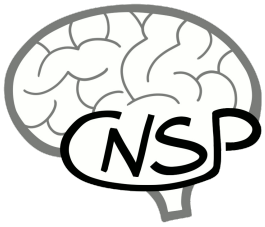

The Continuous-event Neural Data structure (CND) Specifications and guidelines

The CNSP Initiative

The CNSP initiative aims to develop and collect resources, such as analysis scripts and publicly available neural data, for the study of cognition and natural sensory perception. In doing so, we propose a standardised pipeline for recording, analysing, storing, sharing, and comparing datasets on sensory perception involving naturalistic tasks, such as listening to speech and watching a movie. In addition to featuring young researchers at the top of their respective fields of research and connecting scientists from a variety of disciplines (e.g., linguistics, psychology, computer science, engineering), the CNSP workshops provide guidelines and standardised practical and educational resources for analysing continuous-event neural data. Please visit our website at https://cnspworkshop.net and stay tuned!

What this document is about

Please find below the specifications for the Continuous-events Neural Data structure (CND). This document describes how a CND dataset should be organised, both in terms of folder structure and data structure. The guidelines can be then used to store your own data or to convert publicly available data for then availing of the CNSP resources with minimal or no changes to the analysis pipeline. These specifications will be regularly maintained at the link https://cnspworkshop.net/cndFormat.html.

What kinds of experiment designs are typically considered in the CNSP resources

The typical scenario considered here consists of neural signals (e.g., EEG, MEG) recorded as participants performed a natural listening task (e.g., speech listening). We will typically refer to this dataset, which is publicly available and whose CND data structure is available on the CNSP resources webpage. Other types of experiments that involve continuous sensory stimuli (music, artificial sounds) and other response measures (pupillometry, heart rate) are also possible. Interestingly, the CND data structure is also compatible with experiments involving various other continuous tasks, for example continuous motor movements. If you are planning to convert your own EEG/MEG/iEEG data, you can also refer to the BYOData preparation document. Please feel free to contact us, if you have any questions.

What is different between CND and existing data structures?

There exist standardised data structures that allow us to store a large variety of datasets from many technologies and with any experimental paradigm. However, the available solutions are either technology specific (e.g., formats for saving raw data) or general purpose (e.g., BIDS). The CND data structure is a step closer to data analysis, as it was designed to be immediately compatible with toolboxes in the area of natural sensory perception, such as the mTRF-Toolbox and the Eelbrain Toolkit. As such, the CND data structure works at a different layer than general purpose structures such as BIDS, as it is **domain-specific** (CNSP domain) and technology independent. Indeed, it is our intention to provide conversion scripts between CND and the most common general purpose data structures, providing the community with a rapid way to analyse and compare the increasingly large (yet heterogeneous) set of publicly available neural data in the domain of natural sensory processing.

The CND data structure

If you would like to use one of the CND datasets available on the website, please refer to this other guide.

The CND data structure provides guidelines for folder and data organisation, as well as providing a precise naming convention that should be adopted for guaranteeing the immediate re-use of resources, such as analysis scripts.

To follow along with our tutorials, we suggest you organise your dataset files according to the following folder structure:

```
--- Dataset folder (e.g., LalorNatSpeech)
       --- eeg
           - raw data file 1 (e.g., bdf, xdf)
           - raw data file 2
             ...
           - raw data file N
       --- stim
           - wavfile1.wav
             ...
           - midifile1.mid % or whatever other stimulus datafile is useful
             ...
       --- dataCND
            - dataSub1.mat
            - dataSub2.mat
               ...
            - dataSubN.mat
            - dataStim.mat
```

Crucially, your CND data must be stored into a folder named ‘dataCND’. Please make sure that you are using the exact naming convention discussed in this document for filenames, structure variables, and their fields. By doing so, the only change necessary for using the CNSP analysis scripts is to change the dataset path to *datasets/your_dataset/dataCND*. In this folder structure:

- Each *dataSub*.mat* file includes all neural data (e.g., EEG) from a given subject or session (no separate files for distinct trials/runs/blocks). The format of the neural data variables are given below.
- Subjects are indexed numerically (not nominally, and using 1, 2, etc, not 01, 02, etc). For guaranteeing anonymisation or appropriate pseudonymisation, the subject index should not reflect the order of the recording session.
- The *dataStim.mat* file includes stimulus features that were presented to subjects. The format of the stimulus features are given below.
    - Typically, participants listen to the same stimuli, either in the same order (e.g., audio-story) or in random order (e.g., several short stories). In that case, a single *dataStim.mat* should be provided. Trials have to be sorted in the same order in each *dataSub*.mat*, and a field indicating the original presentation order should be included. This is the case primarily covered during the workshop.
    - In some cases, different participants may hear different sets of stimuli. In such scenarios, it is necessary to store one *dataStim*.mat* file for each participant with the exact stimuli they heard. The CNSP tutorial scripts will need minor changes for dealing with this scenario (i.e., *dataStim*.mat* must be loaded for every subject, rather than just once for all of them.
- The neural data in the CND format should be synchronised to the stimulus. The sampling rate of neural data and stimulus have to be the same. I.e., sample *i* in dataSub*.mat must correspond to sample *i* in dataStim.mat.

Please see an example script that converts a publicly available dataset (the *LalorNaturalSpeech* dataset) to CND. These conversion scripts depend on the particular experiment, and this example might be useful for you to see how easy it is to prepare the CND structure.

