Automated Data Cleaning for the Muse EEG

Arnaud Delorme
CerCo CNRS, Paul Sabatier University, Toulouse, France
SCCN, INC, UCSD, La Jolla CA, USA
https://orcid.org/0000-0002-0799-3557

Jeffery A. Martin
Center for the Study of Non-Symbolic Consciousness, Newport, KY, USA
Lab. for Consciousness Science, CIHS, Carlsbad, CA, USA
Transformative Technology Lab, Stanford, CA, USA
School of Medicine, Stanford University, Stanford, CA, USA
https://orcid.org/0000-0003-4158-9571

2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-0126-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIBM52615.2021.9669415

***Abstract*—Wearable EEG headsets have transformed the landscape of EEG research. It is no longer necessary to use expensive equipment and over 30 minutes of preparation time to collect EEG data. Instead, participants may do it themselves in a few minutes from the comfort of their home. Confronted with processing thousands of such recordings, manual data cleaning has become a bottleneck, so we tested automated methods for cleaning data. To validate these methods, we asked three trained EEG human raters to clean 100 files of 12 minutes each, and use these manual rejections to assess the performance of automated cleaning methods for both channel rejections and continuous data rejections. We showed that rejecting channels based on abnormal spectrum yielded the best results. We also showed that the Artifact Subspace Reconstruction rejection method was the best method to reject continuous portions of data. Inter-rater consistency is the gold standard to assess the quality of automated data rejection methods, and we showed that our best rejection methods were not significantly different and might even outperform human raters. We provide a simple recipe and plugin for the popular EEGLAB software for automated data cleaning of Muse data. We hope this new tool will allow more widespread use of wearable EEG in clinical and research settings where large quantities of wearable EEG data need to be processed.**

Keywords— Muse, EEG, data cleaning, artifact subspace reconstruction, inter-rater consistency

I. Introduction

Wearable EEG, ranging in a couple of hundred dollars have transformed the landscape of EEG research. The data quality of such devices has proven of sufficient quality to collect EEG in various conditions, including continuous EEG [1] and event-related potentials [2]. Some of these wearable systems, like the Muse, use active electrodes similar to the one available on research-grade high-density EEG systems [3]. While the number of channels of wearable EEG systems remains limited (4 channels for the Muse), because of volume conduction, most EEG channels record activity from the whole brain. Therefore, for some use cases, a few channels might be sufficient to draw conclusions regarding the frequency content of the EEG and by inference some cognitive mechanisms. While wearable EEG will not replace the use of professional-grade EEG systems anytime soon, it allows collecting data repetitively on a large quantity of participants, in our case several thousand recordings. This feat is not possible with professional-grade EEG systems; because of long preparation time and expenses associated with collecting data, most experimental studies include less than 32 participants.

Thousands of recordings bring new challenges to process data. With a few dozen recordings, it is usually possible to manually reject artifacts. However, this is no longer possible with thousands of recordings. Methods have been developed to automatically reject artifacts [4,5,11,12,13]. However, first these methods are not compared to human inter-rater reliability so it is difficult to assess their actual performance compared to humans. Second, they are designed for high density high quality EEG recording: they are not properly validated for wearable EEG where the number of channels is reduced and the data quality is usually lower than research-grade systems, especially when participants are asked to fit the wearable headset by themselves.

Here, we validate automated data rejection methods applied to wearable EEG, based on the manual rejection of three EEG human raters. Inter-rater consistency remains the gold standard for assessing automated rejection methods, so we compared how various rejection methods matched manual rejections.

II. Methods

A. Data collection

123 participants self-enrolled in a 4-month long meditation course, for which they were asked to collect their EEG brainwave on a daily basis with the Muse headset [9]. Participants signed consent forms and the study was approved by the Center for the Study of Non-Symbolic Consciousness Ethical Review Board.

Data collection was performed with the 4-channel Muse 1 headset at 220 Hz sampling rate and a custom smartphone app developed by NeuroMore Inc. saved the raw time series (this private app was designed for the meditation course and is not available for public download). Participants went through a

short video training for how to collect EEG data but were not trained by research staff on an individual basis. Each participant was assigned a unique ID, the data was saved as a custom *.mat* MATLAB file, and automatically sent to a Dropbox cloud server. The 123 participants contributed 8,455 EEG sessions amounting to 58.9 Gb of data. The number of sessions per participant ranged between 1 and 114, with most participants contributing between 40 and 110 sessions. 7,331 sessions contained more than 30 seconds of data, and 4,881 of these files contained raw EEG data - because of software configuration issues, some sessions only contained EEG spectral estimates computed by the Muse headset.

