Trodes

TrodesToMatlab

These are instructions for importing information from Trodes .rec files into matlab.

1. Setup

Setup for first-time users

Matlab will need to know the path to various files. In matlab, edit the path to add the following folders (without subfolders):

TrodesToMatlab
TrodesToMatlab/TrodesToFilterFrameWork
Trodes program folder

Setup for each recording session

1) All .rec files associated with a recording session should go into a folder dedicated to that session. The camera module output and other outputs for that session should go there too.

2) When using Trodes, if you use the Annotate function, a .trodesComments file gets created. The format of the file is simple, it has one line for each entry. The file can be edited later if you want to change the time entries. If you didn't use annotation during the recording, you will need to create one .trodesComments file for each .rec file (it should have the same name, but a different extension). Here is what one file will look like that has two epochs (between 5:00 and 25:00 and then between 30:00 and 50:00):

```
9000000 epoch start
45000000 epoch end
54000000 epoch start
90000000 epoch end
```

Times are counted sampling frequency, so 30000 is the same as 1 second. If the session is spread across multiple .rec files and the clock was reset in one or more of them, you need to add ‘time reset’ as the first line of trodesComments files with the reset time.

The .trodesComments file should have all the searchable terms in it to describe the epochs. This is important for when the info gets imported to filter framework. This is done with a <fieldname fieldvalue> line. Avoid using spaces in either the fieldname or the fieldvalue, but there should be a single space in between. The value can be either a string or a number. Plus, any events that happened during the recording uses <timestamp description> lines. For example:

```
time reset
9000000 epoch start
task sleep
environment sleep_box
45000000 epoch end
54000000 epoch start
task run
environment track_a
70000000 injection saline
90000000 epoch end
```

3) You can reconfigure any .rec file if you do not want to use the same settings as those used when the data were saved. The way this works is that if there is a .trodesconf file in the same directory as the .rec file, and with the same name as the rec file, then Trodes will use that workspace during playback instead of the one embedded in the recoding file. If you open up the recording file as a source in Trodes, change some settings, then save the workspace, the default name will be the right one to use.

4) The extraction code will follow that same rule. If a .trodesconf file exists with the same name, it uses the settings in that file by default. So now, you can tweak your reference and filter settings while seeing and hearing the data streaming, and then use those settings during export. Now, you are ready to export data for this session.

2. Data extraction

Extracting time- required for all later steps

Let's say your .rec file is called myFile.rec. For extraction, the file mask will be 'myFile'. If you have more than one .rec file from the same session, make sure they have the first part of the names in common (example myFile01.rec and myFile02.rec would use a 'myFile' mask).

1) In matlab, navigate the working directory to the the session folder.
2) extractTimeBinaryFile('myFile')

A new folder is created with the .time extension. Inside there is a file containing a list of all continuous blocks of recording. Tne file can be read in to matlab using readTrodesExtractedDataFile.m.

Extracting spikes

Let's say your .rec file is called myFile.rec. For extraction, the file mask will be 'myFile'.

1) In matlab, navigate the working directory to the the session folder.
2) extractSpikeBinaryFiles('myFile')

A new folder is created with the .spikes extension. Inside there is one file for each nTrode containing the time of each spike and the waveforms. Any one file can be read in to matlab using readTrodesExtractedDataFile.m.

Extracting Matclust files

After spike extraction, if you want to create matclust files for spike sorting, type "createAllMatclustFiles" and another folder with the .matclust extention will be created with matclust formatted files. Each param_* can be opened by matclust. There will be a times.mat file in the .matclust dir that you can use to cluster by your defined epochs in the .trodesComments file. After clustering, make sure you keep the default name for saving, with the "matclust_*" pattern.

Extracting LFP

Again, let's say your .rec file is called myFile.rec. For extraction, the file mask will be 'myFile'

1) In matlab, navigate the working directory to the the session folder.
2) extractLFPBinaryFiles('myFile')

A new folder is created with the .LFP extension. Inside there is one file for each channel containing the filtered LFP traces. Any one file can be read in to matlab using readTrodesExtractedDataFile.m.

Extracting DIO's

1) In matlab, navigate the working directory to the the session folder.
2) extractDIOBinaryFiles('myFile')

A new folder is created with the .DIO extension. Inside there is one file for each channel containing the DIO state change times. Any one file can be read in to matlab using readTrodesExtractedDataFile.m.

Position tracking

If your experiment requires tracking an animal's position using the camera module, you will need .videoPositionTracking files saved in the same directory as the .x264 files. You can have multiple .videoPositionTracking files for each .x254 file, as long as their data do not overlap in time. The .videoPositionTracking files are created during data collection if tracking is turned on. Otherwise, you can create them offline by running the CameraModule program and opening the .x264 file(s).

3. Raw data access functions

We also provide matlab functions to directly read in data from the .rec file into the matlab workspace. Note, that these functions are quite slow compared to the extraction methods described above. Here is a list of provided functions:

readTrodesFileContinuous.m — used to read in raw continuous data from selected neural channels from the .rec file.

readTrodesFileDigitalChannels.m — used to read in raw digital I/O channels from the .rec file

readTrodesFileAnalogChannels.m — used to read in continuous analog I/O channels from the .rec file.

readStateScriptFileTrials.m — used to parse a StateScript module log file.