How to use CellBase

Database

CellBase stores a hierarchical structure of three different entities: animals, sessions and cells. This hierarchy is reflected in the directory structure of CellBase as well as the animal, session and cell identifiers. Raw data and all derived files for each session are stored in the corresponding directories (see Directory Structure).

List of cell identifiers, analyses performed and user-defined information (analyses results and other properties) for each cell are stored in the three core variables of CellBase: CELLIDLIST, ANALYSES and TheMatrix. These variables can be accessed in Matlab via loadcb (see Core variables of CellBase).

New cells can be added or removed by using addcell, addnewcells and delcell functions (see also Database management). Cells or specific subgroups can be located by using findcell, findcellpos, findallcells, selectcell, tetrodepairs and nontetrodepairs. Information about certain cells or other elements of CellBase can be obtained with listcell, getvalue and listtag. Cell properties can be added or changed with insertdata and setvalue. Analyses and properties can be located by using findanalysis and findprop. They can be edited by addanalysis and delanalysis.

A number of conversion functions help switching between cell identifiers, file names or animal, session and tetrode tags: cellid2fnames, fname2cellid, cellid2tags and cellid2vals.

Data preprocessing

The first step of data analysis in CellBase is usually data preprocessing (see the list of functions aiding data preprocessing here). The raw data files are first converted to CellBase format.

TTL pulses recorded in Neuralynx systems are converted to an EVENTS.mat file storing TTL pulse times and corresponding event strings. This step is performed by converter files provided by Neuralynx.

Behavioral data is converted to a structure called ‘TrialEvents’. TrialEvents is a general format of CellBase, in which different fields store different behavioral variables, e.g. reaction time, response type, stimulus difficulty. Each field contains a numeric or cell array with length corresponding to the number of trials. The analysis programs provided in CellBase rely on this format. Hence, it is advisable to convert all behavioral data of any source into TrialEvents format. Per convention, trial start time stamps are saved as absolute times and all other time stamps are saved as relative times with respect to trial starts (all time stamps are in seconds). Some analysis functions rely on this property (see abs2reltimes and rel2abstimes for conversion between absolute and relative time stamps). See solo2trialevents4\_auditory\_gonogo for an example converter. If behavioral and neural data are collected using separate systems, the time stamps of the two systems should be synchronized for neuronal analysis with respect to behavior. In this case, the final TrialEvents file should reflect the synchronized time stamps. See MakeTrialEvents2\_gonogo for an example on synchronization.

In Cellbase there is an opportunity to store stimulation data separately. This option is extremely useful for optogenetic applications. The stimulation data is converted into a structure called ‘StimEvents’. This structure is similar to TrialEvents: it contains fields corresponding to stimulation parameters like stimulus onset time, duration, intensity, etc. Each field holds an array with length corresponding to the number of unitary stimulation events - e.g. light pulses. See MakeStimEvents2 for an example converter.

As a next step, neural data (spike times) are aligned to predefined events based on TrialEvents and StimEvents. A subset of the fields containing time stamps are chosen in event definition files like defineEventsEpochs\_gonogo and defineEventsEpochs\_laserstim. These functions are passed to prealignSpikes, which saves spike times relative to the defined events in STIMSPIKES and EVENTSPIKES files (see abs2reltimes and rel2abstimes for conversion between absolute and relative time stamps). This data preprocessing step speeds up most subsequent analyses.

Other data preprocessing functions include trial selection, segment selection, spike time or waveform extraction based on selected segments, etc. See a list of data preprocessing functions here.

See quickanalysis2 for an example on data preprocessing and basic analysis.

Data analysis

CellBase provides a set of data analyses and plotting functions. These include various types of peri-stimulus and peri-event time histograms, raster plots, spike clustering, waveform analysis, optogenetic tagging, auto- and cross-correlation functions. See a list of currently implemented analyses here. The modular structure of CellBase is meant to encourage users to expand the analysis functions with their own code (see also Programming in CellBase).