A standardised open science framework for sharing and re-analysing neural data acquired to continuous sensory stimuli

Giovanni M. Di Liberto[1,*,**], Aaron Nidiffer[2], Michael J. Crosse[3,4], Nathaniel Zuk[5], Stephanie Haro[6,7], Giorgia Cantisani[8,1], Martin M. Winchester[1], Aoife Igoe[1], Ross McCrann[1], Satwik Chandra[1], Edmund C. Lalor[2], Giacomo Baruzzo[9].

1 School of Computer Science and Statistics, University of Dublin, Trinity College, Ireland; ADAPT Centre, Trinity College Institute of Neuroscience
2 Dept. Biomedical Engineering, Dept. of Neuroscience, Del Monte Institute for Neuroscience, Center for Visual Science, University of Rochester, Rochester, NY, USA
3 Segotia, Galway, Ireland
4 Department of Mechanical, Manufacturing and Biomedical Engineering, Trinity Centre for Biomedical Engineering, Trinity College Dublin, Dublin, Ireland
5 Department of Psychology, Nottingham Trent University, Nottingham, UK
6 Human Health and Performance Systems, MIT Lincoln Laboratory,Lexington, Massachusetts, USA
7 Speech and Hearing Bioscience and Technology, Harvard Medical School,Boston, Massachusetts, USA
8 Laboratoire des systémes perceptifs, Département d'études cognitives, École normale supérieure, PSL University, CNRS, 75005 Paris, France
9 Department of Information Engineering, University of Padova, Padova, Italy

*Corresponding author: diliberg@tcd.ie
**First and senior author

Conflicts of interest: none declared.

Word count abstract: 216.

Word count (excluding abstract, title page, and references): 9080.

Abstract

Neurophysiology research has demonstrated that it is possible and valuable to investigate sensory processing in the context of scenarios involving continuous sensory streams, such as speech and music listening. Over the past 10 years or so, novel analytic frameworks for analysing the neural processing of continuous sensory streams combined with the growing participation in data sharing has led to a surge of publicly available datasets involving continuous sensory experiments. However, open science efforts in this domain of research remain scattered, lacking a cohesive set of guidelines. As a result, numerous data formats and analysis toolkits are available, with limited or no compatibility between studies. **This paper presents an end-to-end open science framework for the storage, analysis, sharing, and re-analysis of neural data recorded during continuous sensory experiments.** The framework has been designed to interface easily with existing toolboxes (e.g., EelBrain, NapLib, MNE, mTRF-Toolbox). We present guidelines by taking both the user view (how to load and rapidly re-analyse existing data) and the experimenter view (how to store, analyse, and share). Additionally, we introduce a web-based data browser that enables the effortless replication of published results and data re-analysis. In doing so, we aim to facilitate data sharing and promote transparent research practices, while also making the process as straightforward and accessible as possible for all users.

Introduction

In our daily life, we navigate complex sensory environments containing overlapping streams of information, such as auditory, visual, and somatosensory signals. The human brain processes these signals, while integrating and interpreting them based on our prior knowledge and expectations, and it does so in real time[1]. Decades of impactful discoveries have shed light on the neural architecture of sensory processing one piece at a time through ingenious, well-controlled, laboratory-based experiments[2-4]. The extensive literature on sensory processing with such controlled methods led to increased interest in testing whether these findings generalise to more ecologically-valid tasks involving sensory signals that occur over long (> 1 second) periods of time – what we refer to as *continuous* experiments[5,6]. These continuous experiments are designed and analysed differently from protocols that rely the presentation of more discrete stimuli. Extending the time course of the experiment expands the kinds of questions that can be addressed. For example, would the findings on the neural processing of isolated syllables and words apply to natural speech?[7] Furthermore, tasks involving continuous stimuli enable the study of neural processes that would be otherwise inaccessible, such as the connection between sound statistics and music enjoyment.

There is a growing body of studies involving continuous speech and music[8,9], with data and analysis code being shared more frequently within the community[10-13]. However, in contrast to scientific fields with a more evolved open-science framework (for example, bioinformatics – **Box 1**), clear open-science guidelines and tools are missing when working with continuous sensory experiments, making the current literature fragmented into lab-specific procedures for data storage, analysis, and sharing. We aim to improve the degree to which results can complement each other by suggesting a framework for continuous human auditory neurophysiology experiments. This paper presents **a cohesive end-to-end open science framework, with user-friendly guidelines and tools for the storage, analysis, and sharing of continuous sensory neural data and analysis code**. We also demonstrate how this framework enables immediate access to existing datasets, offering libraries and tools for the rapid replication of published results, re-analysis (e.g., power analysis) with different configurations, and hypothesis formulation through a new data simulation toolkit. Finally, we present a first-of-its-kind web-based data browser that enables effortless replication of results and data re-analysis.

Investigating sensory processing with continuous sensory stimuli.

In neurophysiology, the nature of the sensory stimuli influences the experience of the listener and what sensory processes are recruited in the experiment. Sensory-based experimental protocols can be categorised into two groups based on their presentation design. Sensory inputs can be presented *continuously* over a long stretch of time or *discretely* over the course of many short trials (**Figure 1**). '*Continuous sensory experiments'* refers to tasks involving uninterrupted auditory, visual, or tactile

sensory streams[14,15,11,16,17-19,20]. By contrast, *discrete* sensory experiments involve presentation streams that are based on brief, separate items, such as oddball paradigms involving sequences of isolated syllables. Music and speech streams are considered continuous, despite being composed of a sequence of well-defined individual items (i.e., music notes). For example, in monophonic melodies, the listener does not perceive the separate disconnected notes, but instead experiences a continuous melody where each note is meaningfully placed within a broader context. It should be noted that one can extract discrete features from continuously presented sensory streams. Some experiments are continuous in sensory presentation but are designed to investigate both continuous and discrete features embedded in continuous streams (e.g., the sentence onsets of continuous speech; switching of attention in multi-talker scenarios) (**Figure 1**). **In this work, *continuous sensory experiment* refers to the temporal nature of the sensory experience.** The analysis that follows may involve the examination of continuous, discrete, or mixed features, providing us with experiments where a variety of methodologies can be utilised to study more ecologically relevant research questions.

The temporal nature of the sensory experience (i.e., discrete vs. continuous) can change how the brain processes the input[7] and change the analyses techniques used to characterise these processes. In experiments involving discrete events, the only pieces of stimulus-related information necessary are the *timestamps* indicating the start of each event (e.g., syllable onsets, button-click) and the *categorical labels* indicating how to classify that event (e.g., frequent vs. infrequent, syllable identity). This information can be used to conduct the well-known Event-Related Potential/Field analysis (ERP/ERF). Timestamps are used to segment the electroencephalography, electrocorticography, or magnetoencephalography data (EEG, ECoG, and MEG, respectively) and the categorical labels are used to group epochs according to the labels. Conversely, the absence of carefully designed discrete events makes continuous sensory experiments less suitable for ERP analyses (although some previous work attempted to run ERP analyses on EEG responses to continuous speech[21]), and instead system identification analyses involving the estimation of the input-output relationship are more appropriate. This type of analysis has typically been conducted using continuously varying properties of the stimulus, such as the speech envelope, but it can also be carried out using event timestamps, which are represented using a binary mask i.e., sequences of zeros and ones, where the latter indicate the onset or the entire occurrence of a given event (e.g., word onset). Note that some properties are represented in a manner that is discrete in time but continuous in magnitude, such as semantic dissimilarity and lexical surprise[22-24].

4
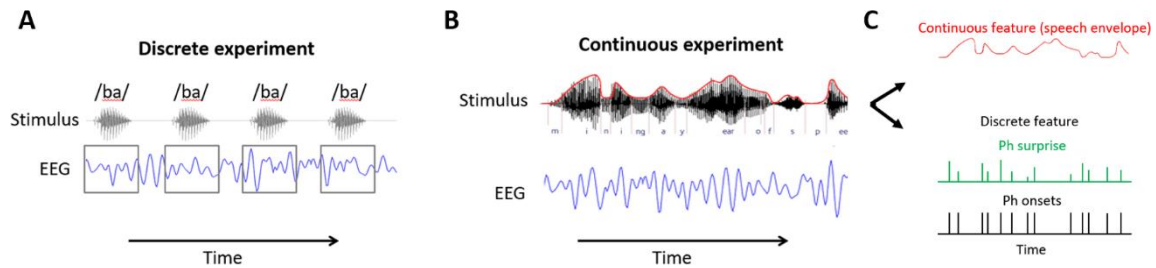
Figure 1: Investigating sensory processing with discrete versus continuous events and features. (A) *An EEG experiment involving listening to individual syllables that are presented discretely. The discrete syllable timestamps can be used to epoch the data.* ***(B)*** *An EEG experiment involving natural speech listening. The stimulus was continuous speech, hence there is no isolated, discrete event in the traditional experimental sense that can be used to epoch the data.* ***(C)*** *A continuous experimental stream is rich in information and can be described using both continuous and discrete features. Some stimulus features are encoded continuously by the participant so the feature of interest is continuous in time and value, such as the sound envelope (top panel). Events of interest in the continuous stream can be described using discrete timestamps. For example, phonemic surprisal describes points in time where surprisal differs during continuous speech listening (middle panel). Additionally, a binary mask can be used to indicate discrete events over the continuous experiment, such as phoneme onsets (third panel).*

