



HAL
open science

SLAM – A thin-client for interoperable annotation and biomedical signal handling

Vivien Kraus, Guillaume Dollé, Margaux Alloux, G. Loron, Nathalie Bednarek, Jonathan Beck, François Rousseau, Nicolas Passat

► **To cite this version:**

Vivien Kraus, Guillaume Dollé, Margaux Alloux, G. Loron, Nathalie Bednarek, et al.. SLAM – A thin-client for interoperable annotation and biomedical signal handling. SoftwareX, In press. hal-04576724

HAL Id: hal-04576724

<https://hal.science/hal-04576724>

Submitted on 16 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SLAM – A thin-client for interoperable annotation and biomedical signal handling

Vivien Kraus^a, Guillaume Dollé^b, Margaux Alloux^a, Gauthier Loron^{a,c},
Nathalie Bednarek^{a,c}, Jonathan Beck^c, François Rousseau^d, Nicolas Passat^a

^aUniversité de Reims Champagne Ardenne, CRESTIC UR 3804, 51100 Reims, France,

^bUniversité de Reims Champagne Ardenne, CNRS, LMR, UMR 9008, 51100 Reims, France,

^cService de médecine néonatale et réanimation pédiatrique, CHU de Reims, France,

^dIMT Atlantique, LaTIM INSERM U1101, 29238 Brest, France,

Abstract

Current approaches to biosignal handling in a medical environment increasingly rely on machine learning, which requires precise annotations. In multicentric annotation scenarios, the diversity in annotation protocols leads to imprecise semantics of the annotations. This context shows the need for interoperable annotations, that can be reused with little cost across different protocols. The presented free and open-source software is an application for handling biosignals. It implements a novel method to add asynchronous and interoperable annotations with precise semantics. The annotations are stored in Allotrope Data Format (ADF), a structure based on the Hierarchical Data Format (HDF5). It allows for high performance I/O and aggregating of related patient data (e.g. different acquisition modalities). This research tool can be used solely in a web browser to facilitate integration in more complete environments.

Keywords: Biosignals, EEG, Annotation, Visualization, Semantic Web, ADF, HDF5

Required Metadata

Current code version

Nr.	Code metadata description	Please fill in this column
C1	Current code version	0.7.2
C2	Permanent link to code/repository used for this code version	https://github.com/neonatool/slam
C3	Permanent link to Reproducible Capsule	https://neonatool.github.io/slam
C4	Legal Code License	CeCILL-B
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	C/C++, javascript, python, R
C7	Compilation requirements, operating environments & dependencies	<ul style="list-style-type: none">• python \geq 3.0.0,• rdflib (python) \geq 5.0.0,• esbuild \geq 0.15.12,• plotly.js \geq 2.11.1,• rdflib (JavaScript) \geq 2.2.32,• adftool (github: neonatool/adftool, v0.8.0),• emscripten
C8	If available Link to developer documentation/manual	https://neonatool.github.io/adftool
C9	Support email for questions	vivien.kraus@univ-reims.fr*, guillaume.dolle@univ-reims.fr

Table 1: Code metadata (mandatory)

1. Motivation and significance

The analysis of biosignals is a crucial issue, especially in a medical context. The nature of the data relating to these biosignals is, by definition, complex. ElectroEncephaloGraphy (EEG) recordings are representative examples of such data. In particular, the complexity of the EEG data is inherent in several characteristics, including but not limited to their multidimensional and heterogeneous nature (multiple sensors with spatial and temporal components); their volume (readings can be taken over several hours or even days); their quality (presence of acquisition noise, artifacts).

The analysis of such data by clinical users, who often have limited time and expertise in digital data analysis tools, is therefore a real challenge. To facilitate their work, machine learning algorithms have been proposed to deal with the large amount of data. For instance, [1] proposes a toolbox dedicated to EEG data processing, with an emphasis on deep learning. Other approaches, such as [2], focus on feature extraction, again for machine learning purposes, while some partially rely on symbolic processing of the EEG data, with the help of an ontology-based formalization, for sleep stage classification [3], brain computer interface [4], or epilepsy [5].

Nonetheless, these tools do not replace the need for clinical users to read and analyze the signal. This task becomes even more complex when several users want to manipulate data by combining their work, but without having a clearly defined protocol or formalism for doing so. This type of problem often arises when several clinicians are involved in the care of the same patient, or when several clinical teams collaborate in the analysis of multicenter data obtained within the framework of cohorts.

General-purpose time series annotation tools are available [6, 7, 8, 9]. Such applications are reviewed in [10]. A common use of time series annotation tools is to label anomalies to train automated anomaly detection models [11], or annotate audio tracks [12]. However, some annotation tasks require more specific tools such as large satellite telemetry data [10], ElectroCardioGraphy (ECG) [13, 14], or time-synchronized multi-modal data [15]. As for EEG, precise control over the montage and filtering parameters is required, and general-purpose annotation software is not sufficient.

