Spike Sorting Procedures

# Function Overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Input | Process | Output | Notes |
| 1 | Folders with .continuous files (open ephys output)  **OR**  YYMMDD\_01.bin  YYMMDD\_02.bin  YYMMDD\_06.bin… (a collection of bin) | formatBin\_wrapper.m (SECTION 1)  **OR**  formatBin\_wrapper.m (SECTION w) | **Split** data\_padded.bin  **Split** data\_adc.bin  **Merged** data\_merged.bin  **Meta** merge\_info.csv | Adds desired data channel padding.  Interlaces data channels for increased sampling rate.  Combines (merges) multiple trials from the same experiment for ease of spike sorting.  Extracts ADC channels.  **NOTE: Merge\_info.csv needs to be read back in Split\_trials.m** |
| 2 | YYMMDD\_merged\_converted.bin  StandardConfig.m  chanMap.mat | **Local:** Kilosort\_wrapper.m (KiloSort)  **HPC:** kilosortJob.sh | spike\_times.npy  spike\_clusters.npy  ….  params.py | **This is the KiloSort process - Matlab**  Push it through HPC if the data is long or if no GPU  **HPC:** Modify kilosortJob\_general.sh |
| YYMMDD\_merged\_converted.dat  (rename .bin to .dat)  YYMMDD\_merged\_converted.prm  PROBE.prb | **Local:** klusta YYMMDD\_merged\_converted.prm  **HPC:** klustaJob.sh | YYMMDD\_merged\_converted.kwik | **This is the Klusta process - Python** Always push through HPC |
| 3 | Output from 2 | **Kilosort:** phy template-gui params.py  **Klusta:** klustaviewa **or** phy kwik-gui YYMMDD\_merged\_converted.kwik | Updated versions of outputs from 2 | This is mostly to mark noise clusters and the good ones |
| 4 | spike\_times.npy  spike\_clusters.npy  ….  YYMMDD\_merge\_info.csv | extractTrialUnits\_wrapper.m (SECTION 1)  **Run section by section (ctrl+enter)** | YYMMDD\_XX\_sorted.mat   * s * trial\_spikes * spike-clusters | Unit extraction |
| YYMMDD\_merged\_converted.kwik |
| 5 | YYMMDD\_XX\_sorted.mat  YYMMDD\_XX.bin | extractTrialUnits\_wrapper.m (SECTION 2-3)  **Run section by section (ctrl+enter)** | YYMMDD\_XX\_extracted.mat   * s (updated) * m * filenames | Wave extraction + secondary template match |

# Step by Step Guide

1. Ensure your data matches the form YYMMDD\_XX.bin
   1. If you are using **OpenEphys** Data hold on, skip to Step 4a)
2. Run Merge\_trials.m – this script allows you to merge multiple experiments done on the same animal on the same day – it’s output also helps further analysis.
   1. Change nChans to match the number of channels your data has
   2. Change numFiles to represent the XX numbers in your binary file name (See Step 1)
      1. Note: this variable is an array, so you can use conventions such as 1:5 to merge trials 1 to 5
   3. Change namePart1 to the name of your data file (without the extension)
   4. Change namePart2 to the extension of your binary data file – e.g. .bin or .dat
3. Step 2) will produce a \_merge\_info.csv file output, this stores the names of the files merged and how many samples long they are.
4. Run convertOurData.m [**Only for KiloSort**] **– KiloSort** does some math that requires at least 4 channels – hence we add some dummy channels (of 1s)
   1. **[Only if coming from 1a] OpenEphys** does not output a binary file – hence we use:
      1. convertOurData\_OE.m
         1. SetdataChan to the <> in the OpenEphs 100\_CH<>.continuous files
         2. Set nChans desired to length(dataChan)
      2. convertOurData\_OE2.m
         1. SetdataChan to the <> in the OpenEphs 100\_CH<>.continuous files – the order of numbers in dataChan is important. This array is split in half and the second half is interlaced into the first half. E.g. If [1 2 3 4 5 6] – 1 and 4 will be interlaced, 2 and 5, 3 and 6.
         2. Set nChans desired to length(dataChan)/2
      3. ADEL NEEDS TO MAKE IT SAVE THE ADC CHANS TOO
      4. Use the output binary files in Step 1)
   2. Set nChans to match the number of channels your data has
   3. Set sigChannel to match which channel(s) stores your actual ephys signals (this is an array so you can specify multiple)
   4. Set nChansDesired to match the number of channels you want your data to have – if it is equal ot the number of signal channels then it will not add any dummy channels
   5. Change namePart1 to the name of your binary data file (without the extension)
   6. ADEL NEEDS TO MAKE IT SAVE THE ADC CHANS TOO
5. We know have the raw data to run the sorting programs:
   1. KiloSort
      1. Open the config file (e.g. StandardConfig\_MOVEME.m)
      2. Change ops.fbinary to the name of your binary file
      3. Change ops.root to the starting directory for KiloSort (e.g. storing where your binary is) – if you are running on the HPC this value is empty – because you are expected to keep the config file and master file in the same place as your data.
      4. Change ops.chanMap to the appropriate channel map:
         1. If you are using 1 data channel with 3 dummy channels use: chanMap-1x4.mat
      5. You can now run the master\_file\_example\_MOVEME.m
   2. Klusta
      1. Rename the .prm file to match the name of your binary file
      2. Rename the extension of the binary file to .dat
      3. Edit the .prm file
         1. experiment\_name – should match the name of your binary file
         2. prb\_file should match the electrode configuration you are using:
            1. If first channel is signal and remaining are dummy / if single electrode: 1x1\_electrode.prb
6. After the sorting is complete we use Split\_trials to pull out the spike times associated with each cluster
   1. KiloSort – Spil\_trials.m
      1. Change the files variable so that it contains the starting 6 numbers of your data e.g. [‘181028’] – quotation marks are needed
      2. Change startTrial to the first data set used in the merge e.g. if merged trials 4 to 6, startTrial = 4
         1. NOTE: this only changes the naming of the output – e.g. the output would be 181028\_4.mat instead of 181028\_1.mat if you forgot to change startTrial to 1 from 4.
      3. Put the \_merge\_info.csv in a place where the code can access it e.g. in path or same folder
      4. ADEL SHOULD MAKE THIS LESS CONFUSING BY CHANGING CODE TO USE FILEDETAILS>FILENAME FROM READCSV
   2. Klusta – Split\_trials2.m
      1. For this to work corretcly you may need to open KlustaViewa or Phy and ‘Move Unit to Good’
      2. Change kwikFileName to match the filename of your kwik output
      3. Put the \_merge\_info.csv in a place where the code can access it e.g. in path or same folder
      4. ADEL NEEDS TO MAKE THIS LESS CONFUSINGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG – READ CSV FIRST WHICH CAN THEN BE USED TO GET FILENAME?
7. Analysis Code
   1. ADEL NEEDS TO CREATE A BAREBONES VERSION THAT EXTRACTS WAVEFORMS ONLY