B. Data selection for manual data cleaning

Out of the 4,881 raw EEG data files, we randomly selected 100 files of a duration of about 12 minutes (average of 12 minutes and 28 seconds with 56-second standard deviation).

C. Manual data cleaning

Three human raters (first author AD, CB, and KM; see acknowledgments), trained in EEG research, cleaned 89 files by visual inspection of the EEG data (volunteer KM only cleaned 89 of 100 files, and files that were not cleaned by all human raters were not analyzed). Human raters first imported the data into the EEGLAB software [6], using a custom-made EEGLAB plugin that imported the raw EEG data, removing the DC offset (the average value from each channel), and low-pass filtered each channel at 40 Hz (FIR linear filter *pop_eegfiltnew* of EEGLAB, 75 points, 10 Hz transition bandwidth, 40 Hz passband edge, and 45 Hz cutoff frequency (-6 dB)).

Human raters first identified bad channels based on their spectrum and raw data traces. A transient artifact in an EEG channel did not automatically qualify it as bad, because bad data regions might be removed at a subsequent stage (see below). However, an EEG channel with many transient artifacts throughout the recording might qualify as bad: this decision was left to the best judgment of the human rater. Figure 1 shows examples of artifactual channels and data regions.

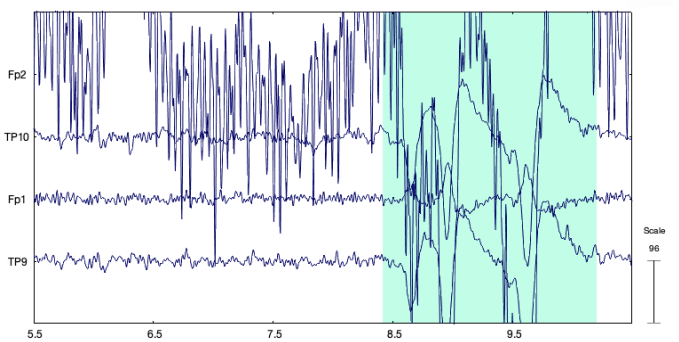

Figure 1. Example of Muse EEG data with artifacts labeled by one human rater. The top channel is bad throughout the entire recording (of which 5 seconds are shown here). The green highlighted region also corresponds to a bad data region. Time is indicated in seconds and the amplitude scale is in μV.

Bad regions of data were selected based on the raw spectral traces. The percentage of data rejected differed for each human rater, ranging from an average of 35% to 46% of the data (Table 1). The three human raters were asked to use their best judgment and expertise to visually inspect the data, and label bad channels and bad portions of data. Cleaning all the files amounted to a total of 10 to 20 hours for each human rater, spread out over several days. The three human raters who cleaned the data will be referred to in the rest of this manuscript as R1, R2, and R3.

D. Algorithm performance assessment

We randomly partitioned the 89 datasets into two sets containing about 50% of the data: a training set (n=44) and a validation set (n=45). The training set was used to identify the optimum algorithm and the ideal parameters for that algorithm. The validation set was used to assess the performance of the optimum algorithm on new data. Because we tested many algorithms and many parameters for these algorithms, it is possible that we overfitted the data, and found some parameters and algorithms that happened to closely match manual rejection, but might not generalize to new data. The use of a validation set ensures that the chosen algorithm has high performance on new data.

For assessing algorithm performance, we computed algorithm accuracy ranging from 0 (no match) to 1 (perfect match). Accuracy is the percentage of correct classification and combines the true positive and true negative rates. We computed the match between pairs of human raters in the same fashion.

For channel rejection performance of a given dataset, a perfect match (accuracy of 1) would mean that the same channels are selected for rejection by two different methods (for example a specific human rater and a specific algorithm). Accuracy was then averaged across datasets.

