

This is a guide for batch processing all tetrodes with mountainsort pipeline and converting the output to LFPO-readable form for post-processing. The mountainsort pipeline is based on instructions in [Mountain_Sort_25/10/19](#)

Relevant Python/Bash scripts are located in `/d1/data/bin/igor`, which is normally part of the data user's PATH variable. If you run this pipeline as a different user, make the scripts available from your PATH. You need `ms_run_and_adapt.sh`, `ms_lfpo_proc_tet.sh`, `ms_merge_clures.py`, `sgclust_to_lfpo.py`, `mount_sortb.sh`, `mountain_to_sgclustb.py`, `axtrk_to_whl.py`, `Dat_to_Mda.py`, `ws_interpolate`, `dat_to_mdab.sh`.

Execute the following commands in the working directory containing `dat` and `axtrk` files.

1. Copy relevant files to working directory:

```
cp /d1/data/lfpo1/ms_and_lfpo/* .
```

2. Remove unwanted channels from `TEMPLATE.par`

3. Create a file named **BASELIST** with a list of basenames (1 per line) of `dat/axtrk` to be processed, in chronological order e.g.:

```
jc103-3005_01l
```

```
jc103-3005_02s
```

```
Jc103-3005_03l
```

To create one for all `dat` files in working dir, run (double check session order):

```
for f in *.dat; do printf "%s\n" "${f%.dat}"; done > BASELIST
```

4. Run master script with following parameters: ANIMAL, DAY, DTHOLD (spike detection threshold for mountainsort), NCHAN (64 or 128) e.g.:

```
ms_run_and_adapt.sh 5401 07102020 5 34
```

5. Open the output in LFPO by specifying tetrode number and number of channels in this tetrode, e.g.:

```
./lfpo_online ../Res/spike_display_mountainsort_TEMPLATE.conf  
channel.num=64 animal=jc103 day=3005 tet=1 nchan=4
```

6. [Optional] Perform the semi-automated cleaning steps (see below)

7. [Optional] Open clusters from all tetrodes together:

a. Merge clu/res: `ms_merge_tetrodes.sh jc103_3005`

b. [OPTIONAL] Adjust `pf.sessions/pf.groups` - if you want to build rate maps

c. Open with config `spike_display_mountainsort_alltet.conf`

8. [Optional] Adapt output to Jozsef's dataset format:

a. `ms_to_sgclust.py sgclust/DRCK15-11032018 ./ 0`

Parametres are: 1. basename for output files 2. Directory containing tet*/

3. 1 if downsampling 24kHz->20kHz is needed

b. `clures_merge_tetrodes_by_sessions.py DRCK15-11032018-1724`

Run this in the output directory of previous step (sgclust/ in example above)

- Template display config file `spike_display_mountainsort_TEMPLATE.conf` is in `/d1/data/lfpo1/Res/`. It also relies on tetrode config files `tetr_1tetr_*` that are in `/d1/data/lfpo1/Res/tetr/`

- For post-processing, the most useful feature is displaying spikes that contaminate the autocorrelogram, which can be triggered with '**c**'.

KNOWN ISSUES

- Pipeline doesn't work with symbolic links to *.dat
- Might not work with a single tetrode - add a dummy tetrode in TEMPLATE.par to get over
- If a whole tetrode is broken, in the tetr configuration file just remove the line that contains the channels (instead of writing "0 [channels]")

SEMI-AUTOMATED CLUSTER CLEANING

By default, all scripts are run in the day directory (the one containing tet/ directories)*

This pipeline assumes you have first executed the mountainsort pipeline (see above).

1. [Pre-clean in **lfp_online**] Quickly go through all clusters - obvious splits/merges, delete hopeless clusters
2. [FIT] Run cleaning script in the Fit mode **nohup autoclean_alltet.sh &**
3. [CLEAN] When fit ruIT mode - takes few minutes per cluster:
n is over, run clearing script in CLEAN mode:

run_autoclean.sh

For every cluster, choose % of dirtiest spikes to be removed based on cleaning curve - click with middle mouse button or just close window to ignore the cluster.

Results are written to path with added 'clean' suffix after all clusters in the current tetrode have been cleaned.