When considering tasks such as speech and music listening, continuous stimuli enable the design of more ecologically-valid experiments. But **why is it so important to study sensory processing in ecologically-valid scenarios?** One reason is to test whether findings from artificially controlled paradigms mirror brain processing during natural, real-world conditions. Secondly, more realistic scenarios can increase participant comfort and engagement, allowing testing of individuals who typically struggle with traditional experiments (e.g., children, individuals with neurocognitive deficits). Thirdly, continuous sensory tasks – such as speech listening – engage many neural processes, allowing us to assess them and their interaction simultaneously, rather than studying those processes in isolation (e.g., syllables in the context of a sentence vs. a syllable oddball paradigm). Finally, there is one additional advantage that does not necessarily impact the experimenter, but rather the research community at large. Namely, unlike controlled experiments, which are designed with specific objectives and analysis methods in mind, more naturalistic neurophysiology experiments have the benefit of producing data that can be shared for purposes that extend beyond results replication. In fact, it is possible to re-analyse the data with objectives and methods that the experimenter may not have initially considered, which can be particularly useful for hypothesis design and preliminary results generation as well as for methodological development. Indeed, **a word of caution** is necessary here, as this new possibility comes with risks such as "*p*-hacking" or, similarly, "fishing" for results. Nevertheless, the tools for mitigating such risks exist and should be adopted, such as the careful use of cross-validation within and between datasets, and by replicating the results on a new dataset.
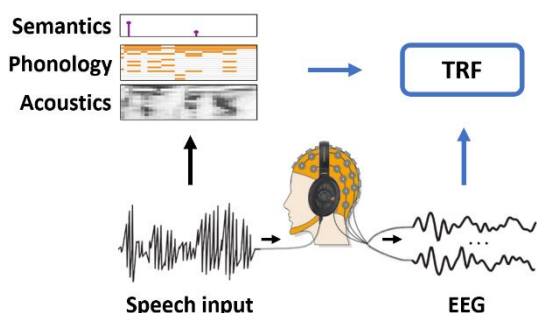
Figure 2: The multivariate temporal response function (TRF) framework. *Neural signals are recorded as participants are presented with sensory stimuli, such as speech. A specific task may be part of the experiment (e.g., answering comprehension questions). Speech and language features are extracted from the stimulus and are simultaneously related to the neural signal with methods such as lagged linear regression. The weights of the model inform us about the spatio-temporal relationship between each stimulus feature and the neural signal.*

Processing neural data acquired to continuous sensory stimuli.

The past few years have seen a surge in the use of ecologically-valid tasks, largely due to the development of recent methods for neural data analysis that are suitable for continuous sensory events. This report primarily focuses on a particular analytical framework, the Temporal Response Function (TRF), which captures how neural signals react to changes in a sensory input (**Figure 2**). The TRF is a quantitative estimate of the stimulus-response relationship, assuming that such a relationship exists, that it is approximately linear and measurable, and that it is reasonably consistent over time (i.e., time-invariant). TRF analyses rapidly gained popularity primarily due to the fact they can be used to study how our brains track, encode, and build expectations of continuous stimuli, such as speech and music. For example, the envelope TRF estimates the relationship between the sound envelope, a key property for speech processing[25,26], and from the neural response as measured using EEG/MEG for example. For implementation, we refer to the mTRF-Toolbox[23], a library for estimating multivariate TRFs based on regularised time-lagged regression. Please see Crosse et al.,[27,28] for detailed information on the multivariate TRF methodology and interpretation

and Obleser and Kayser[29] for a perspective on the neural mechanisms that may generate speech TRF results. Of course, alternative approaches exist for relating neural data to ongoing stimuli, such as Canonical Correlation Analysis (CCA)[30], or even other types of recording, such as human and non-human intracranial EEG, which will not be discussed here for simplicity. More recently, the substantial leap in machine learning methodologies, especially deep learning models (e.g., GPT-4), have further increased interest in studying ecologically-valid stimuli, by providing us with new ways to investigate continuous sensory processing. In particular, such large language models can be related to neural data directly by considering the weights in their hidden layers[31], or they can be used to estimate specific linguistic aspects of a speech stream, such as lexical surprisal, which can then be related to neural responses with TRF methods[24].
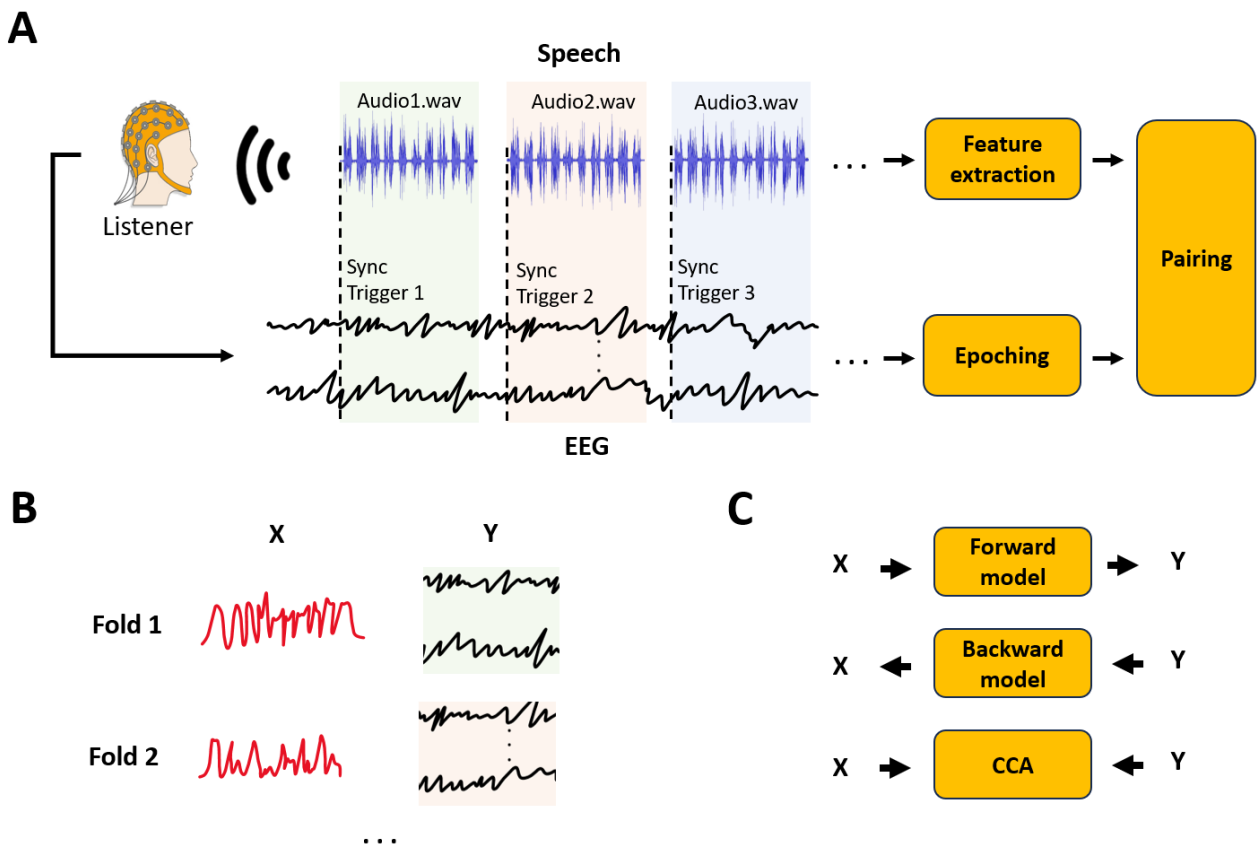
Figure 3: Continuous experiment neural data acquisition and analysis pipeline. The figure focuses on the typical continuous speech listening scenario. (A) EEG/MEG is recorded as the participant listens to speech segments. Synchronisation triggers are used to epoch data into trials of continuous speech, in contrast to being used to epoch data around discrete stimulus tokens. The value of each trigger corresponds to the index of the audio file (e.g., 1: audio1.wav, 2: audio2.wav). Stimulus features (e.g., sound envelope) are extracted for every audio-file and paired with each corresponding EEG/MEG epoch. (B) Neural data and stimulus feature pairs are organised into data structures, X and Y, that are time synchronized. Trials of continuous responses can be used as the folds for a leave-one-fold-out cross-validation. (C) X and Y can be used to investigate the EEG/MEG encoding of the input features of interest using a forward model or multivariate Temporal Response Function (mTRF)[32] approach. Conversely, X and Y can be used to build decoding models or backward models[33]. While mTRF-based forward and backward models are limited to multivariate-to-univariate mappings, relationships where both X and Y are multivariate can be studied with methods like canonical-correlation analysis(CCA)[30].