In brief, there are two key types of data structures. A stimulus structure (*dataStim.mat*), whose key fields are **`stim.data`** and **`stim.fs`**; and a structure for the neural data (*dataSub*.mat*), whose key fields are **`eeg.data`** and **`eeg.fs`**, if this was EEG data. Indeed, one could have a CND data file for multiple recording modality on the same experiment (e.g., eeg, pupillometry, accelerometers). As you can note, the key fields are always **data** and **fs**. And the data should be syncrhonised and with the same sampling frequency, making the data directly compatible with toolboxes such as the mTRF-Toolbox. Please find below the details of these data structures.

dataStim.mat - This is a file containing all the preprocessed stimulus features for all trials. The file contains the variable **stim**, a structure with the following format:

```
stim: struct with fields:
      names: {1 x M cell}
            % cell array containing character array labels for each feature
            % set, M. E.g., {'speech envelope vectors','word onsets'}
      data: {M x N cell}
            % cell array containing the univariate or multivariate stimulus
            % feature vectors for each M feature(s) and N trials. E.g.,
            % two stimulus feature-sets (envelope, word onset) and twenty
            % trials. Each cell is a (timeSample x featDimension) matrix and
            % corresponds to one trial/run (e.g., 1 chapter
            % of an audiobook). Different trials can have different
            % lengths.
      stimIdxs: [1 x N array]
            % Stimulus indexes. This is important as, for example, the same
            % stimulus may have been repeated (not in this particular
            % dataset). Note that each recording session may have presented
            % these stimuli in a different order. See each 'eeg' structure
            % for information on the original presentation order.
      condIdxs: [1 x N array]
            % Condition index corresponding to each data trial. In our natural
            % listening example, condIdxs = ones(1,20);
      condNames: {1 x P cell}
            % Condition names e.g., {'Listening'}
      fs: scalar
            % sampling frequency in Hertz e.g., 128
```

You are free to save any additional fields that may be useful for you (e.g., you may want to save labels indicating what the featDimension in the 'data' field corresponds to, such as frequency-bands or phonemes). We suggest calling such fields 'additional information'. But make sure to have at least the key fields: **'data'** and **'fs**'.

In **stim.data** has dimension M x N, where *M* corresponds to different feature-sets (i.e., "models") and *N* to the number of trials. Feature-sets can be comprised of one or more univariate (e.g., envelope, word onsets) and/or multivariate (e.g., phonemes, spectrogram) features. Each element of stim.data has dimension *timeSamples* x *featDimension*, where *featDimension* refers to the number of those dimensionality of each feature set (i.e., how many features in the set).

For example, let's consider **stim.data: 4 x 20**, which describes 4 stimulus feature-sets over 20 trials. Each trial could have a speech envelope as first feature-set (*featDimension=1*), a 16-band speech spectrogram as second feature-set (*featDimension=16*), the labeled occurrences of 19 phonetic features as third feature-set (*featDimension=19*), and the joint model of spectrogram and phonetic features as fourth feature-set (*featDimension=35*).

dataSub1.mat, …, dataSubN.mat - These are files containing the neural signals recorded during the presentation of the stimuli in *dataStim.mat*. Each file contains one variable per recording modality. In our case, we have only one variable, **eeg**. Note that we could potentially have

another variable called, for example, **pupilDilation** with fields **pupilDilation.data** and **pupilDilation.fs**, or **meg.**

```
eeg: struct with fields:
      dataType: character array
            % type of response measure, e.g., 'EEG'
      deviceName: character array
            % Name of the recording system (if available), e.g., 'BioSemi'
      fs: scalar
            % sampling frequency in Hertz
      data: {1 × N cell}
      % cell array. Each cell is a (timeSample x channels) matrix.
            % Ideally, one trial/run per cell (e.g., 1 chapter of an
            % audio-book). Different trials can have different lengths, as
            % long as they match their corresponding stimulus features. Trials
            % are sorted so that CND data from all recording sessions use the
            % same stimulus order (e.g., 1:20). The field 'origTrialPosition'
            % indicates the position of a given EEG segment (trial) in the
            % presentation order of the original experiment, in case it is
            % relevant for the analysis. In this experiment, trials were
            % presented in order (audio-book). If the experiment had 2
            % conditions (e.g., listening vs. imagery), with 20 trials each,
            % then 'data' would have dimensionality {1x40 cell} and we would
            % need to include the condition indices of each trial in an
            % additional field stim.'condIdxs'.
      origTrialPosition: [1 x N array]
            % eeg.data cells have a 1-to-1 correspondence with the stim.data
            % cells. This means that the eeg.data{i} corresponds to
            % stim.data{i} and that data from all subjects is sorted in the
            % same way. However, the actual stimulus presentation order in the
            % experiment may have been different. This field is used to
            % remember the stimulus presentation order and indicates the
            % original position of each trial in that order. Handy suggestion.
            % To recover the EEG data in the original presentation order use:
            % clear X; X(eeg.origTrialPosition) = eeg.data;
      chanlocs: [1 × C struct]
            % Channel location variable (EEGLAB format)with C c
      extChan: {[1 × 1 struct]}
            % External channels. For example, mastoid or EOG. Each
            % field of this structure is a cell array like data (one field for
            % each type of external channel).
```

Please get in touch if you would like to share your data. We will be happy to share the link to your data on the CNSP website.

Last update: 8 July 2022

Giovanni Di Liberto and Aaron Nidiffer