In clinical routine, EEG handling, visualization and annotations are often carried out via commercial software. Inspired by seminal contributions ten years ago, especially the popular EEG analysis toolbox MNE [16], some efforts were geared towards developing alternative software tools dedicated to this task. One can cite, non-exhaustively, Robin Viewer [17], an open-source, platform-independent, interactive web application, designed for allowing deep-learning paradigms; a clinical EEG research platform [18] designed to allow progressive model construction, including in particular EEG visualization and annotation; copla-editor [19], a web-based visualization, assessment and collaboration tool for multidimensional biosignals; and the time series plug-in of CrowdCurio [20], a generic annotation library with support for EEG data.

Each of these software tools implements some specific functionalities. To

our best knowledge, [18] is not available in open-access. While CrowdCurio [20] claims to support EEG annotation, it is unclear how to run the program [10]. Regarding the other two freely available software platforms, namely Robin's Viewer [17] and copla-editor [19], our contribution shares common properties and purposes, but also differs in various points.

Both projects implement an EEG viewer, with the possibility to add annotations in a web-based user interface. [17] makes extensive use of the MNE library, which enables signal processing tools such as band-pass filters. It can also be used for active learning of a segmentation task, such as artifact detection. [19] is also a web-based EEG visualizer with annotation capabilities. However, these projects do not work solely on a web browser, and do not provide a solution for interoperable annotations. We also note that [17] and [19] must deal with EDF+-compatible annotations, which are single-channel, ambiguous, and do not allow global descriptive meta-data. Other approaches require to keep data and meta-data in the same file, such as BIDS-EEG or SEDS [21]. Linked Data can organize collections of heterogeneous signals [22]. However in order to use the linked data paradigm while keeping the data and meta-data in the same file, the file format must be adapted.

In this context, our contribution is a software tool named Semi-automatic Allotrope Labelling in Medicine (SLAM) that proposes a novel implementation of the following features:

- ★ **Interoperability:** agglomerating and structuring annotations produced across multiple users and centers to enable their capitalization and reuse, based on a semantic web philosophy;
- ★ **Enrichment:** allowing simple semantic annotation of the data based on a dedicated data format (namely ADF), for efficient and versatile management of annotation heterogeneity in cases involving multiple users.

Considering the above requirements, this tool has been developed in accordance with the following guidelines:

- **Interactivity:** facilitating navigation (time, frequency) in large biosignal data;
- **Usability:** such a tool needs to be usable by an expert clinician who is not necessarily familiar with digital technologies. It is therefore essential that it can be deployed in a *light technical environment* (e.g. via a web browser) in a *standalone fashion* (i.e. with minimal dependencies), and that it can be used via a simple, intuitive interface;
- **Modularity:** this tool was initially designed to allow the manipulation of EEG data, but it is potentially adaptable to suit the characteristics of other types of multi-channel temporal signals;
- **Isolation:** the tool must fulfill its objectives with the least amount of privilege (e.g. it can run within a sandbox);

- **Privacy:** the software can work with as little private patient data as necessary.
- **Traceability:** the modifications are incremental (using delta-updates), and no record of operation can be lost.

Our approach runs entirely in a web browser without requiring a running back-end server in the background. It can be easily integrated with third-party tools, without sacrificing the convenience of band-pass filtering for reading. It focuses on inter-operable annotations, where the types of annotations carry precise semantics. Using RDF data enables storage of general metadata without needing a precise data hierarchy in the file. This facilitates interoperability between centers or studies that do not use the exact same vocabulary.

2. Software description

SLAM is a thin open source licensed software (see Table 1) developed to provide solutions to the needs enumerated in Section 1. It enables visualization of biosignals along with their metadata, and labelling of the biosignals in a semi-automatic way. Beside, SLAM aims to facilitate the development of new innovative detection algorithms by allowing visualization through modular components.

SLAM takes into account domain specific applications through profiling for the view model (the User Interface (UI)) and the global setup (the configuration). As an example, a first profile dedicated to neonatal EEG studies is proposed. Specific electrical signal processing algorithms have also been implemented. Considering this setting, the UI was designed to be close to similar EEG software (e.g. Deltamed Coherence [23], Micromed BrainQuick [24], EEGLab [25], MNELab [26]) dedicated to EEG in order to target clinical specialists preserving their daily handling habits in clinical contexts. This profile allows to select the EEG head mount setup (e.g. raw channels, bipolar references), but also a class of semantic annotations that are hierarchically structured to allow defining sub-classes.

The software is developed partially in C language focusing on portability with minimal dependencies. SLAM contains several C modules provided as C libraries exported to javascript through emscripten to allow future integration to data management platforms. The current implementation has been developed to achieve full functionality within the security sandbox provided by the web browser. The annotation tool has been designed to handle data modifications as incremental delta updates, thanks to the ADF file format. The workflow was designed for two typical usages: (1) allowing hot visualization of EEG data aside their metadata; and (2) feeding metadata with new labels (dedicated to subsequent supervised algorithms).