Given the importance and power of experiments based on continuous, naturalistic stimuli, there has been a marked increase in the number of available datasets recorded using such stimuli. However, these datasets have been severely underexploited given the lack of any coherent set of data storage and sharing protocols across the community. This has meant that data are stored in idiosyncratic ways particular to each research team, or even inconsistent within the same team – making it inefficient and sometimes impossible for other researchers to use those data to answer new questions. Nonetheless, while these inconsistencies extend to aspects such as data formatting, naming of variables and files, and the fine details of the analysis, there exist strong procedural similarities that can be codified. **Figure 3** attempts to depict these consistent procedural steps by considering a widely used and particularly simple scenario involving listening to natural speech segments (e.g., chapters of an audio-book). In that case, the first part consists of a) extracting the features of interest from the stimulus (e.g., speech envelope); b) segmenting the neural data (epoching); and c) resampling and aligning the speech features with the corresponding neural segments (pairing). The resulting stimulus and neural segments are precisely synchronised (same start sample and number of samples). The combination of all stimulus and neural signal segments, which we refer to as X and Y respectively, can then be used to fit input-output models, such as TRFs and CCA. This manuscript also discusses how the typical key steps to obtain X and Y, which are depicted in **Figure 3**, can also be adapted to scenarios involving multiple experimental tasks or conditions (e.g., listening vs. imagery).

The proposed end-to-end framework.

Here we present the first end-to-end specifications and resources for the analysis of neural data acquired to continuous sensory stimuli, including data storage specifications, standardised datasets, learning resources, and analysis tools. One of the key motivations of having a cohesive procedure is to simplify and expedite all operations from data storage to analysis. To this end, **we define a new domain-specific data structure, called Continuous-event Neural Data (CND)**, offering well-defined how-to guidelines on storing new data. Doing that will enable the **immediate use of all resources produced by this project, from preprocessing scripts, analysis scripts and GUIs for running TRFs and CCA analyses, as well as a simulation toolkit**. CND can be loaded directly by other widely used toolkits such as NapLib and EelBrain. Furthermore, we provide import/export functions connecting CND with general-purpose toolkits relevant to neurophysiology and neuroimaging research at large (e.g., MNE, EEGLAB). This work has been carried out by an interdisciplinary, international team that was initially assembled in 2020 with the goal of propelling open science and thus supporting basic and translational research in this domain. The resulting open science initiative, named Cognition and Natural Sensory Processing (CNSP), is an ongoing international research collaboration with headquarters in Trinity College Dublin.
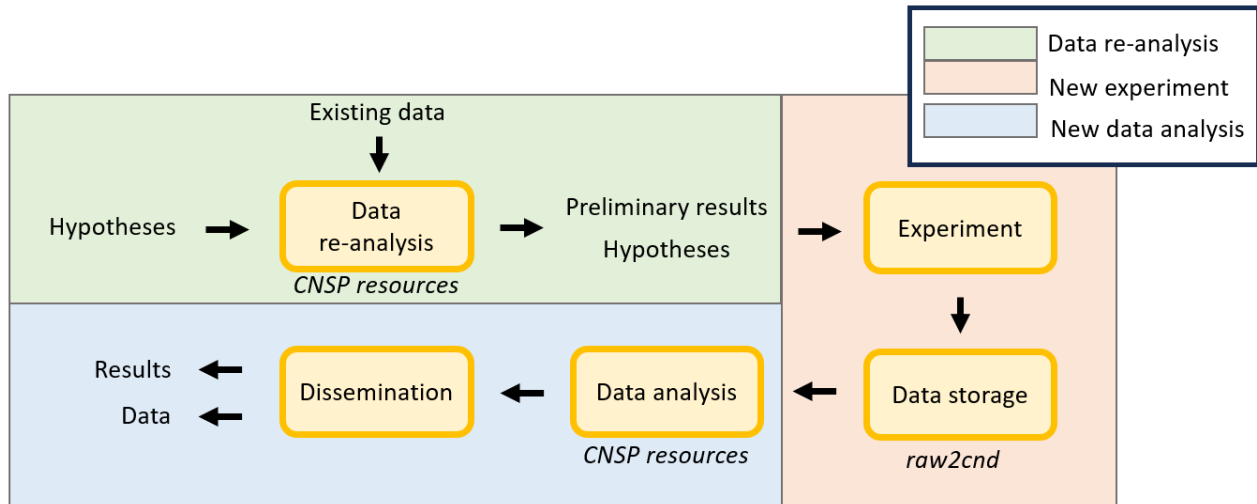
Figure 4: An end-to-end experimental pipeline for hypothesis formulation and testing. *Hypotheses are preliminarily tested and refined by re-analysing existing data. This step can either be carried out by using a publicly available dataset that is similar to the target scenario, or via data simulation. The refined hypotheses are then tested on a new experiment. Data stored in a standardised format can then be analysed with the same procedure used with the preliminary data. The proposed standardised data structure connects these steps into a cohesive end-to-end pipeline, enabling the immediate use of the resources in this project, such as analysis tools and simulation toolkit.*

This report is organised by considering the view of an experimenter that is conducting a new neurophysiology investigation from scratch and intends to formulate precise hypotheses supported by existing data, and then run a new experiment (**Figure 4**). First, we present the CND standard for data storage (**Storage and sharing**). Second, we detail a standardised analysis pipeline for neurophysiology data analysis, from minimal preprocessing, TRF, CCA, and cross-correlation analyses, including libraries and learning resources such as video-tutorials (**Analysis pipeline**). Third, a new data browser is presented that allows for rapid result replication and re-analysis of previous datasets, as well as the performance of comparisons between datasets, models, and methodologies; all with a user-friendly environment (**Data browser GUI**). Next, we present a simulation toolkit and describe its value for experimental design and hypothesis formulation (**Simulating**). Finally, we discuss the limitations of the current framework, ongoing work, and future directions, including pointing at areas of development where the research community is encouraged to contribute (**Limitations and future directions**).

Storage and sharing

Sharing data and analysis scripts is key for advancing science and promoting transparency in research. However, the lack of clear guidelines can lead to a frustratingly heterogeneous set of publicly-available resources. Shared datasets and scripts in the field of auditory neuroscience are often difficult to use without additional help from the authors. In some cases, the authors may no longer be available or may not remember how to run their own code, which can further complicate the process. On the other hand, overly rigid guidelines for data sharing can also be problematic, as they can make the sharing process overcomplicated and become counterproductive. The framework described here aims to strike a balance between these two extremes, providing guidelines that are specific enough to be useful but not so complex as to be burdensome. Unlike *general-purpose* tools such as EEGLAB and MNE, our objectives are *domain-specific* as they target continuous sensory experiments only and are built by focusing on the growing audio and audio-visual literature on speech and music perception. This allowed us to design guidelines that keep the methods as straightforward as possible for both the experimenters (who save their data) and future users (who load and re-analyse the data).

Firstly, **experimenters should ensure that they have the right to share all data, including the original stimulus files (e.g., audio files); and this should be verified before data collection**, at the experimental design stage. This trivial yet sometimes neglected point is key to avoiding a reliance on the "available upon request" statement, which translates to multiple time-consuming direct interactions between users and those who shared the data. Secondly, it is extremely important that experimenters adopt a standardised data structure. Here, standardised refers to the presence of a clear protocol on data types, folder structure, and naming, with documentation and how-to guidelines. Crucially, it also indicates the existence of import/export scripts allowing for the effortless translation of the stored data from one standardised data structure to another. As such, the

experimenter can adopt the data structure that best fits their project and expertise, and then export to a different data structure when needed.

The data structure that is most widely adopted for data sharing in neuroimaging and, more recently, neurophysiology is the Brain Imaging Data Structure, or BIDS. This data structure is general-purpose, as it can be used to store virtually any kind of experiment involving fMRI, EEG, MEG, and other recording modalities. BIDS and its extensions are also compatible with widely used toolboxes such as EEGLAB (through a specific plugin) and MNE (via the extension MNE-BIDS), supporting science by providing tested analysis scripts and pipelines, reducing the risk for analysis errors due to bugs in the code for example. To fulfil the goal of being general-purpose, BIDS focuses on storing raw data, while it lacks guidelines and constraints that may be necessary in particular domains of research. The input-output models used in continuous sensory neurophysiology require the extraction of stimulus features and the alignment of stimulus and neural signals, as depicted in **Figure 3**. However, the lack of domain-specific detailed specifications on how experimenters should store their stimuli (e.g., raw audio files or midi files, sound envelopes, note onsets), led to disparate interpretations that have to be addressed in each specific toolbox and custom script.

While BIDS can indeed be used for saving continuous sensory neurophysiology data, the literature lacks clarity on how that should be done. Here, we define a new domain-specific data structure that is tailored to continuous sensory experiments, called the Continuous-event Neural Data structure (CND). This data structure provides a more tailored approach for data storage and sharing, greatly simplifying result replication and re-analysis, as input-output models such as TRF and CCA can be run directly on the CND structure. The domain-specific constraints and guidelines ensure consistency across datasets, as well as simplifying the data storage procedure. Crucially, we connected CND and BIDS via specific import/export functions, enabling the use of general-purpose toolkits when useful. Next, we present a summary of the CND data structure. Please also refer to the CNSP website (https://cnspinitiative.net/) and CNSP documentation (https://cnsp-resources.readthedocs.io/) for more detailed specifications.

Continuous-event Neural Data (CND)

CND is a domain specific data structure tailored to continuous sensory neurophysiology experiments. In contrast to BIDS, the data structure was designed to feed directly into input-output analysis methods, such as encoding models (e.g., TRFs), decoding models, and CCA. To so do, CND stores input (e.g., speech features) and output (e.g., neural signal) as numerical matrices, already epoched (i.e., segmented) and synchronised (i.e., the first sample of each epoch corresponds to the same time-stamp in both input and output). In addition to the raw stimulus files, CND must include a data structure with the stimulus features (e.g., the sound envelope). This way the features that are key in a given study are already included and are readily-available for analysis,

while future users will additionally have access to the raw data for their preferred preprocessing pipeline. **Figure 5A** depicts the folder structure, with folder and file names that should be used exactly as reported here. This will ensure compatibility with CNSP resources such as tutorials, Data Browser, and Simulation toolkit.