For continuous data rejection, an accuracy of 1 would mean that the exact same samples (out of about 160,000 samples corresponding to 12 minutes at 220 sampling rate) would be selected for rejection - which is extremely unlikely. The accuracy thus estimated overlapping regions of rejection. When comparing a given automated method for rejecting data with a human rater, with the human rater considered as ground truth (note that accuracy calculation is symmetrical so it is not important which method is considered as ground truth), then accuracy is equal to the true positive rate (TPR) plus the true negative rate (TNR). For example, say both methods reject 30% of the data, and the methods overlap on 20% of the rejected data. The true positive rate is 0.2, the false positive rate (FPR) is 0.1, the false-negative rate is 0.1 (FNR), and the true negative rate is 0.6, leading to an accuracy of TPR+TNR=0.8 (which is also equal to 1-FPR-FNR).

For channels, we considered the error rate (1 minus accuracy) and for continuous rejection, we considered accuracy directly. This is because channel rejections are rare events, and it is easier to read a table (Table 1) with single-digit errors, usually below 10%.

In this paper, we do not consider *specificity*, *precision* or the *F1 metric*, which are other important concepts in signal detection theory. This is because, for each dataset, we need to combine algorithm performance calculated against several human raters. Accuracy, which combines performance on common rejected data portions and common clean regions is the most natural metric to use. For each dataset, the accuracy performance of a given automated rejection method is calculated for each human rater and then averaged across raters.

Note that for automated rejection of continuous data, we first applied the optimum algorithm for channel rejection. This algorithm rejected all data channels for 1 of the 89 datasets. Because it was not possible to assess the performance on that dataset, for continuous rejection only, we considered 44 datasets for the training set and 44 for the validation set.

E. Automated channel rejection methods

We considered several methods for artifact rejection. For each method, we scanned the parameter space (grid-search) and identified the parameter that best matched manual rejection on the training set.

- *Channel correlation.* We used the *clean_channels_nolocs* function of the *Clean_rawdata* v2.5 [8] plugin of EEGLAB. We varied the correlation threshold parameter from 0.2 to 0.3 in 0.01 increments. We set the *IgnoredQuantile* parameter to 0.05 and used the default values for other parameters.

- *Channel standard deviation.* We computed the standard deviation of each channel raw signal and set a threshold ranging between 1 and 200 microvolts in increments of 5 microvolts. When a given channel standard deviation exceeded the threshold, it was labeled for rejection.

- *Spectral thresholding.* We computed the log-power spectrum of each channel using the *pop_spectopo* function of EEGLAB which uses the *pwelch* function of MATLAB (window and discrete Fourier transform length of 1 second with an overlap of 0 seconds). Different frequency bands were considered 0-5 Hz, 5-15 Hz, 15-25 Hz, 25-35 Hz, 35-45 Hz, 45-55 Hz, 0-55 Hz, 5-55 Hz, 15-55 Hz. For each frequency band, we scanned an array of threshold values ranging from 10 to 50 $\log_{10}(\mu V^2)$/Hz in increments of 1. We disabled normalization and used default values for other parameters. Usually, with high-density montage, the *pop_rejchan* function normalizes measures across channels allowing to set threshold in terms of the standard deviation of the spectral measure. However, with only 4 channels, it is not possible to normalize measures across channels to reject a few outliers, so the absolute spectral measure value had to be used. When a given channel measure value exceeded the threshold, it was labeled for rejection.

For each algorithm, we took the best rejection performance. We verified that for each algorithm, the best performance corresponded to a maximum.

We also tried rejecting data channels based on *probability* and *kurtosis* using the *pop_rejchan* function of EEGLAB. These functions failed, the probability function because it is not possible to assess improbable channels using only 4 channels, and the Kurtosis function because absolute Kurtosis tends to vary widely across datasets, not allowing to set a common threshold across dataset (the best performance we obtained with Kurtosis was when the threshold was high and no data was rejected).

F. Automated continuous rejection methods

We considered the three continuous rejection methods.

- *Artifact Subspace Reconstruction (ASR).* We used the *clean_artifact* function of the *Clean_rawdata* v2.5 [8] plugin of EEGLAB. We varied the *WindowCriterionTolerances* argument from 5 to 15 in increments of 1, set the *WindowCriterion* parameter to 0 (to automatically reject bad portions of data instead of trying to correct them), and disabled all other features including *BurstRemoval*. We used default values for filtering parameters.