The software architecture was designed to be decentralized. It may allow asynchronous annotations or live collaborative annotations in future versions. The current implementation already enables the storage of delta updates

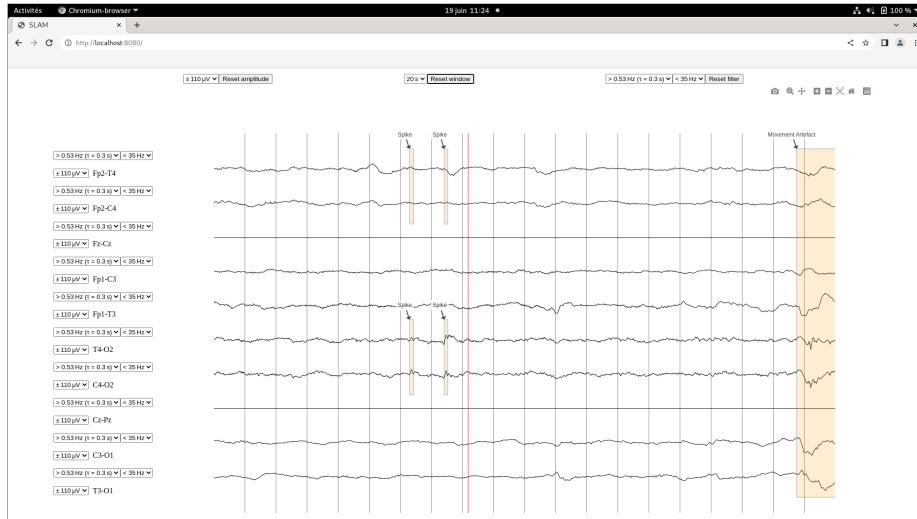


Figure 1: Main view of SLAM.

such as label modifications thanks to the Allotrope Data Format (ADF) format and our dedicated ADF utility tool (adftool) [27]. Considering a blockchain paradigm in further development may enhance the global security and traceability of our software. In particular, for the multi-centric co-annotation case, tracking and authenticating user actions may prevent data integrity loss, for instance caused by a malicious attack due to a security breach in the system of information of one of the centers. This first version of SLAM has not been optimized yet with the purpose to achieve high performance.

2.1. Functionalities

The software allows to plot the different EEG channels data and navigate through them. The main functionalities are the following:

- Reading EEG (viewing the EEGs, modifying reading parameters, and navigating through them):
 - Window size tuning (10-second, 20-second or 30-second). A grid of vertical lines across the horizontal axis is displayed, regardless of the actual time window length.
 - Window navigation. The current view can move left or right by a half window length, for the sake of readability.
 - Amplitude selection. Clinicians who can comfortably read EEGs on a computer monitor with an equivalent resolution of $10\mu V/mm$ according to DeltaMed, have the same experience reading a signal with SLAM if a range of $220\mu V$ is in use. Other values are possible, for activities that are abnormally low or high. This parameter can be tuned independently for each channel, for the sake of readability.

- Filter selection. In EEG, activities are sought at specific frequencies not exceeding 70 Hz. Signal at other frequencies are generally removed. SLAM uses a finite impulse response convolution filter (with the same default parameters as the Python MNE library). SLAM can also be configured to use a low-pass filter with a cutoff frequency of 35 Hz for a (set of) channel(s). Existing EEG recording devices often use a high-pass filter.
- Overriding parameters for all channels. SLAM can override the amplitude and/or filter parameters for all channels at once.
- Adding annotations. SLAM can:
 - set the bounds of the annotations;
 - choose an annotation class from the reading profile;
 - choose the channels where the annotation occurs.

The annotations are saved directly in the file. Once the annotation job is done, the updated file can be downloaded from the application.

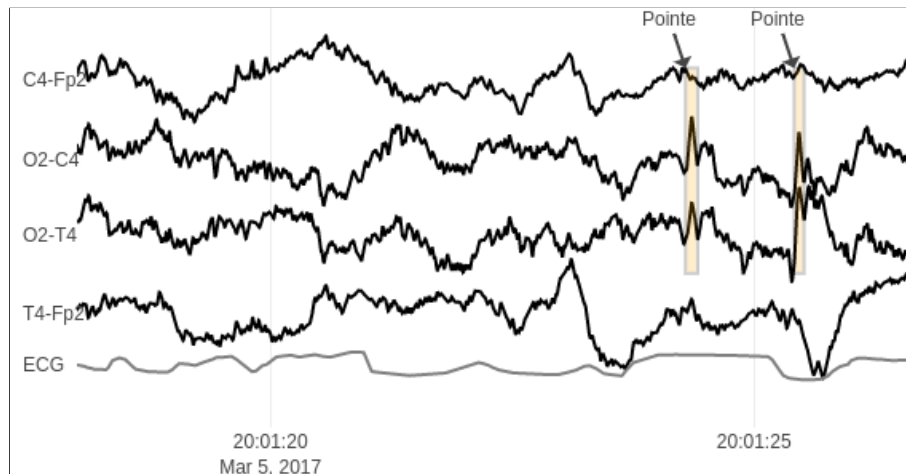
2.2. User interface

SLAM provides a web-based UI, illustrated in Figure 1. The current layout is designed as follows:

- A top bar menu for opening a file, selecting a profile, exporting, displaying help and setting the global parameters (amplitude, filter parameter values, length of the time window).
- A central block component for plotting the different biosignal channels. The left and right arrow keys control the movement of the view. When moving the pointer over the plot window, a red vertical line is displayed to show the current position of the cursor. Clicking on the plot sets the start or end of a new annotation.
- On the left side, per channel settings that can be set (amplitude and filter parameters of each channel) with precedence over the global parameters.
- A bottom part where annotations can be added, by providing start and stop times, annotation type, and the location, shown in Figure 2. The list of possible annotation types and channels match what is displayed in the plot window. (Three meta-channels can be selected: all channels, left side, or right side.)

2.3. Software architecture

SLAM is composed of several components. While it entirely runs offline in a web browser, it features different execution contexts that run in a front-end or back-end (see Figure 3).



Depuis : 2017-03-05T19:00:07.199Z => Jusqu'à : 2017-03-05T19:00:13.382Z

Classe d'annotation :

- Annotation
- PPR
- Artéfact Myogramme
- Alpha
- Encoche
- Thêta Temporal
- Artéfact Mouvement
- Pointe
- Non Artéfact

Canaux :

- C3-Fp1
- O1-C3
- O1-T3
- T3-Fp1
- Cz-Fz
- Pz-Cz
- C4-Fp2
- O2-C4
- O2-T4
- T4-Fp2
- ECG

Supprimer sauvegarder

Début	Durée	Score	Désaccord
2017-03-05T19:48:58.000Z	19.996094	1.305453881729902	0.04136721472565945
2017-03-05T19:52:48.000Z	19.996094	1.2082110631792344	0.03154650994208996
2017-03-05T19:51:18.000Z	19.996094	1.1740265817315527	0.026591558805460314
2017-03-05T19:49:58.000Z	19.996094	1.1635138154397557	0.026513550652336983
2017-03-05T19:51:58.000Z	19.996094	1.15590253791925	0.01981468370080293
2017-03-05T19:48:48.000Z	19.996094	1.126720642150175	0.020606964924381722
2017-03-05T19:49:08.000Z	19.996094	1.1217027841290810	0.025424800500752128

Figure 2: Visualization of an EEG in bipolar montage, annotation editing and artifact detection report.

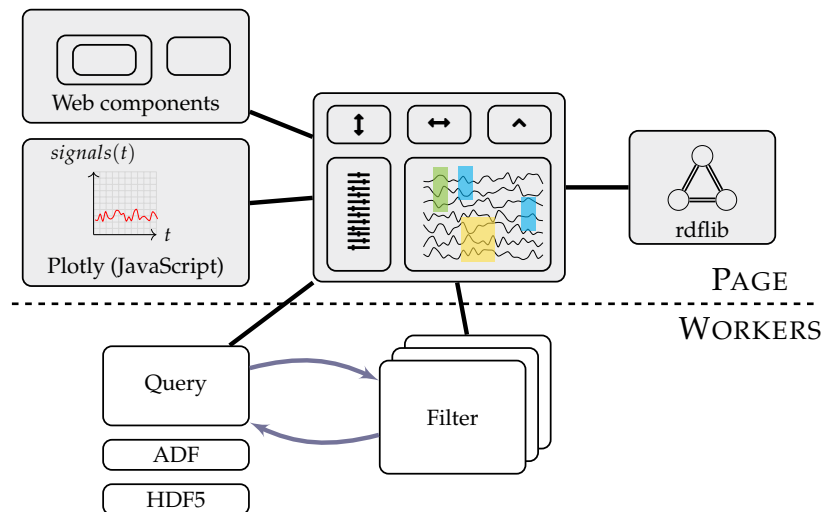


Figure 3: The different components of SLAM.

Adftool core. The core library provides an Application Programming Interface to read and write ADF files (see Section 2.4) with raw channel data. It comes with a command-line interface for low-level direct manipulations of the file, and R, Python and JavaScript bindings, since it can be compiled with emscripten.

Web workers. Web workers are used for parallelizing access to the data in the file, and filtering the data. One worker can access the file, while many independent workers are responsible for filtering the data. The threads communicate during the filtering phase; see Figure 4. The main thread (the UI) requests filtered data to a worker. The filter thread communicates with the worker holding the file in order to load the data, with a margin. The margin is used by the convolutional filter, and its length depends on the filter parameters. Then, the filter is applied, and the filtered data is returned to the main thread.

Plotly backend. To facilitate integration of SLAM with other platforms, and given the efficiency requirements for plotting millions of data points, Plotly was chosen for the rendering back-end. The data to display is loaded on demand, based on the actual range requested by Plotly.

Rdflib. To link the data in the file (e.g. channel and annotation types) and data in the profile (e.g. the anode and cathode of bipolar references in the montage) and ontology (e.g. sub-classes of annotations), SLAM uses rdflib, the JavaScript library to process RDF data.