A main folder structure is provided containing all the experimental files, organised into 'code', 'tutorials', 'datasets', and 'libs' (libraries) folders. For simplicity, we advise users to have one main folder for each study. If multiple experiments are being considered, such as when using the Data Browser, it is also possible to include many datasets in the 'datasets' folder. Each dataset includes a 'dataCND' folder, containing as many 'dataStimX.mat' (stimulus files) and 'dataSubX.mat' (neural data) files as the number of participants. If all participants were exposed to identical stimuli, then a single 'dataStim.mat' file can be used. In that case, stimuli might have been presented in different orders for distinct participants (e.g., [1,2,3,…], [2,3,1,…]) and, as such, the neural data segments in 'dataSubX.mat' will have to be sorted to match the order of the single stimulus file. The 'rawStimuli' and 'rawNeural' data folders contain the unstructured original data and custom processing code (e.g., *raw2cnd*, *feature extraction* in **Fig. 2A**). Both folders are optional but recommended. For example, in the case of relatively large datasets (>10 GB), we suggest experimenters share two versions of the data, with and without the original raw files. A how-to guide is provided on converting raw data into CND (see *bids2cnd* function).
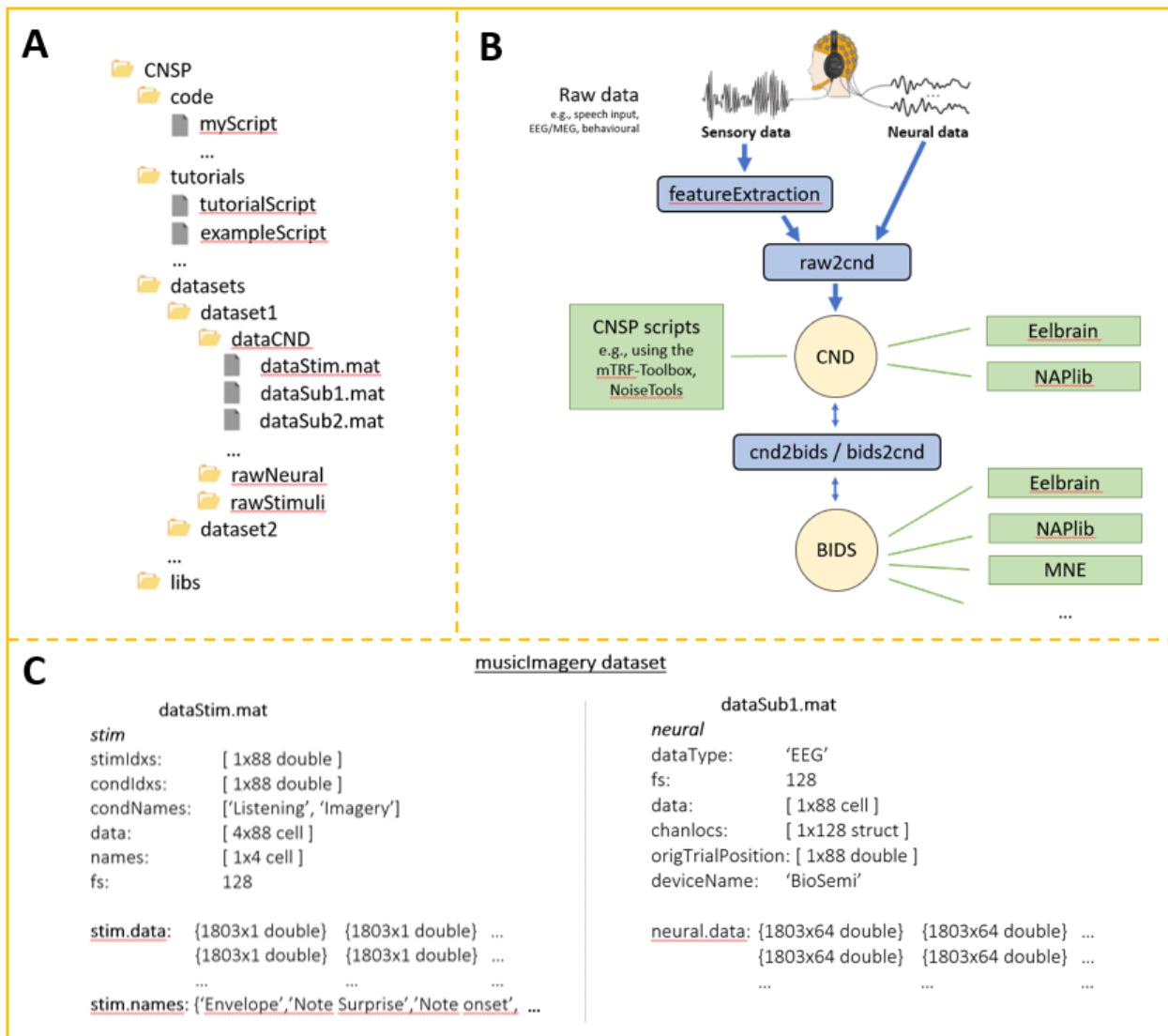
Figure 5. The Continuous-event Neural Data (CND) structure. (A) *CND folder structure.*
'dataStim.mat' refers to the stimulus features, while 'dataSubX.mat' refers to the neural data. In the
example, all participants were presented with the same stimuli. It is also possible to associate a
'dataStim' file to each participant if needed e.g., dataStim1.mat, dataStim2.mat. ***(B)*** *Conceptual data*
preprocessing and analysis pipeline. In addition to the CNSP resources, Eelbrain-toolkit and NAPLib
can also process CND data structures. It is also possible to export the CND data structure into Brain
Imaging Data Structure (BIDS), enabling the use of general-purpose toolboxes such as MNE-Python.
(C) *Example of dataStim and dataSub structures. dataStim contains the stimulus data matrices*
(stim.data) for each of the four stimulus features and for each of the 88 experimental trials. 'names'
specified what the four stimulus features are i.e., envelope, note expectation, note onset,
metronome. 'stimIdxs' indicate the stimulus index of each trial. The 'rawStimuli' folder should include
documentation regarding what stimulus file corresponds to each index. Here, indices 1-4 refer to
pieces named 'chor-096', 'chor-038', 'chor-101', and 'chor-019' respectively. 'condIdxs' refers to the
task that was carried out in each trial i.e., music listening or imagery. The EEG data structure is
straightforward in that it simply contains basic meta-data, such as the 'deviceName', and a data
matrix (neural.data), with a [samples x channel] data matrix per trial.

Interoperability: Interconnecting CND with existing resources

In the last decade or so, the importance of adopting open science strategies in our field has progressively become clearer. As a result, several lab-specific toolboxes have been released, each based on custom code written by individual scientists for their own work (e.g., NoiseTools[34], mTRF-Toolbox[35,36], EelBrain-toolkit[37], NapLib[38]). In some cases, distinct toolkits run similar (but not identical) analyses. Therefore, they cannot be easily compared because they are not directly compatible. Interconnecting these tools can be time-consuming for scientists skilled in computer programming and inaccessible to others. As such, one key output of the present work is that we have established an infrastructure to connect such resources. Specifically, our approach consists of replacing lab-specific custom data structures with CND, while BIDS serves as a go between, connecting this work with general-purpose analyses resources such as MNE-Python (**Fig. 2C**)[39]. Finally, domain-specific toolboxes such as EelBrain and NapLib have also been extended by their respective authors to support the CND data structure.

Storing raw data in CND format: practical considerations

We provide guidelines, a video-tutorial, as well as simple MATLAB/Octave script (*bdf2cnd.m*) for converting EEG BioSemi datasets into CND. Similar considerations apply for data from other devices or recording modalities. Note that the pipeline is experiment and device specific, so the script will have to be modified to fit the specific neurophysiology recording device and synchronisation trigger protocol. Indeed, we encourage researchers to contribute by sharing a conversion script for other devices and technologies.

Here, we discuss a *bdf2cnd* conversion for a scenario where EEG signals were recorded during speech listening, with audio segments presented in a random order (different order for distinct participants). The synchronisation protocol is simple, as it involves only one type of trigger indicating the start of an audio segment (as in **Figure 3**). The code of that trigger corresponds to the index of the audio file (e.g., *audio1.wav*). To save a new dataset into CND, the first step is to create a folder structure as in **Figure 5A**. We strongly recommend the naming of files as 'audio1.wav', 'audio2.wav', and so on, while also including the original source of those stimuli or corresponding conditions in the documentation, but not in the filename. The simplest way to build a CND dataset is to create a 'dataStim' and 'dataSub' file for each participant. 'dataSub' contains a structure with the epoched neural data and meta-data, such as the sampling frequency ('fs') and the channel location information ('*chanlocs*'; see **Figure 5B**). The epoching simply consists of chunking the neural data into segments, where each segment paired and synchronised with a specific audio input. This data matrix will correspond to the stimulus data matrix in 'dataStim', which will have the same number of trials and start samples. Note that the CND files can contain raw epoched EEG data, which may be at a different sampling rate than the stimuli. That will necessarily be corrected at the preprocessing

stage, where it is ensured that the same sampling rate and number of samples are present in the stimulus and neural data. Stimulus features are stored as data vectors (when univariate) or data matrices (when multivariate), such as the sound envelope (*timeSamples* x 1) and spectrogram (*timeSamples* x *numberOfBands*) respectively. Rather than adopting a sparse representation, CND requires a vector/matrix representation preserving all datapoints, including the zeroes, which is necessary for toolboxes such as the mTRF-Toolbox and NoiseTools. A full list of specifications is available on https://cnsp-resources.readthedocs.io, with example scripts and video-tutorials.

