# EPOS: EEG Processing Open-source Standardization

Corresponding Author:

Dr. Johannes Rodrigues

Julius-Maximilians-Universität Würzburg

Marcusstraße 9-11

97070 Würzburg, Germany

Phone: +49 931 31-81771

Email: johannes.rodrigues@uni-wuerzburg.de

Co-authors:

Martin Weiss

Julius-Maximilians-Universität Würzburg

Prof. Dr. Johannes Hewig

Julius-Maximilians-Universität Würzburg

Prof. Dr. John J. B. Allen

University of Arizona

OSF: https://osf.io/cw5qv/

**Abstract**

Since the replication crisis, standardization has assumed even greater importance in psychological science and neuroscience. Many widely used methods are therefore being reconsidered and the degrees of freedom these methods provide to researchers are discussed as a potential source of inconsistencies across studies. With the aim of addressing these subjectivity problems, we have been working for some time on standardizing EEG analysis in order to achieve an automated and standardized processing pipeline based on the suggested semi-automated analysis proposed by Delorme and Makeig. In this work, two scripts are now presented and explained, each containing 9 steps to perform a basic, informed ERP and frequency-domain analyses, including data export to statistical programs and visual representations of the results for validation of the results. The open-source software EEGlab in MATLAB is used as the data handling platform, but also included are scripts based on the code provided by Mike Cohen (2014). We hope that this (pre-) processing chain may offer a standardized way of analyzing data, and one that clearly explains and openly shows how the processing impacts the data, especially for beginners and newcomers in EEG-analysis. The need for standardization and replication is clearly important, yet it is equally important to look carefully into the details rather than blindly following a set of prescribed procedures. Here, we provide this tool to the community to enhance the understanding and capability of EEG-analysis, as part of one small and very basic step to enhancing rich, comprehensive and reliable methods for neuro-scientific research.

## EPOS: EEG Processing Open-source Standardization

The electroencephalogram (EEG) is one of the most important tools in both applied and clinical neurophysiology as it offers a high temporal resolution and a high safety due to its non-invasive application (Cohen, 2017b). The central properties of this measurement instrument for electrical activity, first described by Berger (1929), are frequency (oscillations per time period) and amplitude (maximum value of an oscillation during one period). Years later and despite many technical developments of the systems and the existing software for subsequent processing, there are still major problems in the replicability of findings in EEG research due to methodological variations across laboratories (Bishop, 2007). However, the problem is not limited to EEG research. Recently an article was published which shows how much flexibility in preprocessing affects the results of MRI research (Botvinik-Nezer et al., 2020), which has major implications for scientific conclusions.

The EEG signal is strongly affected by sources of interference, which are caused by the application of the electrodes (e.g., electrode displacement), the experiment itself (e.g., flickering frequencies) or by the activity of the participants, partly in interaction with the previous factors (e.g., eye-movements or muscle activity). These unwanted signals can be much larger than the actual signal of interest and therefore massively interfere with the measurement of electrophysiological correlates of neural activation if the artefacts are not corrected (e.g., Cuevas, Cannon, Yoo, & Fox, 2014). The resulting corrections and the further processing of this data raises obstacles to replicability. The replication crisis in psychophysiology was addressed by Larson and Moser (2017) in a special issue entitled "Rigor and Replication: Towards Improved Best Practices in Psychophysiological Research" in the International Journal of Psychophysiology. Included are, among other things, contributions on general improvement of rigor (Baldwin, 2017), reliability analysis of ERPs (Clayson & Miller, 2017),

replication of time-frequency data (Cohen, 2017a) or sample size calculation for electrophysiology (Larson & Carbine, 2017).

In current EEG research, there are high degrees of freedom for the researchers in terms of analysis but also in terms of reporting in publications, which leads to an increase in the false positive rate of research findings (Simmons, Nelson, & Simonsohn, 2011). Several sets of guidelines for data consistency and replicability have been published (Keil et al., 2014; Picton et al., 2000; Pivik et al., 1993), but tools for ensuring consistent processing are still needed (for the most recent approach, see Debnath et al., 2020). For EEG data, this means a flexible choice of time-window, frequency band, filtering specifications, electrodes, reference, measurement, artifact rejection, and outlier exclusion. Most researchers use different filters, references, and criteria for artefact removal prior to the actual analysis for a variety of (good) reasons. Nevertheless, this process is by no means standardized, making it almost impossible to combine data sets from different data sources for analysis without preprocessing them jointly. Consequently, Keil et al. (2014) pointed out that standardization and automation in the processing of electrophysiological data will be indispensable.