- *Spectral thresholding.* We use the *pop_rejcont* function of EEGLAB to reject artifacts in the 5-55 Hz frequency range with a threshold ranging from 5 to 15 $\log_{10}(\mu V^2)$. All other parameters were left as default. We informally tried other frequency ranges as well, including some up to 120 Hz, but this did not increase performance. Given that 5-55 Hz was the best frequency range for removing bad channels, it was deemed appropriate for continuous data rejection.

- *Data limits.* We used the *pop_continuousartdet* function of ERPLAB (v8.10) [7] which rejects data exceeding pre-defined amplitude limits (*ampth* parameter). We set the threshold from 50 to 110 μV in increments of 5. All other parameters were left as default, including the bandpass filter parameters - not set by default. Using the bandpass parameter would be similar to using the *pop_rejcont* spectral thresholding function described above.

For each algorithm, we use the best rejection performance. We verified that for each algorithm, the best performance corresponded to a maximum.

III. Results

We first assess performance on channel rejection, then on continuous data rejection after bad channels had been removed.

A. Automated channel rejection

Table 1 shows the performance for different channel artifact rejection methods (See Methods). Several spectral thresholding methods have similar performance. The best

method is spectral thresholding between 5 and 55Hz with a threshold of 25 $\log_{10}(\mu V^2)$/Hz (see Methods). This method also has a wide frequency range which makes it more robust to changes at specific frequencies. The channel correlation method is not significantly better than no rejection (the 95% confidence intervals for the two rejections overlap). The standard deviation method performed significantly better than no rejection.

	# rejections	vs R1	vs R2	vs R3	Average error
R1	31	n/a	9.4	6.5	8
R2	45	9.4	n/a	9.9	9.7
R3	42	6.5	9.9	n/a	8.2
No rejection	0	17.6	25.6	23.9	22.3 (18.8-26.2)
Correlation	18	14.5	19.9	18.5	17.6 (14.4-20.8)
Standard-dev	30	8.8	11.9	11.6	10.8 (8.1-13.3)
0-5 Hz	16	13.1	17.9	16.2	15.7 (12.7-18.7)
5-15 Hz	29	10.8	11.4	14.2	12.1 (9.3-14.8)
15-25 Hz	29	5.7	9.4	9.9	8.3 (5.9-10.3)
25-35 Hz	31	5.7	9.4	9.9	8.3 (6.0-10.6)
35-45 Hz	30	5.4	9.7	9.4	8.1 (5.7-10.2)
45-55 Hz	35	8.5	10.5	9.9	9.7 (7.0-11.9)
0-55 Hz	17	12.5	17	15.6	15.1 (12.1-18.2)
5-55 Hz	37	7.1	7.1	9.9	8.0 (5.5-10.0)
15-55 Hz	32	6.2	8.5	10.5	8.4 (6.1-10.6)

Table 1. Channel automated rejection methods' performance on the training set. R1, R2, R3 are human raters 1, 2, and 3. Different rejection methods are shown on each row and compared with the rejection of R1, R2, and R3 in columns 3, 4, and 5. "# rejections" indicates the total number of channels rejected across all datasets of the training set. The last column shows the average error for a given method (averaging errors obtained against R1, R2, and R3) as well as the 95% confidence interval in parenthesis obtained by bootstrap. The different rejection methods on each row are: no rejection, rejection based on channel correlation, standard deviation, and spectral thresholding in different frequency bands. For each method, we scan a collection of parameter values, and the best performance is retained (see Methods). The best method is spectral thresholding between 5-55 Hz with a threshold of 25.

We then assessed the performance of the best method on the validation data. The selected rejection method achieved lower inter-rating error compared to all human participants, although with only 3 human raters it was not possible to assess significance. Because human rater average error is included in the selected automated channel rejection method's 95% confidence intervals, this method is not significantly worse than a human rater (if different, it is likely better).

	# rejections	vs R1	vs R2	vs R3	Average error
R1	37	n/a	5.3	10.1	7.7
R2	40	5.3	n/a	10.6	7.9
R3	53	10.1	10.6	n/a	10.3
5-55 Hz	38	4.4	6.9	1065	7.3 (5.0-9.2)

Table 2. Best automated channel rejection method performance on the validation set. Only the best method from Table 1 is applied to the validation set. See Table 1 for additional column and row descriptions.