Analysis pipeline

One of the core goals of the CNSP Initiative is to facilitate analysis, sharing, and reanalysis of continuous sensory event neural data. To this end, a library of scripts has been shared with the CNSP community as part of a workshop series. These scripts, which include preprocessing, analysis, and plotting routines as well as numerous low-level support functions (e.g., filtering, down-sampling) were intended to serve as a blueprint for standardising future analysis pipelines. A standardised approach allows for easy sharing, reanalysis, and comparison across datasets and methodologies. We don't see these scripts as a finished product. Instead, we encourage users to contribute additions or corrections through the CNSP repository on GitHub (https://github.com/CNSP-Workshop/CNSP-resources)[40]. These scripts were written to require minimal or no customisation when analysing a new dataset in CND format. As such, they allow for easy interfacing with many toolboxes (such as the mTRF-Toolbox) used for the analysis of continuous neural signals. This is possible by restricting the domain of interest to continuous sensory perception scenarios, or other scenarios that can be coded similarly. For example, although not optimal, a typical mismatch-negativity scenario could be stored and analysed according to these same guidelines.

Preprocessing

One benefit of analysing continuous neural data is that analysis often requires minimal preprocessing if the dataset is not excessively noisy. For a typical experiment, preprocessing involves at least a high-pass filter or detrending step to remove potential drift, down-sampling data to a more manageable size (but be wary of anti-aliasing filters in functions such as MATLAB's *resample*), and epoching the neural recording into manageable segments[28]. These preprocessing steps are included in the shared CNSP libraries and example scripts. Minimalism in preprocessing has added benefits with respect to open science policies. Firstly, secondary users have access to potentially informative aspects of the data that might have been removed or altered by extra preprocessing steps (e.g., dimensionality reduction, filtering) and which are spared from potential sources of contamination (e.g., filtering[41]). And secondly, the minimalist, natural structure of many continuous-events stimulus

paradigms (e.g., participant listens to an excerpt of narrative speech) makes replication of findings across different datasets highly convenient so long as there is some consistency in preprocessing. Indeed, when sharing minimally preprocessed data, users have maximum freedom to make choices and include additional preprocessing steps to suit the needs of their analysis pipeline. At the very least, preprocessing steps should be documented so that the secondary user can be aware of what has been done to the data. The CND specifications allow space to document preprocessing steps and the CNSP scripts implement this functionality. Finally, in addition to the minimally preprocessed data, we encourage researchers to share the raw data files and the minimal preprocessing scripts used to generate the CND data.

Analysis

The CND specifications were designed so that data can be easily input into several toolboxes (mTRF-Toolbox, EelBrain-toolbox, NapLib) with very little or no reshaping. The mTRF-Toolbox follows the same format and thus no changes are necessary for it, and several conversion functions have already been written to facilitate loading data saved in CND format into the users' preferred toolbox format. Once data have been loaded, we advocate for a standardised general approach to fitting and testing temporal response functions. Broadly speaking, TRFs should be *cross-validated*. In other words, evaluation should be done on data unseen by the fitting procedure. This is most typically accomplished by splitting data into training and testing partitions, fitting TRFs on the former partition, and evaluating their ability to predict data from the withheld partition. More conservatively, when optimising the regularisation parameter, we recommend carrying out a nested loop cross validation – that is, three partitions: one for fitting the model, the second one for determining the optimal value for the regularisation parameter, and a third partition for evaluating the final model. TRF results can be visualised with the functions in the tutorials and example scripts, showing regularisation tuning curves, temporal response functions, and scalp maps. Again, we look to the community to contribute new plotting functionality via the git repository.

Tutorials

The CND data format was developed alongside the CNSP workshop. Several tutorial scripts have also been developed that read in data in CND format and implement the abovementioned analysis pipeline. These scripts and video recordings of the tutorial sessions have been shared on the CNSP website (https://cnspworkshop.net) to serve as a blueprint for novice users wishing to utilise these resources and use the CND format. They additionally serve to standardise analyses and reduce the risk of coding errors. We don't see these scripts as a finished product – instead we encourage members of the CNSP community to contribute changes and corrections so that we can reach a broader consensus on a standard approach to analysing continuous neural data.
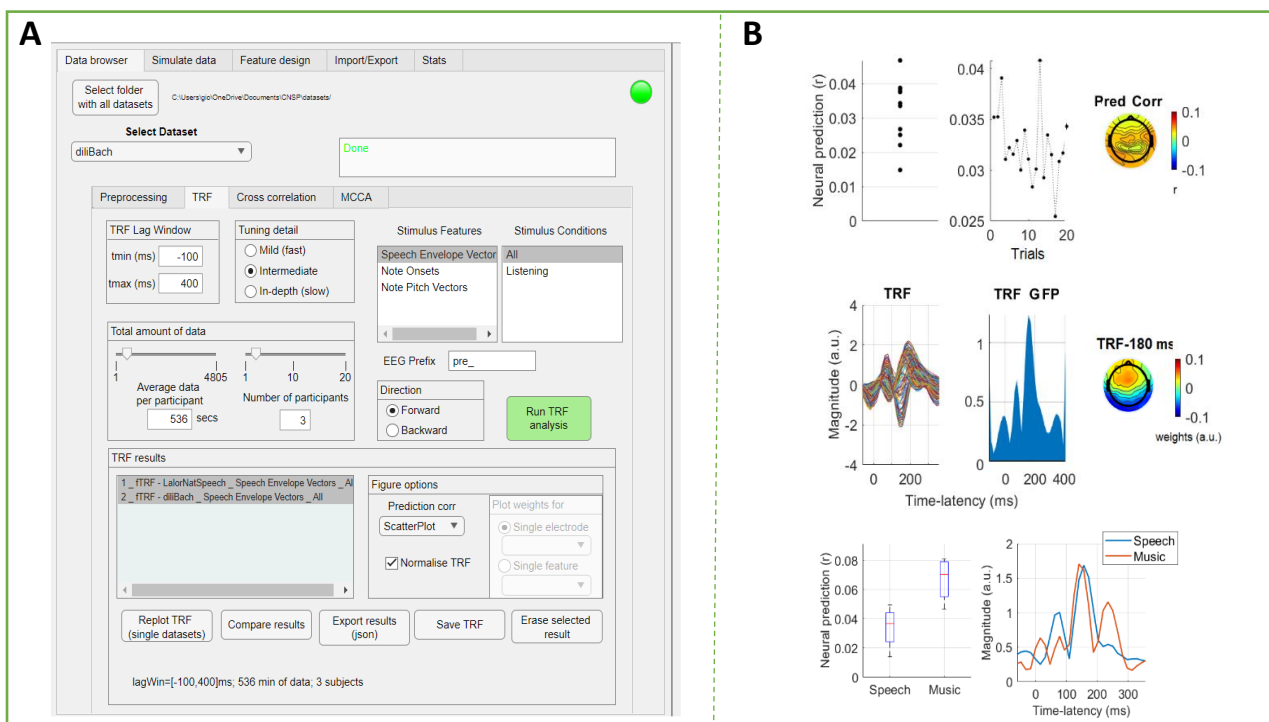
Data browser GUI

The tutorials and example scripts described in the previous sections aim to serve both as a learning resource and as a blueprint for conducting new analyses. Despite our recommendations on the analysis pipeline, there remain a number of analytic choices that depend on the specific assumptions and objectives of the study. For example, preprocessing choices involving data filtering, as well as analytic choices on the specific methodology (e.g., mTRF, cross-correlation) and parameters (e.g., TRF lag-window) to use. To simplify the process, we used MATLAB-software to develop a graphical user interface (GUI) where those choices and the specific options in a given dataset are clearly presented to the user. This GUI, which is referred to as the CNSP Data Browser, **enables the rapid and effortless re-analysis of existing datasets**, guiding the users through parameter selection. We expect this GUI to be particularly useful when getting familiar with the CNSP resources for the first time or when teaching, and well as serving as a tool for expert users when needing to rapidly re-analyse existing data. While all MATLAB code is shared, we have also made available an executable version of the Data Browser, bypassing the time-consuming process of setting up the libraries and dependencies and which, in fact, does not require an installation of MATLAB software in the first place.

