Manual

Structure of the repository

This repository contains data from three multi-electrode array recordings from marmoset retinas that were stimulated with spatiotemporal white-noise patterns of light. The data accompany the manuscript by Zapp et al: "Accelerated spike-triggered non-negative matrix factorization reveals coordinated ganglion cell subunit mosaics in the primate retina". Each directory comprises one recording session and is named by recording date and an identifier for right or left eye, YYYYMMDD_MEATYPE_EYE_RETINALREGION. We used internal specifications for indicating the retinal region (e.g., i1 for inferior, s3 for superior). Each directory comes with a file named *expdata.json*, containing general experiment information, and with one or several directories for each stimulus recording, containing the following files.

- *parameters.json*: contains stimulus-specific parameters
- *frametimes.txt*: contains the timestamps at which the stimulus changed, or "frame times"
- *spiketimes_cell*.txt*: contains the timestamps of spikes, or "spike times". One file for each spike-sorted unit, or "cell".

The files *expdata.json* and *parameters.json* are of the file format *JSON* that is both easily parseable but also human-readable in a text editor. The timestamp files are conventional *TXT* files where each line is one timestamp in seconds counting from the start of the stimulus recording. The choice of formats allows inspection of all files without complex loading routines.

One experiment directory (*20180726_60MEA_left_33*) additionally contains the file *movie0000.h5*. This *HDF5* file contains the stimulus of a natural scene.

1 General experiment information ("expdata.json")

The file contains the following variables:

`animal`	Information on the animal, including sex, age in years, and which region that the recorded retina piece stems from.
`array`	Information on the multi-electrode array, including the type, number of electrodes, their diameter, and the inter-electrode distance in meters.
`fs`	Recording sampling rate in Hz.
`projector`	Information on the light projection system, including the screen refresh rate, screen display delay, mean light level, screen size in pixels, and the pixel size on the retina in meters.

2 Raw spike times, stimulus details and reconstruction

The retinal ganglion cell responses to visual stimulation are represented with raw spike times and stimulus frame times. Spike times were extracted with a Kilosort-based algorithm from the multi-electrode-array recorded data and stored in seconds in individual files for each spike-sorted unit named with a unique cell identifier.

To align recorded spike times with stimulus presentation, we also recorded pulses that indicate when a new stimulus appears on the screen. The times of these pulses, or frame times, are represented here in seconds. How the pulse frequency aids in stimulus reconstruction is specific to each stimulus and described in the file *parameters.json*. This file contains the following stimulus-specific parameters (written in the following paragraphs in `block letters`) that are necessary for the stimulus reconstruction.

2.1 FrozenNoise

The spatiotemporal white-noise stimulus contained in this repository is called *FrozenNoise*. The stimulus consists of black and white squares (100% contrast). Each square was randomly assigned to black (-1) or white (1) with a probability of 50% each, and had a side length of `stixelwidth` pixels. Spatial patterns were updated with a frequency of `fps/Nblinks`, where `fps` is the monitor refresh rate which is, here, equivalent to the provided pulse

frequency. For screen updates for `Nblinks=3` refresh rates, we therefore used every third frame time to recreate when a new white-noise image appeared.

To obtain the number of spikes per white-noise image, the spike times can be binned by the frame times using the provided Python script `binspikes.py`. Example usage of the function:

```
from binspikes import binspikes
runningbin, frozenbin = binspikes('20220208_60pMEA_right_i1/FrozenNoise_1')
```

A fixed "frozen" white-noise sequence was repeated periodically between series of "running" white-noise sequences. In particular, **after** `RunningFrames`, a number of `FrozenFrames` was presented with a fixed random-number-generator seed `secondseed`. After each run of the fixed sequence, "running" presentations were resumed with the last seed that was generated before the fixed sequence presentation. The "frozen noise" can be used as held-out data for model testing.

Check the repository https://github.com/gollischlab/RecreateWhiteNoiseStimuliWithPython (for Python) or alternatively https://github.com/gollischlab/RecreateWhiteNoiseStimuliWithMatlab (for MATLAB) for reconstructing the frame sequence. Minimal usage example in Python, where `p` is a dictionary loaded from *parameters.json*:

```
from retinawhitenoise import binarystimulus as stimulus

# Create the 18-trials-long stimulus trial-by-trial in a loop
for stim in stimulus.recreate(p['seed'], (p['Nx'], p['Ny']), p['RunningFrames'], 18):
    pass
    # At each iteration 'stim' is a (p['Nx'], p['Ny'], p['RunningFrames']) array
```

Some of the recording sessions contain several directories with stimulus parameter and timestamp files. These are separate stimulus recordings of the same recording session. In order to analyze the data together, binned spikes and recreated stimuli of the subdirectories can be concatenated.

2.2 Checkerflickerplusmovie

Unlike the other two, the recording session *20180726_60MEA_left_33* contains a stimulus of the type *checkerflickerplusmovie* which is an extension of the *FrozenNoise* stimulus. It contains an additional movie repeatedly presented after the "frozen noise" presentations, that is, **after** `RunningFrames` and `FrozenFrames`, a fixed number of `MovieFrames` was presented. The movie stimulus is included in the repository in the file *movie0000.h5* represented as an array with the dimensions `screen_x` × `screen_y` × `num_frames`, containing contrast values between -1 and 1 for the screen dimensions for all movie frames. The movie consists of a still image of a gray scale natural scene from the DOVES database, shifted across the screen simulating eye fixations.

van der Linde, I., Rajashekar, U., Bovik, A.C., Cormack, L.K. (2009). „DOVES: A database of visual eye movements". *Spatial Vision, 22*(2): 161-177. URL: http://live.ece.utexas.edu/research/doves