Web components. The user interface widgets have been designed as ECMAScript standard web components, to avoid adding new dependencies to the project.

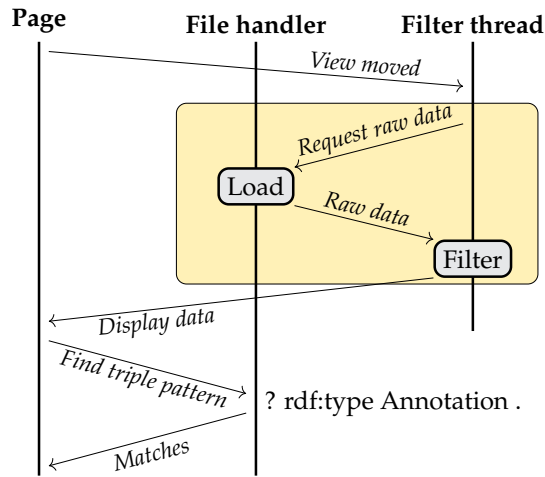


Figure 4: Communication between three different contexts: the main thread, the file reader thread, and one filtering thread, in order to filter the signal and query data. The interaction between the filter thread and file handler (in orange) only happens for a (partial) cache miss.

2.4. Data storage model

SLAM uses a linked data model to store EEGs and annotations. The data may be stored in the EEG file itself, and in publicly accessible online ontologies, describing the operation done with the data. For interoperability purposes, reading profiles are expected to be published. A default ontology is provided with SLAM for the purpose of annotating (newborn) EEGs, but other sources may be added.

In order to store both raw biosignal data, and relevant metadata (including annotations), while keeping an extension possibility for multimodal data, SLAM is based on the processing of HDF5 files. The biosignals may have a very large dimension, and the access to their data is skewed towards many reading operations, mostly in sequential access, and very few if any write operations. The processing of RDF data is very light in comparison, so the file format to exchange biosignal data had been primarily chosen to enable fast and parallel readings of row-wise slices of long matrix data. However, since one of the primary roles of SLAM is to add metadata, the file format should also be able to easily add more RDF triples or remove them. This constraint dismisses most existing RDF serialization formats. Since we also want to be able to store general metadata that are not timestamped, and that the density of timestamped annotations may not be uniform, it is impractical to use EDF+ as a basis¹.

¹In the EDF standard, the biosignal is split into time-contiguous epochs, and only a limited file space is dedicated to metadata. This specification was later extended with EDF+, enabling the addition of a fixed amount of time-constrained metadata per epoch. Each EDF+ annotation may have at most one channel where the annotation is located, which is not practical for the annotation

The ADF file format is a collection of specifications that enables processing of RDF data on top of HDF5 files. It has been used for pharmaceutical problems [28]. A distinctive feature of ADF is to incrementally mark deleted triples, instead of removing them from the file, improving traceability of the operations. While a partial reference implementation exists under the name H5LD, it is not complete enough, and is not able to query simple quad patterns, add new data, or remove data. Other popular linked data exchange formats, such as JSON-LD, do not take advantage of the HDF5 file format, which makes it all the more difficult to add data, or build indices for fast triple pattern queries. The upper layers of the ADF specification concern integration with tools such as Apache Jena, that we do not use.

We thus propose a new file format, as a hybrid between plain HDF5 and ADF Quad store. The raw signal values are written as an HDF5 dataset, under a well-known dataset name, and the file is also used as an ADF Quad Store, that can be queried for any quad pattern. This part of the implementation is written in C and links to the HDF5 library. It is compiled to webassembly using emscripten, by a GitHub action in a separate repository [27].

3. Illustrative example

In this section, we propose a toy-example, which is simpler but also easier to understand than actual use cases encountered in clinical contexts. Let us assume that a first clinical unit studies EEGs of patients in a cohort study (e.g. neonates, which is the current case presented in the impact section). In this study, denoted as “study A”, we assume that two kinds of patterns are of interest, namely synchronous and asynchronous frontal sharp transients (FSP). A few asynchronous annotations have been added in study A, such as the one in Figure 6a, displayed as “asynchronous FSP (study A)”.

Now, let us assume that a few months later, a second clinical unit aims to study the same data, but with slightly distinct purposes. In this second study, denoted as “study B”, we assume that many kinds of patterns are of interest, namely frontal sharp transients but also other kinds of spikes. The distinction between synchronous and asynchronous frontal sharp transients is not relevant. In this second study B, it may be relevant to take advantage of the annotations performed in study A.

The computer systems in both clinical centers are supposed to be inflexible, and no data scheme is supposed to be updated between studies. The situation is summarized by Figure 5.

In order to set preferences for reading EEGs, SLAM defines a reading profile, which consists of the montage used to display the signal, but also the classes of annotation of interest. Whenever an annotation has a type that is

of many types of biomarkers that can be annotated on signals, especially using a bipolar referenced montage.