In addition to the import/export, preprocessing, and analysis functions presented in the previous sections, the Data Browser includes functions for running power analyses, extracting and combining stimulus features, and simulating neural data. *Power analyses* are useful for determining an appropriate sample size when planning an experiment. While that analysis can be carried out on previous data from the experimenter or based on published results, new studies may involve unexplored analyses or parameters that could be run on existing data but would typically be heavily time-consuming. The CNSP Data Browser provides access to existing datasets, with rapid re-analysis functions and a simulation toolkit that can be used to generate results which are as close as possible to the new experiment that the user intends to run. In doing so, the Data Browser firstly aims to facilitate the design of future experiments, for example by speeding up the estimation of a reasonable sample size for the future experiment through the functions for power analysis in the GUI. Second, it provides a preliminary set of functions for the extraction of basic stimulus features. A function is also provided for generating multivariate features by combining existing features (e.g., spectrogram concatenated with lexical surprisal), simplifying the use of multivariate TRF analyses. Finally, the Data Browser is also the front end of the new functions for neural data simulation, which is discussed in detail in the next section.

Figure 3 depicts the appearance of the Data Browser GUI. Here we discuss the functions that are most commonly used: the preprocessing and TRF analyses. For further details please refer

to the CNSP website (https://cnspworkshop.net), where documentation, tutorials, and video-tutorials are available and up-to-date. The GUI offers basic preprocessing functions: Data re-referencing (e.g., global average referencing), band-pass filtering, envelope extraction (e.g., for high-gamma or spiking data), downsampling, and bad-channel removal. Filters were designed to minimise the impact of their artifacts on the data[41]. Specifically, zero-phase shift Butterworth filters of order 2 are used. Encoding and decoding (i.e., forward and backward) TRF analyses can be run with different selections of parameters (e.g., time-lag window, stimulus features). Interestingly, there is a function for limiting the number of participants and the amount of data per participant (in minutes). This functionality can be used for testing how much data is needed for the effect of interest to emerge. For ease of use, we also include a separate *Stats* tab with a power analysis calculator, which can be used to formally determine the minimum sample size needed to detect an effect of a given size.

Figure 6. The Cognition and Natural Sensory Processing (CNSP) Data Browser. (A) *Screenshot of the Data Browser (left). There are five tabs coresponding to experimental analysis steps (data browser, stimulate data, feature design, import/export, and stats). Under the 'Data Browser' tab, the drop down menu permits dataset selection. There are four tabs corresponding to data browser settings (pre-processing, TRF, cross-correlation, MCCA). Under these settings, a user can select a specific band-pass filtering bandwidth, downsampling ratio, and TRF model hyperparameter values. There are also tabs for feature selection, data exporting, and statistical testing. **(B)** The GUI can visualize results across datasets or parameter configurations .The plots in the top and middle panels are the typical visualisations for forward TRF models. The top panels show, from left to right, the* <u>*EEG prediction correlations*</u> *for each participant (average across channels and trials), for each trial (average across channels and participants), and for each EEG channel (average across participants and trials). The middle panels show the* <u>*TRF weights*</u> *of a speech envelope TRF model for each channel (average across participants), the Global Field Power of the TRF weight (GFP), and the topography of the TRF weights at the peak latency of the GFP. The bottom plots* <u>*compare speech and music TRFs*</u> *from different datasets in terms of EEG prediction correlations and GFP.*

Simulating

One key issue with neural signal analysis is that the ground-truth signal is not available. In other words, we do not know what exact neural signal of interest is buried behind the large EEG/MEG noise[a]. The only signal we have access to is the mixture of the neural signals of interest and noise, from multiple channels. Estimating the consistent neural response behind the EEG/MEG noise is the goal of methodologies such as ERPs/ERFs and TRFs, which allow us to test experimental hypotheses on that hidden neural response. However, there are scenarios where the ground-truth neural signal must first be known. One such scenario is the development of novel methodologies for neural signal analysis. For example, methodologies such as Denoising Source Separation (DSS) and Canonical Correlation Analysis (CCA) have been tested by considering various scenarios on simulated data, such as different signal-to-noise ratio (SNR), where the ground-truth signal was known by construction. Another important scenario where simulated data is necessary is the generation of numerical expectations on a given hypothesis. For example, it can be difficult to have clear expectations on what a TRF model would capture when considering certain multivariate models, especially when the stimulus features are highly correlated with each other. Therefore, simulated data can be used to examine the results when considering a specific algorithm, set of parameters, and hypothesis.

Figure 5 depicts the graphical front-end for the simulation toolkit. The simulation consists of convolving a ground-truth TRF with a given stimulus feature, and adding a specified amount of noise. The noise can either be white noise with a different magnitude, or segments of real EEG/MEG signals taken from other time-points, trials, or experiments. The approach can also be expanded to multivariate models in an additive manner. For example, EEG signals may be simulated by summing EEG noise, neural responses to the speech envelope, and neural responses to word onsets, where the neural responses are simulated by convolving each feature with their specific ground-truth TRF.

The simulation pipeline involves three steps: 1) Design TRF (optional), 2) Select TRF, and 3) Simulate neural signal. First, a TRF can be designed from scratch by indicating a list x-y values (e.g., time and magnitude of a TRF peak). The TRF will then be obtained by fitting a curve based on those datapoints (**Fig. 5A**). Univariate TRFs can be combined into multivariate TRFs through the 'combine' function. In the 'Select TRF' step, the user should select a TRF (univariate or multivariate) that was either built through step 1, or which was obtained from the analysis of actual EEG/MEG data. Finally, the selected TRF (univariate or multivariate) is used to generate simulated neural data, with the addition of the selected type of noise with the given SNR. Simulated data can be generated

---

[a] Here, "noise" refers to the part of the signal that is not of interest (e.g., cortical activity not relevant to the study, motion artifacts, eye-blink artifacts).

for a single electrode or for all electrodes, for a single participant or multiple participants, depending on the specific requirements. **Figure 5B** shows selected examples of simulated neural traces based on different combinations of parameters.


Figure 7. Simulation toolkit. The simulation front-end is organised into three tabs. (A) First, the user can design a new temporal response function (TRF) by interpolating a list of given datapoints. A typical speech envelope TRF is designed by indicating datapoints describing the P1, N1, and P2 components. Datapoints should capture the magnitude and latency of each component as well as the time needed for the component to return to baseline. Multivariate TRFs can also be built by concatenating pre-built TRF. (B) The user may decide to utilise a TRF they built with the TRF designer, or one of the standard TRFs that are included with the simulation toolkit. (C) The user may then proceed to the 'Simulate' tab. The selected TRF will be convolved with an existing stimulus feature time-series (within dataStim.mat), producing a ground-truth EEG/MEG trace. Noise will be added with the selected signal-to-noise-ratio (SNR) to simulate more realistic EEG/MEG data. The noise can be white noise or real EEG/MEG data, preferably from another experiment. The resulting simulated data can then be analysed with the data browser as a dataset.

Web-based Data Browser

The work presented above primarily relies on MATLAB software. While the data browser GUI eliminates the client-side need for a MATLAB license, the user remains in charge of appropriately installing the application and downloading the datasets. However, the rapidly growing number of publicly available datasets will likely challenge the use of a GUI in local due to space limitations. Here we present the first web-based data browser for continuous sensory neurophysiology. While the functionalities will be similar to those of the MATLAB-based GUI, the current version focuses on forward and backward TRF models only. This prototype web-application stores datasets and runs analyses directly on the cloud, meaning that the user is no longer limited to the computational constraints of their own machine. In turn, the use of Python language for implementing the web-

20

based data browser, relying on the Python implementation of the mTRF-Toolbox[36,42], enhances the accessibility of this project.

