- a. SpikeGLX (Release_v20221212-phase30).
- b. CatGT (Use experimental version on lab's Github with 'gbldmx' till it is officially released)
- c. TPrime (v1.7)
- d. C_Waves (v2.0)

9. **Kilosort (v2.5)** (Clone from Server)

Documentation: <https://github.com/MouseLand/Kilosort>

Download v2.5 (unzip it) and keep it in a convenient location.

You can check if Kilosort is running by starting MATLAB and:

```
>> cd \my\kilosort\directory\
```

```
>> kilosort
```

10. **Npy-matlab** (<https://github.com/kwikteam/npymatlab>) (Clone from Server)

Download the code from Github (unzip if necessary) and keep it in a convenient location.

11. **spikes-master** (Clone from Server)

Used for generating drift map. Download and keep it in a convenient location.

After completing these steps successfully, the pipeline should be ready to run. Table 1 provides the versions of software used (as of 12/30/2022) and tested for a working pipeline.

Software	Version
Python	3.8.10
MATLAB	R2022a
CUDA Toolkit	11.2
SpikeGLX	v20221212-phase30
CatGT	Experimental version
TPRime	v1.7
C_Waves	v2.0
Kilosort	v2.5
Visual Studio	2019 (Community)

Table 1. Software versions for a working pipeline (as of 12/30/2022)

**Software highlighted in red should be cloned from Github.