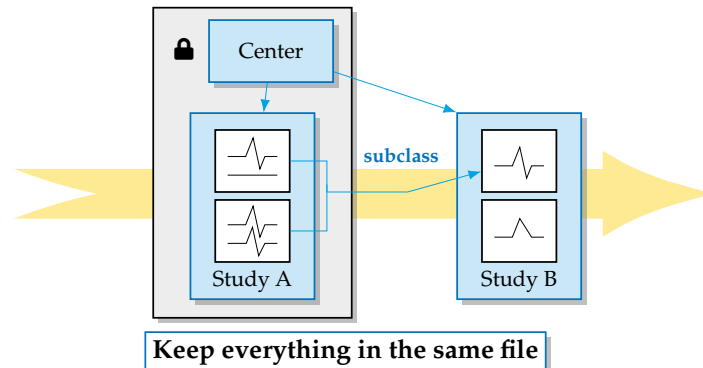


Figure 5: A new study, reusing both kinds of frontal sharp transients from a previous study, in an inflexible computer system. (Illustrative example in Section 3.)

marked as a subclass of an annotation class of interest in the profile, SLAM displays it as the class used in the profile. Thus, a new profile can be created in order to merge both kinds of frontal sharp transients, by defining a generic frontal sharp transient class as a superclass of synchronous and asynchronous frontal sharp transient classes.

By opening a file that has already been annotated by study A, with the second profile, SLAM displays all instances of frontal sharp transients the same way, thanks to the subclass relationship. The previously annotated FSP is now displayed as “FSP (study B)” automatically in Figure 6b. It also lets users annotate the signal with extra generic frontal sharp transients instances, and also with new spikes. By using the new profile, due to the subclass relationship, SLAM purposefully hides the differences between specific kinds of frontal sharp transients.

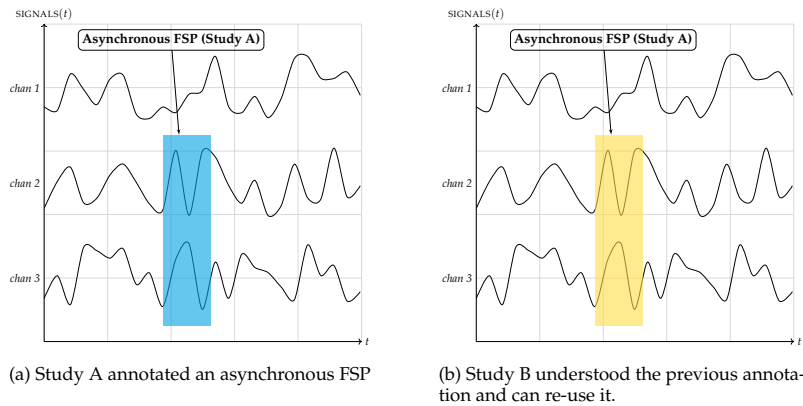


Figure 6: An asynchronous FSP is re-used from study A to study B.

If the file is updated by study B and subsequently revisited through study

A, the existing annotations are still recognized correctly for study A (they appear again as “asynchronous FSP (study A)”), and the new annotations are ignored, until the profile for study A is amended.

4. Impact

The impact of the proposed software tool is, at least, twofold. First, in a clinical setup, SLAM facilitates collaborative analysis of electrophysiological data. This is especially true in the context of multicentric research. Indeed, it is already involved in the analysis of electroencephalographic data of the LyTONEPAL study [29]. This observational study has collected extensive, clinical data of 794 newborns with anoxic-ischemic encephalopathy in France from September 2015 to March 2017 [29]. These patients underwent EEG monitoring, leading to massive, multicentric amount of biosignal data presenting challenging properties, e.g. in terms of heterogeneousness and quality of the signal. An in-depth analysis of these data aims at identifying specific features stronger classification, in order to understand which of these features may be used as biomarkers, helping in care and prognosis of these newborns. In this context, our tool is of precious use to the clinicians, on the one hand to carry out annotations and on the other hand to share and homogenize the analyses performed in each center. User feedback from partnered clinicians has been integrated in an iterative development process in order to adapt its interface and functionality, with discussions about ergonomics, features, and the demand for interoperability.

Second, the annotation functionalities offered by SLAM may be of precious use for building ex-nihilo and/or collecting and unifying some annotations that may be further involved in machine learning / deep learning methods and tools dedicated to biosignal analysis. Indeed, over the recent years, many efforts were geared towards the development of efficient approaches for pattern / artifacts detection and/or classification. By contrast, fewer efforts were dedicated to develop tools allowing to feed these tools with accurate annotated data. SLAM provides a contribution to tackle this important issue, that may be hopefully useful in conjunction with data analysis tools.

5. Conclusion and perspectives

In this article, we proposed SLAM, a software tool that allows user-friendly, interactive visualization and annotation of EEG data in the context of a clinical study with the LyTONEPAL cohort [29], while minimizing manual interactions. It is designed as a thin-client that can be run on a web interface. In particular, it relies on an extensible data exchange format for storing (biosignal) data, associated metadata and annotations. A semantic web paradigm is considered for metadata modeling, which allows to aggregate independent features from different sources, users, and to reuse information from different experimental protocols.