4. [Replace clu/res] This will simply backup pre-clean clu/res and replace then with cleaned ones: **copy_preclean.sh**
5. [Post-clean] Open results in **lfp_online** to merge clusters and clean cross-contaminations.
6. [OPT] To redo cleaning, first restore backup from step 2 and run script in CLEAN mode

RUNNING SEMI-AUTOMATED CLUSTER CLEANING ON DIFFERENT MACHINE

Since FIT run takes time and resources, you can run it on a different machine. For a FIT run you only need fet, clu and res files. For a CLEAN run you then need the files in AUTOCLEAN/ directory that is created in every tet*/ directory. For example, in case of running mountainsort on *pyramidal*, doing pre-cleaning on *home* computer and want to run cleaning in FIT mode on *mec*, you need these commands to sync files:

on mec, in animal data directory

rsync -avP --include="*/" --include="*.fet.0" --exclude="*" igor@pyramidal:/hdr/data/bindata/\$ANIMAL/\$DAY

.

```
rsync -avP igor@pyramidal:/hdr/data/bindata/$ANIMAL/$DAY/TEMPLATE.par $DAY/
# on home, after pre-clean step
rsync -avP --include="*/" --include="*.clu" --include="*.res" --exclude="*" tet*
igor@mec:/temp_store/igor/$ANIMAL/$DAY
```

After FIT that, copy AUTOCLEAN/ directories back to *home* by running:
 # on home, in day directory (with tet*/ directories)
rsync_autoclean.sh igor@pyramidal:/hdr/data/bindata/\$ANIMAL/\$DAY

If you also do pre/post-cleaning on a machine different from the one where mountainsort has been running, you can use this command to avoid copying large spk files (it will interpolate the original waveshapes locally):
rsync_msout.sh igor@pyramidal:/hdr/data/bindata/\$ANIMAL/\$DAY; run_ws_interporalte.sh

FEATURE REQUESTS FOR LFPO [OPTIONALLY] - assign priority on scale 1-5

- (5) Add the ctrl+z or some command for undoing merging
 - Implemented but needs testing
- In relation to the above, maybe also how many spikes fall within the refractory period (maybe when you press c to show them); so one can decide if they are a low percentage of the total...
 - Now can be done with *clu_refractory_fraction.py* (/d1/data/bin/igor)
- A way to see the inter spike interval of a specific cluster in more detail with a scale bar (e.g. so one can tell how wide the refractory period of a cell actually is, in ms)

HOW TO GET LATEST CLUSTERING SOFTWARE

If you don't have code, do **git clone**

https://github.com/igridchyn/lfm_online.git

To update: **git pull origin master**

To compile: (while in lfm_online/sdl_example/Debug) **make lfm_online**

FOR YOAV

Merge whl before running **ms_run_and_adapt.sh**

It is not crucial to do it before, but it is easier.

Before merge, generate session_shifts.txt file. The DIV variable here should be equal <NUMBER OF CHANNELS X 2>, change it if number of channels is different from 34

```
DIV=68; ls -lU `awk '{print $1".dat"}' BASELIST` | awk '{sum+=$5/'$DIV';
$5=sum ; print $5}' > session_shifts.txt
```

```
merge_whl.py session_shifts.txt 4201-04082020.whl 4201-04082020_01.whl
4201-04082020_02.whl 4201-04082020_03.whl 4201-04082020_04.whl
4201-04082020_05.whl 4201-04082020_06.whl 4201-04082020_07.whl
```

For positrack, convert *.positrack files to whl:

positrack_to_whl.py 4201-04082020_01.positrack <SHIFT>

Where <SHIFT> is the timestamp of the first tracking TTL sync signal,

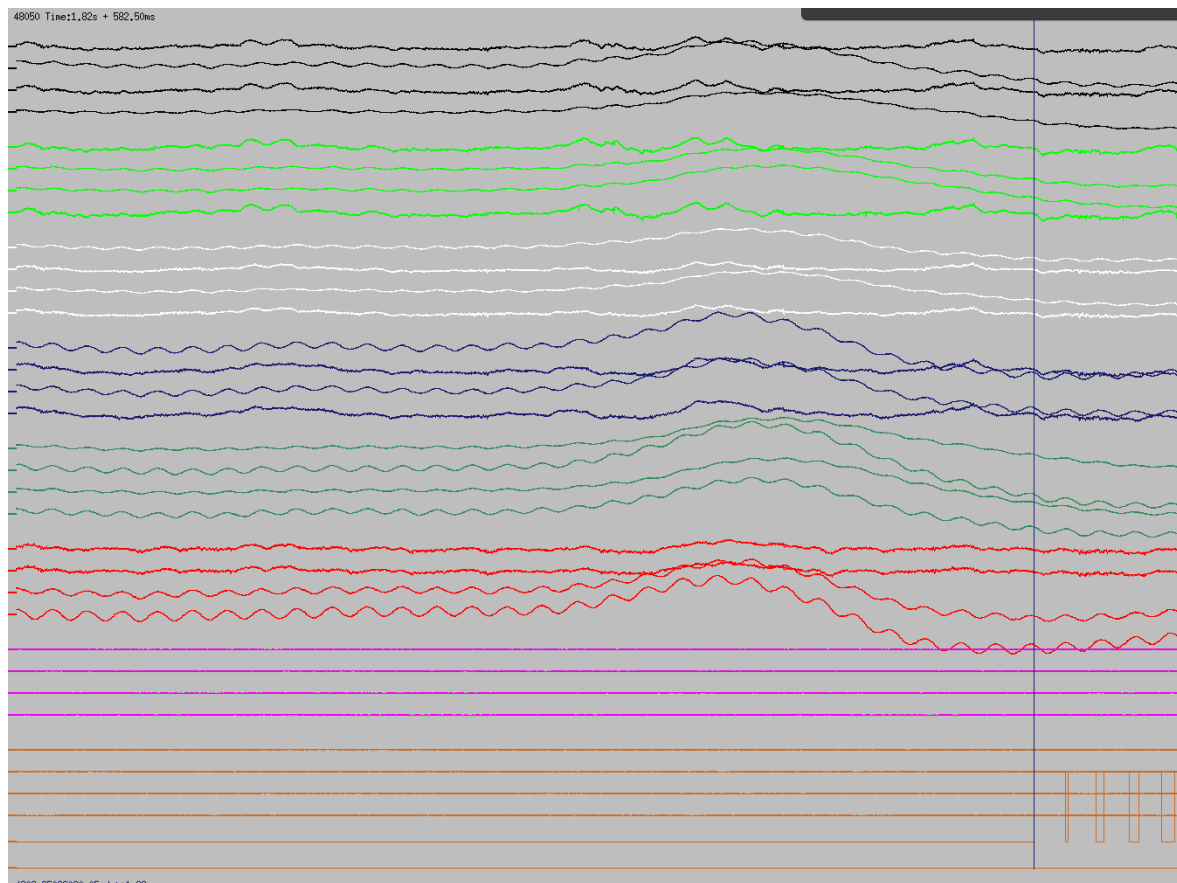
which can be seen in **regaamc11** by putting pointer on it:

(number is in the top left corner, pointer is invoked by 'p')

Command to run regaamc11:

regaamc11 4201-04082020_01.dat 34 50 ../regaa_34_chan.conf

The config file can be found in one of the data folders or in /dl/data/lfp01.



Copy clustering output to ca3. Change IP address if needed.

rsync -avP --exclude='*.mda' tet* guest03@10.4.44.175:/mnt/data_yoav/data/6001/02112020

Run lfp_online to do clustering

**./lfp_online ../Res/spike_display_mountainsort_ca3.conf channel.num=64
animal=4001 day=25072020 tet=3 nchan=4**

If d1 is not mounted ('ls /d1' shows empty folder), mount it with **sudo mount alpha2:/d1 /d1**

cd ~/lfp_online/sdl_example/Debug/

ms_to_sgclust.py sgclust/4001-24072020 ./ 0

cp ../4001-24072020.par .

clures_merge_tetrodes_by_sessions.py 4001-24072020

5.- Generate .light files for every session

\$ dat2timestamps.py 4201-04082020_02.dat 34 33 20 4201-04082020_02.light