To preprocess EEG data, various pipelines have been developed in the recent past to address the growing need for standardization. The PREP pipeline (Bigdely-Shamlo, Mullen, Kothe, Su, & Robbins, 2015) provides a standardized method to remove line-noise (Mullen, 2012) and an average referencing to detect and interpolate noisy channels. However, PREP focuses only on experiment-related artifacts and not on individual artifacts like eye-blinks. The Harvard automated preprocessing pipeline (HAPPE; Gabard-Durnam, Mendez Leal, Wilkinson, & Levin, 2018) adds an independent component analysis (ICA) and uses a Multiple Artifact Rejection Algorithm (MARA; Winkler, Haufe, & Tangermann, 2011) to correct artifacts. But, according to the authors, this pipeline is not suitable for the analysis of event-related potentials. The Computational Testing for Automated Preprocessing (CTAP; Cowley, Korpela, &

Torniainen, 2017) toolbox has a similar approach to HAPPE, but allows the user to compare the outcomes of different preprocessing pipelines. Moreover, the Batch Electroencephalography Automated Processing Platform (Levin, Méndez Leal, Gabard-Durnam, & O'Leary, 2018) was created, which aims to simplify and standardize the replication of existing studies through a collection of preprocessing pipelines applied to new data sets. In addition, Automagic (Pedroni, Bahreini, & Langer, 2019) was introduced, a wrapper toolbox that combines common preprocessing methods. Automagic uses the PREP pipeline per default and adds further processing steps afterwards. The automatic pre-processing pipeline (APP; da Cruz, Chicherov, Herzog, & Figueiredo, 2018) for large datasets proved to be an efficient and reliable method for both resting state and evoked EEG, which was tested for both clinical and healthy participants. Finally, the Maryland analysis of developmental EEG (MADE) pipeline was recently published (Debnath et al., 2020). This pipeline focuses on the standardized and automatic preprocessing of data from pediatric populations using EEGLAB.

However, most of the previous mentioned pipelines focus on some specific concepts and parts of the pre-processing, while our approach tries to orient on the principles provided by Delorme and Makeig as they advised to preprocess the data up to 2019 (https://sccn.ucsd.edu/wiki/Chapter_01:_Rejecting_Artifacts). However, where Makeig and Delorme suggested semi-automatic detections or visual detections of the data, we tried to implement standardized selection criteria based on statistical outlier detection or algorithm and machine-learning based artifact selection (e.g., MARA, Winkler et al., 2011) and therefore come to replicable and standardized (pre-)processing results. Hereby, we provide examples, but other replicable usable software solutions (such as neural networks for IC detection in specific research tasks) could also be used, as long as other researchers may openly access and use the respective solutions and therefore the relevant knowledge. The aim of this paper is therefore to present a script for an EEG processing open-source standardization (EPOS), that may help to

come to openly communicated analyses that can be replicated by other researchers as all relevant information is given and can be reproduced. We do not want to focus on individual processing steps, but rather offer a standardization of the entire process after data collection up to the extraction of the final data for analysis. Except for the screening of complete data sets, no intervention based on subjective non-documented or non-replicable decisions of individuals in the artifact cleaning will be performed, but only replicable and standardized criteria may be chosen. There are very early findings suggesting that algorithmic approaches exceed individual valuation standards, so that actuarial approaches, once validated, should be preferred over subjective judgements (e.g., Dawes, Faust, & Meehl, 1989). In a meta-analysis, Grove, Zald, Lebow, Snitz, and Nelson (2000) were even able to show that the mechanical prediction or more accurately statistically defined prediction criteria performed significantly better than the clinical prediction in 33% to 47% of the studies examined, while the clinical prediction was more accurate in only 6% to 16% of the studies examined. Additionally, subjective standards vary inter-individually as well as intra-individually, while an algorithm has a replicable performance. Hence, we provide a (pre)-processing pipeline that is based on mechanical and reproducible criteria in order to avoid subjective variability.

We provide scripts for data export for statistical analysis in other software as well as for the visualization (ERPs, time-frequency plots, topographical maps) of electrophysiological data, to control for plausibility of the standardized solutions in EEG-analyzes. In the following, we will first present the proposed standard preprocessing pipeline for EEGLab (Delorme & Makeig, 2004) that was provided up to 2019 on https://sccn.ucsd.edu/wiki/Chapter_01:_Rejecting_Artifacts and then present and justify our changes and extensions. This pipeline is not to be seen as the new method for all applications that can simply be thrown at any type of data, but it is a suggestion for a standardized analysis, that may be used in several cases of data-(pre)-processing. Some suggestions seem to be

debatable at first glance (e.g., filtering after a first “segmentation” and not before, maybe causing edge artifacts or filtering with 1 Hz if interested in low frequency bands). However, if you read into the suggestions in detail, one might discover that some assumed problems are not given if followed the suggestions in principle (e.g., taking long first data segments in order to avoid filtering the entire dataset or extracting unfiltered data ICs in case of interest in low frequency bands).

**Preprocessing according to EEGLab**

A summary of the previously proposed preprocessing steps is:

Rejection based on independent data components:

1) Visually reject unsuitable (e.g. paroxysmal) portions of the continuous data.