SLAM provides solutions for tackling the issue of biosignal data annotation, which is a mandatory prerequisite to the development of supervised machine learning / deep learning approaches for high-level analysis, e.g. pattern detection, segmentation and/or classification. In particular, it may be enriched and/or interfaced with analysis modules / libraries in order to design end-to-end solutions for biosignal analysis in actual clinical contexts.

Security and privacy measures must be taken into account when using SLAM over a network or with sensitive private data. It is expected to be used as part of a secure system in an isolated network environment.

Further works will consist of developing specific machine learning modules dedicated to EEG analysis, a task that has already proven useful for EEG data processing [1, 2], scalability and performance of SLAM and integration with existing systems dedicated to the processing of medical data such as Resonant/Girder [30, 31] and 3DSlicer [32]. Future works will also investigate the feasibility to annotate a patient record across modalities (such as MRI) with the help of linked data, potentially in a real-time collaboration environment.

SLAM is not fundamentally tied to the EEG modality, and can work with other 1-D biosignals with a dedicated reading profile. More generally, thanks to the use of HDF5, it should be possible to include different data modalities within the same annotation framework.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

Acknowledgements

This work was supported by the French *Agence Nationale de la Recherche* (grants ANR-15-CE23-0009, ANR-19-CHIA-0015, ANR-22-CE45-0034) and by the American Memorial Hospital Foundation (AMHF).

References

- [1] I. Zubarev, G. Vranou, and L. Parkkonen. MNEflow: Neural networks for EEG/MEG decoding and interpretation. *SoftwareX*, 17:100951, 2022. doi: 10.1016/j.softx.2021.100951.
- [2] L. Cabañero-Gómez, R. Hervás, I. González, and L. Rodríguez-Benitez. eeglib: A python module for EEG feature extraction. *SoftwareX*, 15:100745, 2021. doi: 10.1016/j.softx.2021.100745.

- [3] Bingtao Zhang, Tao Lei, Hong Liu, and Hanshu Cai. Eeg-based automatic sleep staging using ontology and weighting feature analysis. *Computational and Mathematical Methods in Medicine*, 2018:6534041, Sep 2018. ISSN 1748-670X. doi: 10.1155/2018/6534041.
- [4] Konstantin Ryabinin, Svetlana Chuprina, and Ivan Labutin. Ontology-driven toolset for audio-visual stimuli representation in eeg-based bci research. In *GraphiCon'2021 Proceedings*, pages 223–234, 01 2021. doi: 10.20948/graphicon-2021-3027-223-234.
- [5] Satya S. Sahoo, Katja Kobow, Jianzhe Zhang, Jeffrey Buchhalter, Mojtaba Dayyani, Dipak P. Upadhyaya, Katrina Prantzalos, Meenakshi Bhattacharjee, Ingmar Blumcke, Samuel Wiebe, and Samden D. Lhatoo. Ontology-based feature engineering in machine learning workflows for heterogeneous epilepsy patient records. *Scientific Reports*, 12(1):19430, Nov 2022. ISSN 2045-2322. doi: 10.1038/s41598-022-23101-3.
- [6] Marcelli Wac, Raul Santos-Rodriguez, Chris McWilliams, and Christopher Bourdeaux. Cats: Cloud-native time-series data annotation tool for intensive care. *SoftwareX*, 24:101593, 2023. ISSN 2352-7110. doi: 10.1016/j.softx.2023.101593a.
- [7] Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. Label Studio: Data labeling software, 2020-2022. URL <https://github.com/heartexlabs/label-studio>. Multi-type data labeling and annotation tool with standardized output format; Accessed 2024-04-23.
- [8] Juan Haladjian. The Wearables Development Toolkit (WDK), 2019. URL <https://github.com/avenix/wdk>. A development environment for activity recognition applications with sensor signal; Accessed 2024-04-23.
- [9] Geocene Inc. Trainset. URL <https://trainset.geocene.com>. Graphical tool for labeling time series data; Accessed 2024-04-23.
- [10] Bogdan Ruszczak, Krzysztof Kotowski, Jacek Andrzejewski, Christoph Haskamp, and Jakub Nalepa. Oxi: An online tool for visualization and annotation of satellite time series data. *SoftwareX*, 23:101476, 2023.
- [11] Microsoft Inc. Taganomaly. URL <https://github.com/microsoft/taganomaly>. Anomaly detection labeling tool, specifically for multiple time series; Accessed 2024-04-23.
- [12] Paul Brossier, Tintamar, Eduard Müller, Nils Philippsen, Tres Seaver, Hannes Fritz, cyclopsian, Sam Alexander, Jon Williams, James Cowgill, and Ancor Cruz. Aubio: a library for audio and music analysis. February 2019. doi: 10.5281/zenodo.2578765.