The web-based CNSP data browser combines a frontend written in React, a popular user-interface JavaScript library, and a backend in Flask, a micro web framework for Python. The frontend employs React's component-based structure for a dynamic user interface, while Flask handles running the analyses and communication. Google App Engine ensures seamless deployment, auto-scaling, and load balancing while Google Cloud Storage securely manages the stored datasets and images from previous analyses, reducing the computational load. This integrated approach offers a scalable, responsive, and reliable web application architecture that holds promise for future expansion and improvement. The web-based CNSP Data Browser can be accessed from the resource page of the CNSP website: https://cnspworkshop.net/resources.html.
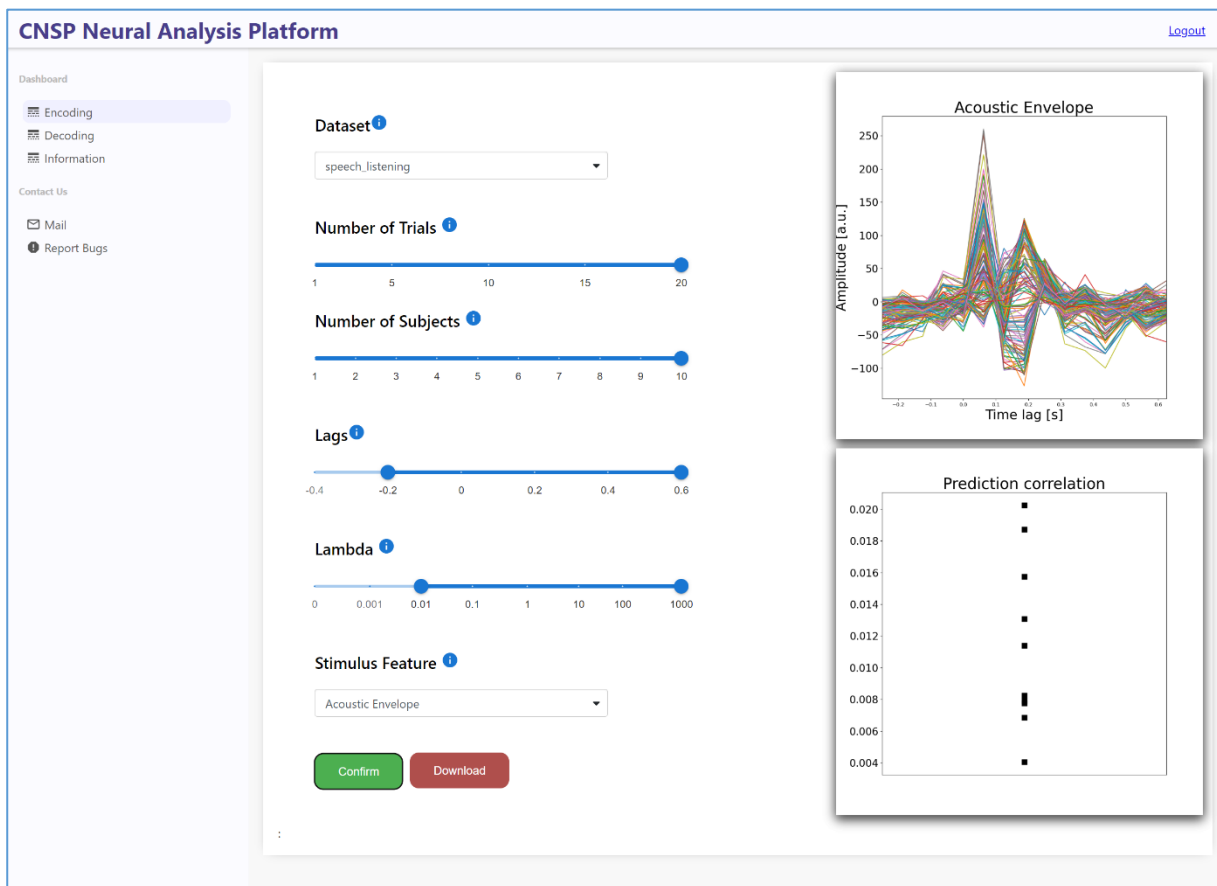
Figure 8: The Web-based CNSP Data Browser*. The web-based data browser aims to serve the same functions of the MATLAB GUI, but from an Internet browser. The web-based data browsed, which is Python-based, currently supports mTRF forward and backward models i.e., encoding and decoding models. One of the key advantages of this innovative piece of software is that it enables the use of TRF analyses without the need for local installation or download; expanding accessiblity to continous-event neural data. Future browser iterations will add functionalities from other toolkits,*

such as CCA, NapLib, and EelBrain, enabling the use of a large set of methodologies on the same dataset.

Discussion

Here, we provide a comprehensive set of resources for the re-analysis and sharing of neural data involving continuous sensory stimuli. The guidelines in this manuscript encompass the entire analytic pipeline, from the definition of a new standardised data structure for storing continuous sensory neural data, to data analysis, import/export functions connecting the present work with other resources in the literature, and simulation functions for the formulation of specific hypotheses. The guidelines in this manuscript are complemented with educational resources, such as tutorial scripts and video-tutorials, which are available on https://cnspinitiative.net.

The proposed approach is in line with the FAIR principles (Findable, Accessible, Interoperable and Reusable)[43]. The set of resources and guidelines are built around the principle of reusability. Interoperability is ensured in multiple ways. The proposed CND structure and conversion mechanisms from/to BIDS ensure that the data can be easily analysed with existing tools across various operating systems and programming languages. The CNSP Data Browser provides executables for various operating systems, bypassing the need for a MATLAB licence. Finally, the web-based data browser is a proof-of-concept front-end for combining and comparing tools from different libraries and programming languages, further contributing to interoperability. Regarding the 'finding' and 'accessible' principle, while addressing these principles directly was beyond the scope of the present article, we encourage the adherence to these principles, for example by relying on reliable and findable repositories providing unique identifiers, such as OpenNeuro (https://openneuro.org/) and Dryad (https://datadryad.org/).

The key intention behind this article is to simplify analysis by establishing an appropriate pipeline and sharing tools that reduce the amount of work required for new experiments. The proposed approach makes it easy to try different methodologies and parameter choices on the same dataset which heavily simplifies the direct comparison of methodologies from different teams. Functionalities such as the simulation toolkit in the GUI aim to encourage the use of simulations. For example, studies with multivariate TRF could greatly benefit from numerical simulations; yet, they are seldom availed of. However, one important observation is that offering easily accessible analytic functionalities through a GUI carries risks. For example, a user might use the GUI as a black-box, without truly understanding what it is doing and, thus, potentially leading to misinterpretation of the results. The GUI was designed to mitigate that risk, by putting an emphasis on speeding-up the analysis pipeline, but without providing the full range of functionalities in the actual CNSP libraries. So, the present version of the data browser offers the necessary tools for running typical TRF (and

other) analyses, replicating existing results, running simulations and power analyses. But all of that is constrained so that only the typical analyses can be carried out, thus minimising the risk of misuse. In that sense, the educational resources such as the tutorials, video-tutorials, and workshop are key as they empower the users, giving them the tools for fully availing of the CNSP resources.

The cohesive set of resources presented in this article highlights the importance of truly committing to the 'reusability' FAIR principle. This is unfortunately not always the case, as the data shared in this research domain is often shared just by including information that is necessary for result replication. For example, an EEG study involving a speech listening task should ideally be shared by including the original audio files and the raw EEG data, so that it can be re-analysed in various different ways. Instead, some existing datasets (including early datasets from the authors of this manuscript) did not include the raw audio files, for example due to copyright limitations. That and other similar issues should be carefully taken into consideration at the experimental design stage, now that the possibility and benefits of data reusability are established. As a result, we encourage journals and reviewers to take reusability more into consideration in the future. In other words, sharing a dataset in a standardised format like BIDS should not be taken as a guarantee in itself that the data is shared appropriately. Instead, the data sharing statement should include considerations on the possible reusability of the dataset, thus reflecting on the best way of sharing data beyond replicability. Another important issue is that the code used for the analyses is often not fully shared during the journal publication process. Instead, we encourage journals to be more strict on that point, which is essential for result replication i.e., scientific papers should include either the custom code used or the exact reference to the tools used, such as a DOI, version number, or at least the commit index (e.g., GitHub commit index).

We expect that future research in this domain will lead to a large set of data that will open new possibilities for both theoretical investigations and methodological development, benefiting neurophysiology research at large. In these re-analysis scenarios, it is important to stress the key role of result replication. In fact, apparently valid results may emerge by chance due to multiple re-analyses of the same dataset from different individuals. For this reason, it is extremely important to replicate the results on another publicly available dataset or, even better, on a new dataset.

Limitations and future work

The present work represents the first comprehensive and cohesive set of analytic resources, from data standardisation to data analysis and simulation, for continuous sensory neural data. In doing so, the main contribution of this work is the attempt to build resources and guidelines that are tailored to the specific needs of this field of research. Indeed, the resources provided are constantly improved as the field advances, making this specific manuscript not immune to technical limitations. Here, we discuss these limitations with the goal of pin-pointing important directions for future work in this area.

One limitation with this work is that the majority of the code and resources shared for neural data analysis rely on the MATLAB language, which limits access to these resources to individuals with prior expertise with that language, and that hold a MATLAB license. That issue was mitigated by producing the executable graphical interface, which was built with MATLAB but can run as a stand-alone piece of software. Furthermore, we also developed a Python-based web-based data browser. While the web-based data browser only has limited functionalities at the time this was written, future work will substantially extend its functions by giving it full access to a variety of relevant toolboxes, such as the mTRF-Toolbox, Eelbrain-toolbox, Naplib, and others. In doing so, this tool aims to encapsulate the open-science change in sensory neuroscience, implementing the relevant FAIR principles and opening new possibilities beyond result replication and re-analysis, such as developments involving the availability of big, aggregated datasets.

The description in this manuscript focuses on speech and music perception datasets. Future research should test the validity of the guidelines and resources onto other relevant domains in sensory neuroscience, starting from research involving rapid continuous stimuli in other sensory modalities, such as visual and tactile modalities[44]. The work could also be expanded to other domains involving motor movements, where methodologies such as TRFs and CCA can be used[45]. As future work will continue developing these tools by incorporating other methodologies and additional datasets, the metadata will be expanded to keep track of the version number of the tools used for storing and manipulating the data.

Beyond the specific resources, scripts, and data structures, we contend that the most important and long-lasting overarching idea will be facilitating the use of lab specific resources (scripts and data) while connecting resources from different teams, so that they are interoperable. Important lessons can be learnt from other research fields, such as bioinformatics, guiding future development in sensory neuroscience toward more efficient research framework, encouraging the use of these domain-specific guidelines for data sharing while avoiding risks such as excessively rigid data sharing policies, which could do more harm than good. Indeed, a collective effort is necessary to achieve that goal, and we invite the research community to contribute to this endeavour by sharing data, well-documented scripts, and tutorials, enriching the CNSP resources with their methodologies and data.

Author contributions

References

1   Clark, A. *Surfing Uncertainty: Prediction, Action, and the Embodied Mind*. (Oxford University Press, 2016).

2   Kutas, M. & Federmeier, K. D. Thirty years and counting: finding meaning in the N400 component of the event-related brain potential (ERP). *Annual review of psychology* **62**, 621-647 (2011). https://doi.org:10.1146/annurev.psych.093008.131123

3   Paller, K. A., McCarthy, G. & Wood, C. C. Event-related potentials elicited by deviant endings to melodies. *Psychophysiology* **29**, 202-206 (1992).