B. Automated continuous data rejection

Table 3 shows the performance for different continuous data artifact rejection methods (See Methods). Inter-rater reliability of human raters was comparable to those reported in the literature [10]. The ASR method performed significantly better than all other rejection methods.

	% rejection	vs R1	vs R2	vs R3	Accuracy
R1	43	n/a	78.7	77.9	78.3
R2	46	78.7	n/a	75.9	77.3
R3	35	77.9	75.9	n/a	76.9
No rejection	0	56.9	55.4	66.1	59.5 (55.4-63.0)
ASR (11)	40	80.1	77.8	77.4	78.4 (76.2-80.6)
Spec (10)	28	72.7	70.9	75.9	73.2 (71.1-75.2)
Data limit (60)	51	68.1	65.3	65.6	66.3 (62.6-70.1)

Table 3. Continuous data automated rejection methods' performance on the training set. R1, R2, R3 indicate human raters 1, 2, and 3. Different rejection methods are shown and compared with the rejection of R1, R2, and R3 in columns 3, 4, and 5. "% rejection" indicates the percentage of rejected data, and the last column shows the average accuracy of a given method (averaging accuracy obtained against R1, R2, and R3) as well as the 95% confidence interval in parenthesis. The methods to reject data are: no rejection, rejection-based ASR, and spectral thresholding in the 5-55 Hz frequency bands, and a method testing if the data exceeds pre-defined limits. The number in parenthesis after the algorithm name indicates the optimal parameter value (see Methods). The best method is the ASR method with a parameter of 11.

We then assessed the performance of the best rejection method on the validation set. The ASR rejection method achieved lower inter-rating error compared to all human participants, although with only 3 human raters, it was not possible to assess significance. 95% confidence intervals indicate that the automated method is not significantly worse than a human rater (if different, it is likely better).

	% rejection	vs R1	vs R2	vs R3	Accuracy
R1	37	n/a	80.7	79.7	80.2
R2	37	80.7	n/a	78	79.4
R3	30	79.7	78	n/a	78.8
ASR (11)	31	81.9	82.4	80.6	81.6 (79.5 83.6)

Table 4. Best automated continuous data rejection method performance on the validation set. Only the best method from Table 3 is applied to the validation set. See Table 3 for additional column and row descriptions.

C. EEGLAB plugin

We have updated the Muse import plugin (*MuseMonitor* v4.0) to include parameters for automated rejection (Figure 2). Upon importing files, users have the choice to automatically remove bad channels and bad portions of data with the optimal parameters found in this report. Note that users may also use this function from the command line for automated processing on thousands of data files.

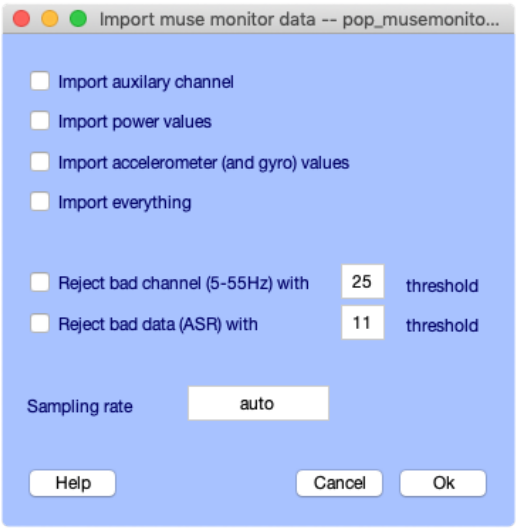

Figure 2. The EEGLAB Muse import plugin (MuseMonitor v4.0) was modified to include options to automatically reject bad channels and bad portions of data based on the methods and parameters found in this report.

IV. Discussion

We have shown that it is possible to find automated data rejection methods that were not significantly worse than human manual rejections. If different, the performance of these algorithms was likely better than human raters, because their rejections were closer to all human raters than raters' rejections were with each other.

It is interesting to note that standard methods for high-density EEG, such as the correlation of neighboring channels, are inefficient with low-density montages. This is because these methods assume a few outliers in a large number of channels. We also tried independent component analysis (ICA) [4] on a few datasets and obtained poor results. ICA performs well on high-density montages, and there is no theoretical reason why it should not perform well with 4 channels, so this requires further investigation.