- [13] Nalan Ozkurt. Prepannecg: A user friendly matlab ecg preprocessing and annotation gui for health professionals. *SoftwareX*, 24:101589, 2023. ISSN 2352-7110. doi: doi.org/10.1016/j.softx.2023.101589.
- [14] Roger Selzler, Adrian D. C. Chan, and James R. Green. Tsea: An open source python-based annotation tool for time series data. In *2021 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–6, 2021. doi: 10.1109/MeMeA52024.2021.9478712.
- [15] Daniele Di Mitri, Jan Schneider, Roland Klemke, Marcus Specht, and Hendrik Drachler. Read between the lines: An annotation tool for multimodal data for learning. In *Proceedings of the 9th international conference on learning analytics & knowledge*, pages 51–60, 2019. doi: 10.1145/3303772.3303776.
- [16] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7:267, 2013. doi: 10.3389/fnins.2013.00267.
- [17] R. Weiler, M. Diachenko, E. L. Juarez-Martinez, A.-E. Avramiea, P. Bloem, and K. Linkenkaer-Hansen. Robin’s Viewer: Using deep-learning predictions to assist EEG annotation. *Frontiers in Neuroinformatics*, 16, 2023. doi: 10.3389/fninf.2022.1025847.
- [18] C. Chen, Z. Mei, and Z. Huang. A clinical EEG research platform to support progressive model construction. In *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Proceedings*, pages 1–6, 2022. doi: 10.1109/CISP-BMEI56279.2022.9979832.
- [19] M. Beier, T. Penzel, and D. Krefting. A performant web-based visualization, assessment, and collaboration tool for multidimensional biosignals. *Frontiers in Neuroinformatics*, 13, 2019. doi: 10.3389/fninf.2019.00065.
- [20] Charles G. Willis, Edith Law, Alex C. Williams, Brian F. Franzone, Rebecca Bernardos, Lian Bruno, Claire Hopkins, Christian Schorn, Ella Weber, Daniel S. Park, and Charles C. Davis. Crowdcurio: an online crowdsourcing platform to facilitate climate change studies using herbarium specimens. *New Phytologist*, 215(1):479–488, 2017. doi: doi.org/10.1111/nph.14535.
- [21] L. Dong, Y. Zhang, L. Zhao, T. Zheng, W. Wang, J. Li, D. Gong, T. Liu, and D. Yao. DRT: A new toolbox for the standard EEG data structure in large-scale EEG applications. *SoftwareX*, 17:100933, 2022. doi: 10.1016/j.softx.2021.100933.
- [22] David J. Brooks, Peter J. Hunter, Bruce H. Smaill, and Mark R. Titchener. Biosignalm1 — a meta-model for biosignals. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5670–5673, 2011. doi: 10.1109/IEMBS.2011.6091372.

- [23] Natus Medical Incorporated. Deltamed coherence, . URL <https://medicatech-dz.com/produit/solutions-deeg-natus-neuroworks>. An EEG software for Deltamed LTM system; Accessed 2024-05-07.
- [24] Natus Medical Incorporated. Brain quick eeg, . URL <https://natus.com/neuro/brain-quick-eeg-software>. Software for EEG review and analysis powers the easy navigation of routine EEG, long-term monitoring (LTM), ICU monitoring, ambulatory, and research studies; Accessed 2024-04-23.
- [25] Arnaud Delorme and Scott Makeig. Eeglab: an open-source toolbox for analysis of single-trial eeg dynamics. *Journal of Neuroscience Methods*, 134: 9–21, 2004. doi: 10.1016/j.jneumeth.2003.10.009.
- [26] Clemens Brunner. Mnelab: a graphical user interface for mne-python. *Journal of Open Source Software*, 7(78):4650, 2022. doi: 10.21105/joss.04650.
- [27] V. Kraus and G. Dollé. Adftool. URL <https://github.com/neonatool/adftool>. An implementation of the Allotrope Data Format for storage of biosignals; Accessed 2024-04-23.
- [28] D. Della Corte, W. Colman, H. Fessenmayr, A. Sawczuk da Silva, and D. E. Vanderwall. Self-reporting data assets and their representation in the pharmaceutical industry. *Drug Discovery Today*, 27:207–214, 2022. doi: 10.1016/j.drudis.2021.07.019.
- [29] T. Debillon, N. Bednarek, A. Ego, and LyTONEPAL Writing Group. LyTONEPAL: Long term outcome of neonatal hypoxic encephalopathy in the era of neuroprotective treatment with hypothermia: A French population-based cohort. *BMC Pediatrics*, 18:255, 2018. doi: 10.1186/s12887-018-1232-6.
- [30] Kitware Inc. Resonant, . URL <https://resonant.kitware.com/>. A platform to provide storage, analysis, and visualization solutions for your data; Accessed 2024-04-23.
- [31] Kitware Inc. Girder, . URL <https://girder.readthedocs.io>. A data management platform; Accessed 2024-04-23.
- [32] Ron Kikinis, Steve D. Pieper, and Kirby G. Vosburgh. *3D Slicer: A Platform for Subject-Specific Image Analysis, Visualization, and Clinical Support*, pages 277–289. Springer New York, New York, NY, 2014. ISBN 978-1-4614-7657-3. doi: 10.1007/978-1-4614-7657-3_19.