4   Yabe, H., Tervaniemi, M., Reinikainen, K. & Näätänen, R. Temporal window of integration revealed by MMN to sound omission. *NeuroReport* **8**, 1971-1974 (1997). https://doi.org:10.1097/00001756-199705260-00035

5   Theunissen, F. E. *et al.* Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli. *Network: Computation in Neural Systems* **12**, 289-316 (2001).

6   David, S. V., Mesgarani, N. & Shamma, S. A. Estimating sparse spectro-temporal receptive fields with natural stimuli. *Network: Computation in Neural Systems* **18**, 191-212 (2007). https://doi.org:10.1080/09548980701609235

7   Bonte, M., Parviainen, T., Hytönen, K. & Salmelin, R. Time course of top-down and bottom-up influences on syllable processing in the auditory cortex. *Cerebral Cortex* **16**, 115-123 (2006).

8   Fahmie, T. A. *et al.* Toward an explicit technology of ecological validity. *Journal of Applied Behavior Analysis* **56**, 302-322 (2023). https://doi.org:https://doi.org/10.1002/jaba.972

9   Tervaniemi, M. The neuroscience of music; towards ecological validity. *Trends in Neurosciences* **46**, 355-364 (2023). https://doi.org:10.1016/j.tins.2023.03.001

10   Broderick, M., Anderson, A., Di Liberto, G., Crosse, M. & Lalor, E.   (2018).

11   Di Liberto, G. M. *et al.* Cortical encoding of melodic expectations in human temporal cortex. *eLife* **9** (2020). https://doi.org:10.7554/eLife.51784

12   Hale, J., Dyer, C., Kuncoro, A. & Brennan, J. R. in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers).* 2727-2736 (Association for Computational Linguistics).

13   Das, N., Biesmans, W., Bertrand, A. & Francart, T. The effect of head-related filtering and ear-specific decoding bias on auditory attention detection. *Journal of Neural Engineering* **13**, 056014-056014 (2016). https://doi.org:10.1088/1741-2560/13/5/056014

14   Di Liberto, G. M., O'Sullivan, J. A. & Lalor, E. C. Low-frequency cortical entrainment to speech reflects phoneme-level processing. *Current Biology* **25** (2015). https://doi.org:10.1016/j.cub.2015.08.030

15   Ding, N. & Simon, J. Z. Cortical Entrainment to Continuous Speech: Functional Roles and Interpretations. *Frontiers in human neuroscience* **8** (2014). https://doi.org:10.3389/fnhum.2014.00311

16   Kern, P., Heilbron, M., de Lange, F. P. & Spaak, E. Cortical activity during naturalistic music listening reflects short-range predictions based on long-term experience. *eLife* **11**, e80935 (2022). https://doi.org:10.7554/eLife.80935

17   Jessica Tan, S. H., Kalashnikova, M., Di Liberto, G. M., Crosse, M. J. & Burnham, D. Seeing a talking face matters: The relationship between cortical tracking of continuous auditory‐visual speech and gaze behaviour in infants, children and adults. *NeuroImage* **256**, 119217 (2022). https://doi.org:https://doi.org/10.1016/j.neuroimage.2022.119217

18   Crosse, M. J., Di Liberto, G. M. & Lalor, E. C. Eye can hear clearly now: Inverse effectiveness in natural audiovisual speech processing relies on long-term crossmodal temporal integration. *Journal of Neuroscience* **36** (2016). https://doi.org:10.1523/JNEUROSCI.1396-16.2016

19   Lalor, E. C., Pearlmutter, B. A., Reilly, R. B., McDarby, G. & Foxe, J. J. The VESPA: a method for the rapid estimation of a visual evoked potential. *NeuroImage* **32**, 1549-1561 (2006).

20   Guilleminot, P. & Reichenbach, T. Enhancement of speech-in-noise comprehension through vibrotactile stimulation at the syllabic rate. *Proceedings of the National Academy of Sciences* **119**, e2117000119 (2022).

21   Khalighinejad, B., Cruzatto da Silva, G. & Mesgarani, N. Dynamic Encoding of Acoustic Features in Neural Responses to Continuous Speech. *The Journal of Neuroscience* (2017). https://doi.org:10.1523/jneurosci.2383-16.2017

22   Broderick, M. P., Anderson, A. J., Di Liberto, G. M., Crosse, M. J. & Lalor, E. C. Electrophysiological Correlates of Semantic Dissimilarity Reflect the Comprehension of Natural, Narrative Speech. *Current Biology* (2018). https://doi.org:10.1016/j.cub.2018.01.080

23   Brodbeck, C., Hong, L. E. & Simon, J. Z. Rapid Transformation from Auditory to Linguistic Representations of Continuous Speech. *Current Biology* **28**, 3976-3983.e3975 (2018).

24   Heilbron, M., Armeni, K., Schoffelen, J.-M., Hagoort, P. & de Lange, F. P. A hierarchy of linguistic predictions during natural language comprehension. *Proceedings of the National Academy of Sciences* **119**, e2201968119 (2022). https://doi.org:doi:10.1073/pnas.2201968119

25   Shannon, R. V., Zeng, F. G., Kamath, V., Wygonski, J. & Ekelid, M. Speech recognition with primarily temporal cues. *Science* **270**, 303-304 (1995).

26   Rosen, S. Temporal information in speech: acoustic, auditory and linguistic aspects. *Philos Trans R Soc Lond B Biol Sci* **336**, 367-373 (1992). https://doi.org:10.1098/rstb.1992.0070

27   Crosse, M. J., Di Liberto, G. M., Bednar, A. & Lalor, E. C. The multivariate temporal response function (mTRF) toolbox: a MATLAB toolbox for relating neural signals to continuous stimuli. *Frontiers in human neuroscience* **10**, 604 (2016).

28   Crosse, M. J. *et al.* Linear modeling of neurophysiological responses to speech and other continuous stimuli: methodological considerations for applied research. *Frontiers in neuroscience* **15**, 705621 (2021).

29   Obleser, J. & Kayser, C. Neural entrainment and attentional selection in the listening brain. *Trends in cognitive sciences* **23**, 913-926 (2019).

30   de Cheveigné, A. *et al.* Decoding the auditory brain with canonical component analysis. *NeuroImage* **172**, 206-216 (2018). https://doi.org:10.1016/j.neuroimage.2018.01.033

31   Caucheteux, C., Gramfort, A. & King, J.-R. Deep language algorithms predict semantic comprehension from brain activity. *Scientific Reports* **12**, 16327 (2022). https://doi.org:10.1038/s41598-022-20460-9

32   Lalor, E. C., Power, A. J., Reilly, R. B. & Foxe, J. J. Resolving Precise Temporal Processing Properties of the Auditory System Using Continuous Stimuli. *Journal of Neurophysiology* **102**, 349-359 (2009). https://doi.org:10.1152/jn.90896.2008

33   O'Sullivan, J. A. *et al.* Attentional Selection in a Cocktail Party Environment Can Be Decoded from Single-Trial EEG. *Cerebral Cortex*, bht355-bht355 (2014).

34   Alain, d. C. *NoiseTools*, <http://audition.ens.fr/adc/NoiseTools/> (2023).

35   Crosse, M. J. *et al.* Linear Modeling of Neurophysiological Responses to Speech and Other Continuous Stimuli: Methodological Considerations for Applied Research. *Frontiers in neuroscience* **15**, 705621-705621 (2021). https://doi.org:10.3389/fnins.2021.705621

36   Crosse, M. J., Di Liberto, G. M., Bednar, A. & Lalor, E. C. The multivariate temporal response function (mTRF) toolbox: A MATLAB toolbox for relating neural signals to continuous stimuli. *Frontiers in Human Neuroscience* **10** (2016). https://doi.org:10.3389/fnhum.2016.00604

37   Christian, B., Proloy, D., Teon L, B., Samir, R. & J, K.   (Zenodo, 2023).

38   Mischler, G., Raghavan, V., Keshishian, M. & Mesgarani, N. naplib-python: Neural Acoustic Data Processing and Analysis Tools in Python. *arXiv preprint arXiv:2304.01799* (2023).

39   Gramfort, A. *et al.* MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience* **7** (2013). https://doi.org:10.3389/fnins.2013.00267

40   Giovanni, M. D. L. *et al. CNSP resources*, <https://github.com/CNSP-Workshop/CNSP-resources Commit ID: 714e044934c94da1c0fc175513ca0952f22a9daa> (2023).

41   de Cheveigné, A. & Nelken, I. Filters: when, why, and how (not) to use them. *Neuron* **102**, 280-293 (2019).

42   Bialas, O. & Dou, J.   (Zenodo, 2023).

43   Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* **3**, 1-9 (2016).

44   Dmochowski, J. P., Ki, J. J., DeGuzman, P., Sajda, P. & Parra, L. C. Extracting multidimensional stimulus-response correlations using hybrid encoding-decoding of neural activity. *NeuroImage* **180**, 134-146 (2018).

45   Di Liberto, G. M. *et al.* Robust anticipation of continuous steering actions from electroencephalographic data during simulated driving. *Scientific reports* **11**, 23383 (2021).