The Journal of Open Source Software

MEG Chicken: Interactive Artifact Detection Training for MEG and EEG data

Kirsten Petras [1*], **Yue Kong** [1*], **Hayley Yingxuan Wu** [1*], and **Laura Dugué** [1,2]

1 Université Paris Cité, INCC UMR 8002, CNRS, F-75006 Paris, France **2** Institut Universitaire de France (IUF), Paris, France ***** These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Statement of Need

Detecting and processing signal artifacts is crucial for analyzing neural time-series data from magnetoencephalography (MEG) or electroencephalography (EEG) recordings (Luck & Kappenman, 2017). While numerous automated and semi-automated artifact detection algorithms exist (e.g., (Jas et al., 2017)), visual inspection and manual labeling remain the most widely used methods for identifying components contaminated by eye movements, muscle activity, or electrical noise. Despite excellent resources describing common physiological and electrical artifacts in MEG and EEG data (Burgess, 2020; Urigüen & Garcia-Zapirain, 2015), decisions about segment or component rejection are ultimately subjective. This subjectivity often results in inconsistencies, particularly when training new lab members.

Implicit or procedural learning refers to the acquisition of skills and knowledge through repeated exposure and practice, without explicit instruction. A well established example of implicit rule learning is chicken sexing, where workers learn to distinguish the sex of day-old chicks based on subtle visual cues. Despite often not being able to articulate which exact features distinguish between male and female chicks, experienced chicken sexers can classify chicks accurately and reliably through extensive experience and feedback (Horsey, n.d.).

`MEG Chicken` uses the principle of implicit learning through immediate feedback to streamline and standardize the process of learning to detect artifacts in electrophysiology data. The open source software tool presents trainees with data containing various types of artifacts and provides immediate feedback on their decisions, enabling consistent rejection criteria to be learned implicitly. Labs can customize the training with their own annotated datasets to ensure alignment with lab-specific standards.

Functionality

`MEG Chicken` is built on the MNE-Python library and employs MNE's user-friendly graphical interface for visualizing time-series data and sensor topographies. The software includes labeled example datasets available for download via Zenodo, and labs can import or create their own labeled datasets through the interactive interface.

The training program includes modules for:

1. **Bad Channel Rejection**
2. **ICA Component Selection**
3. **Labeled Data Import**
4. **Label Creator**

Participants complete the training at their own pace, progressing until they consistently make correct decisions over a predefined number of trials. The full visualization capabilities of MNE are available in both modules, augmented with information slides.

Immediate feedback after each decision supports implicit learning in the absense of explicit rules.

Evaluation and Performance

Testing on 5 observes, naive to MEG and EEG artefacts, showed: - Participants required an average of xx minutes to achieve xx% accuracy in bad channel selection. - Participants required an average of xx minutes to achieve xx% accuracy in ICA component selection. - Performance remained consistent in follow-up tests conducted several days later.

Table:

Task	Average Time to Mastery	Final Accuracy	Retest Accuracy
Bad Channel Rejection	xx minutes	xx%	xx%
ICA Component Selection	xx minutes	xx%	xx%

Funding

This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 852139 - Laura Dugué).

Acknowledgments

We thank Laetitia Grabot for providing the example dataset, and Coline Haro for contributions to an earlier version.

Toolbox Dependencies

- `os`
- `time`
- `random`
- `json`
- `numpy`
- `mne`
- `tkinter`
- `pickle`
- `csv`
- `matplotlib`
- `warnings`
- `scipy`
- `PIL`
- `playsound`
- `copy`

References

Burgess, R. C. (2020). Recognizing and correcting MEG artifacts. *Journal of Clinical Neurophysiology*, *37*(6), 508–517.

Horsey, R. (n.d.). *The art of chicken sexing*. https://web-archive.southampton.ac.uk/cogprints.org/3255/1/chicken.pdf

Jas, M., Engemann, D. A., Bekhti, Y., Raimondo, F., & Gramfort, A. (2017). Autoreject: Automated artifact rejection for MEG and EEG data. *NeuroImage*, *159*, 417–429.

Luck, S. J., & Kappenman, E. S. (2017). *Electroencephalography and event-related brain potentials*.

Urigüen, J. A., & Garcia-Zapirain, B. (2015). EEG artifact removal—state-of-the-art and guidelines. *Journal of Neural Engineering*, *12*(3), 031001.