The validated data rejection tools we presented will allow the automated processing of large numbers of data files. It is possible that automated methods based on machine learning (such as support vector machines, random forest, or deep learning) could achieve even higher performance. However, the standard and validated methods we used have the advantage of simplicity. In addition, the high performance on the validation data shows that, because we only fitted one parameter, these methods are robust to overfitting, which might not be the case for machine learning methods.

Acknowledgment

The authors wish to thank Claire Braboszcz and Karalee Marie, for their assistance in cleaning the data.

References

[1] Wilkinson CM, Burrell JI, Kuziek JWP, Thirunavukkarasu S, Buck BH, Mathewson KE. "Predicting stroke severity with a 3-min recording from the Muse portable EEG system for rapid diagnosis of stroke." Sci Rep. 2020;10(1):18465. 2020 Oct 28. doi:10.1038/s41598-020-75379-w

[2] Krigolson OE, Williams CC, Norton A, Hassall CD, Colino FL. "Choosing Muse: Validation of a Low-Cost, Portable EEG System for ERP Research." Front Neurosci. 2017;11:109. 2017 Mar 10. doi:10.3389/fnins.2017.00109

[3] Cannard C, Brandmeyer T, Wahbeh H, Delorme A. "Self-health monitoring and wearable neurotechnologies." Handb Clin Neurol. 2020;168:207-232. doi:10.1016/B978-0-444-63934-9.00016-0

[4] Delorme A, Sejnowski T, Makeig S. "Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis." Neuroimage. 2007;34(4):1443-1449. doi:10.1016/j.neuroimage.2006.11.004

[5] Nolan H, Whelan R, Reilly RB. "FASTER: Fully Automated Statistical Thresholding for EEG artifact Rejection." J Neurosci Methods. 2010;192(1):152-162. doi:10.1016/j.jneumeth.2010.07.015

[6] Delorme A, Makeig S. "EEGLAB: an open-source toolbox for analysis of single-trial EEG dynamics including independent component analysis." J Neurosci Methods. 2004;134(1):9-21. doi:10.1016/j.jneumeth.2003.10.009

[7] Lopez-Calderon J, Luck SJ. ERPLAB: an open-source toolbox for the analysis of event-related potentials. Front Hum Neurosci. 2014;8:213. Published 2014 Apr 14. doi:10.3389/fnhum.2014.00213

[8] Mullen, T. R., Kothe, C. A. E., Chi, Y. M., Ojeda, A., Kerth, T., Makeig, S., et al., "Real-time neuroimaging and cognitive monitoring using wearable dry EEG." IEEE Trans. Bio-Med. Eng, 2015. 62(11), 2553–2567.

[9] Martin, J. A., Ericson, M., Berwaldt, A., Stephens, E. D., & Briner, L. (2021). Effects of two online positive psychology and meditation programs on persistent self-transcendence. Psychology of Consciousness: Theory, Research, and Practice. Advance online publication. http://dx.doi.org/10.1037/cns0000286

[10] Shirk SD, McLaren DG, Bloomfield JS, et al. Inter-Rater Reliability of Preprocessing EEG Data: Impact of Subjective Artifact Removal on Associative Memory Task ERP Results. Front Neurosci. 2017;11:322. Published 2017 Jun 16. doi:10.3389/fnins.2017.00322

[11] Gabard-Durnam LJ, Mendez Leal AS, Wilkinson CL, Levin AR. The Harvard Automated Processing Pipeline for Electroencephalography (HAPPE): Standardized Processing Software for Developmental and High-Artifact Data. Front Neurosci. 2018;12:97. Published 2018 Feb 27. doi:10.3389/fnins.2018.00097

[12] Saba-Sadiya S, Chantland E, Alhanai T, Liu T, Ghassemi MM. Unsupervised EEG Artifact Detection and Correction. Front Digit Health. 2021;2:608920. Published 2021 Jan 22. doi:10.3389/fdgth.2020.608920

[13] He, B., Dai, Y., Astolfi, L., Babiloni, F., Yuan, H., & Yang, L. (2011). eConnectome: A MATLAB toolbox for mapping and imaging of brain functional connectivity. Journal of neuroscience methods, 195(2), 261–269. https://doi.org/10.1016/j.jneumeth.2010.11.015