**README**

1. System Requirements

* Windows 10 Enterprise, 22H2
* MatLab Mathworks R2020a
* intan Technologies RHD Recording Controller software 2.08
* Intan File Merger
* Intan NEX Converter (to convert proprietary .rdh files into .nex)
* Plexon Offline Sorter 4.5.0
* Anaconda Navigator 2.4.0
* Python 3.8.12
* Jupyter Lab 3.0.14
* Jupyter Notebook 6.3.0
* DAS 0.25.1
* ImageJ 1.54c
* Sound Analysis Pro 2011.104

1. Installation guide

* It is recommended to install the required softwares in the aforementioned order to prevent incompatibility and dependency-issues
* Installation instructions are provided by the individual standard software manufactures with the exception of DAS and Sound Analyses Pro 2011 (SAP 2011)
  + Installation guide DAS
    - GitHub: <https://github.com/janclemenslab/das>
    - Related publication: <https://elifesciences.org/articles/68837>
  + Installation guide SAP 2011
    - Manual: <http://soundanalysispro.com/manual>
    - Related publication: <https://www.sciencedirect.com/science/article/abs/pii/S0003347299914161>
* Estimated installation time depends on the individual machine the software is installed on but should not require more than 3hrs in total

1. Demo

* Instructions to run on data
  + Code segments
    - Opto\_lightstimulation is required to stimulate the designated area with a light-source during extracellular recordings. We used an Arduino Uno Rev3 to drive LED-Lightsources (see methods section)
    - readIntan is used to open recordings in MATLAB
    - Opto\_align-rec-light is required to visualise raw physiological data aligned to light stimulations
    - Opto\_dotraster-psth produces dotraster-plots for individually assigned units including peristimulus time histograms
    - electro\_gui is used to label song elements by hand and produce files such as “Bird1\_example-syllables\_AFTER”, input-data were raw audio recordings that contained the stimulus on- and offsets in a separate trace. This code was initially developed in the Long-Lab at NYU and permission to use has been obtained previously
    - Opto\_spectrogram is used to plot spectrograms of individually selected .wav files
    - Opto\_aligntonew aligns motifs to the onset of the first new syllable
    - Opto\_motif-params is necessary to calculate consistency for self and cross comparison
    - Opto\_accuracy is necessary to calculate accuracy values for self- and cross comparison
    - Opto\_sylchange is necessary to visualise motif count differences
    - Opto\_self-sim visualises similarities within vocalisations per bird as calculated by SAP 2011
    - Opto\_cross-sim visualises similarities across vocalisations per bird as calculated by SAP 2011
    - Opto\_kw-sim tests for differences between similarity scores
    - Opto\_transition is used to calculate transition probabilities between different syllables
  + Exemplary Data
    - Exemplary data for one behavioural stage of song recordings (after playback stage) are included
      * Raw song recordings (Folder: Bird1\_recordings)
      * Labelled syllables (Bird1\_example-syllables\_AFTER)
      * SAP similarity data (Bird1\_SAP\_similarity)
    - Exemplary data for sorted spikes of one in-vivo extracellular recording session of an anaesthetised ArchT-bird are included (BirdA\_example-extracellular\_spikes = sorted spikes from Plexon export; BirdA\_example\_extracellular\_ADC = light-pulses)
* Expected output
  + After running the attached code segments, exemplary figures and test results are provided in their raw format as they were used to compile the figures of the manuscript, final visual design of the figures was adjusted with Adobe Illustrator 2022
* Expected run time for demo
  + The runtime for all code snippets should not exceed 60s per snippet

1. Instructions for use

* Use the exemplary data when prompted by the code to do so, if code dependencies exist, it is mentioned in the documentation of each code snippet