

# Ephys pipeline overview

monkeypsych ↔ TDT ⇒ [https://github.com/dagdpz/Phys\\_preprocessing](https://github.com/dagdpz/Phys_preprocessing)

## Folder structure

| Inputs   | Info   |
|--|--|
| Y:\Data\TDTtanks\Monkey_phys                             | Raw TDT data   |
| Y:\Data\Sortcodes\Monkey_phys                            | sorting related files  |
| Y:\Data\Monkey   | behavioral data  |
| \Dropbox\DAG\phys\Monkey_phys_dpz\Electrode_depths.mat   | electrode depths documentation   |
| \Dropbox\DAG\phys\Monkey_phys_dpz\Same_cells.mat         | same cells across blocks assignment  |
| Outputs  | Info   |
| Y:\Data\Monkey_phys_mat_from_TDT                         | recorded data, arranged in the same trial structure as the behavioral data   |
| Y:\Data\Monkey_phys_combined_monkeypsych_TDT             | trial structures with combined behavioral and ephys data                     |
| Y:\Data\All_phys_preprocessing_log\Monkey_phys           | automatically created log files of all preprocessing performed               |
| \Dropbox\DAG\phys\Monkey_phys_dpz\plx_files.xls          | (for multiple sortings:) used plx file documentation                         |
| \Dropbox\DAG\phys\Monkey_phys_dpz\Mon_sorted_neurons.xls | block/run/unit electrode locations, IDs, ratings, and procedures information |

## General Workflow

1. Fill in [Electrode depths file](#) (required as first step for WC pipeline)
2. WC: Create waveclus pre-clustering files (implemented in [phys\\_gui](#))
3. WC: Run waveclus and sort for all channels, don't forget to save (wave\_clus3new3, MATLAB 2014+)
  - Preferentially separate spike shapes in difficult cases
4. Create plxfiles (implemented in [phys\\_gui](#))
5. Check and clean PLX file and **TAKE NOTES** (single/SNR/stability rating)
6. Now we need to assess which units are the same within and across blocks (based on channel, electrode depths, spike shapes and potentially tuning, carefully check notes for this). For that purpose you might want to do the following steps in any order (potentially even going back and forth):
  - Create combined (phys+behavior) files (implemented in [phys\\_gui](#))
  - Assign same cells across blocks in [Same cells file](#)
  - Plot single unit tuning assuming cells are unique in every block (implemented in [phys\\_gui](#))
    - This is meant to help assessing if two "units" in the same block might actually be the same (if they have very similar tuning properties)
  - Re-sort PLX file (combining same cells sort codes of same block same channel)
7. If you have finalized the sorting, create combined (phys+behavior) files (implemented in [phys\\_gui](#)) (yes, once again, now that you have a different amount of cells)

8. Complete [Same cells file](#)
9. Automatically update the automatic\_sorting sheet from the [Excel sorting table](#) using [phys\\_gui](#)
10. Copy the automatic\_soting entries to the final\_sorting sheet of the [Excel sorting table](#) and complete with manual entries.

## Associated code

For loading and writing plx files we use an external toolbox located in Github  
[External\\_unmodified/Matlab Offline Files SDK repository](#)

The remaining code related to phys preprocessing (not including Waveclus and Plexon) can be found on Github [Phys\\_preprocessing repository](#)

- Data conversion functions
  1. TDTbin2mat\_working.m Debugged and modified version of TDTbin2mat
  2. SEV2mat\_working.m Debugged and modified version of SEV2mat
  3. PLX2SPK.m For reading in PLX files (Stefan Schaffelhofer)
  4. SPK2PLX.m For writing PLX files (Stefan Schaffelhofer)
  5. WC32SPK.m
  6. WC32SPK\_concatenated.m
  7. WC32SPK\_directly.m
- Core functions
  1. phys\_gui\_working.m
  2. phys\_gui\_execute.m
  3. TDT\_trial\_struct.m
  4. ph\_combine\_MP\_and\_TDT\_data
  5. DAG\_update\_sorting\_table.m
- Waveclus pipeline specific
  1. DAG\_WC3\_preprocessing.m
  2. DAG\_parse\_data\_tdt.m
  3. DAG\_SpikefilterChan.m
  4. WC32SPK\_directly.m
  5. ph\_readout\_broadband\_lag.m
- Plexon pipeline specific
  1. DAG\_create\_PLX.m
  2. DAG\_update\_plx\_file\_table.m
  3. ph\_get\_new\_plx\_extension.m
- Extra standalone functions
  1. DAG\_derive\_TDT\_streamer\_broadband\_lag.m
  2. ph\_debugging\_GUI.m
  3. DAG\_take\_over\_sortcode\_PLX2PLX.m
- Only of historical relevance (legacy subfolder)
  1. DAG\_move\_sorting\_files.m Was used to transfer sortcodes to their current destination
  2. ph\_derive\_electrode\_depth.m Was used to create the first electrode\_depths file extracting from sorted\_neurons table
  3. ph\_derive\_same\_cells.m Was used to create the first same\_cells file extracting from sorted\_neurons table
  4. ph\_simulate\_history.m Was used to create the first log files
  5. PLX2TDT.m Was used to create TDT sortcodes from PLX files

## 6. TDT2PLX.m Was used to create PLX files from TDT sortcodes

From:

<http://dag.dokuwiki.dpz.lokal/> - **DAG wiki**

Permanent link:

[http://dag.dokuwiki.dpz.lokal/doku.php?id=ephys\\_pipeline:1\\_pipeline\\_overview](http://dag.dokuwiki.dpz.lokal/doku.php?id=ephys_pipeline:1_pipeline_overview)

Last update: **2022/12/29 07:15**