2) Separate the data into suitable short data epochs.

3) Perform ICA on these epochs to derive their independent components.

4) Perform semi-automated and visual-inspection based rejection of data epochs on the derived components.

5) Visually inspect and select data epochs for rejection.

6) Reject the selected data epochs.

7) Perform ICA a second time on the pruned collection of short data epochs

8) Inspect and reject the components. Note that components should NOT be rejected before the second ICA, but after.

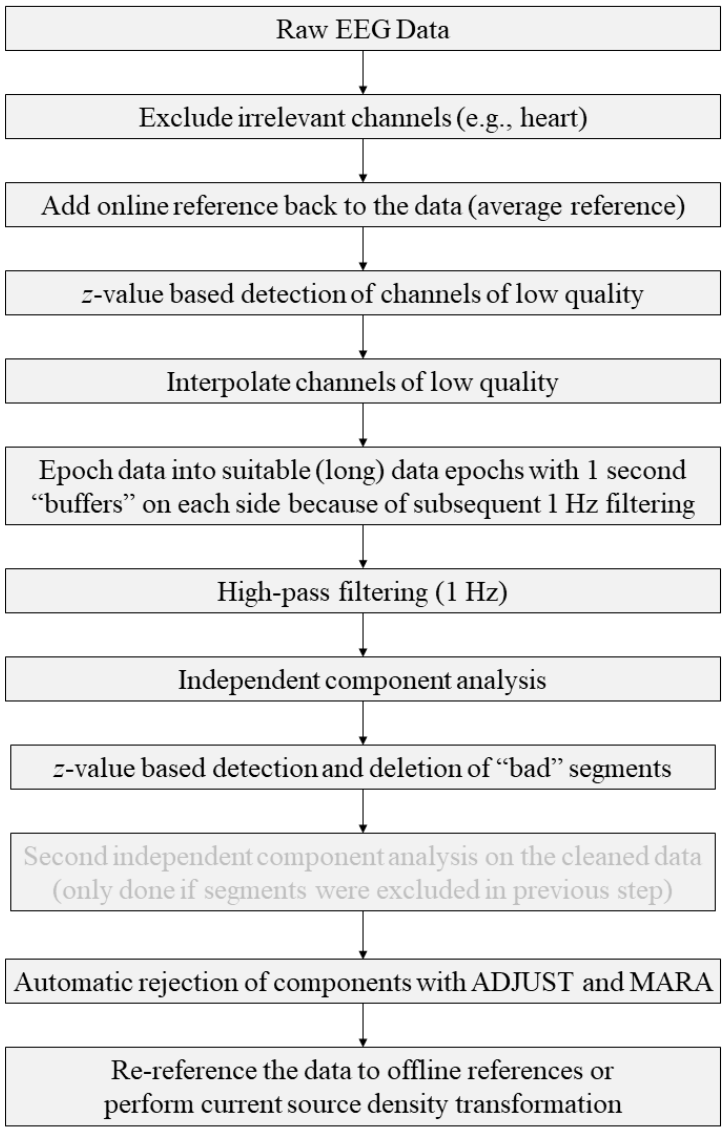


*Figure 1: Schematic representation of the preprocessing steps as recommended by the EPOS pipeline.*

## Preprocessing according to EPOS

As mentioned above, we tried to replace subjective un-replicable influences with standardized approaches.

For this purpose, the preprocessing pipeline proposed here will need the following software toolboxes: EEGLAB (Delorme & Makeig, 2004), ADJUST (Mognon, Jovicich, Bruzzone, &

Buiatti, 2011), MARA (Winkler et al., 2011), SASICA (Chaumon, Bishop, & Busch, 2015) and the CSD Toolbox (Kayser, 2009; Kayser & Tenke, 2006a; Kayser & Tenke, 2006b) or the CSD transformation provided by Cohen (2014). All these packages run on MATLAB (MATLAB, 2011), but some attempts are done to convert these packages to Octave (Eaton, 2002), an open source version of MATLAB.

First, we would like to point out that a good and standardized preprocessing can be worth a lot, but clean EEG data recording is essential ("garbage in → garbage out"). Therefore, the first step that is never mentioned but which is absolutely essential is to take your time with the data acquisition and apply the EEG caps/electrodes responsibly and with care.

The new (pre-)processing "chain" that is proposed based on the previous mentioned chain, as illustrated in Table 1:

1. Statistically detect and interpolate channels of low quality.
2. Separate the data into suitable data epochs.
3. High-pass filtering.
4. First independent component analysis.
5. Detection and deletion of bad segments based on *z*-value detection on ICs
6. Second independent component analysis.
7. Automatic inspection and rejection of the components using ADJUST and MARA with SASICA.
8. Re-reference (to current source density (CSD)).

**Table 1.** Comparison of the individual steps in the preprocessing pipelines according to EEGLab and EPOS.

| EEGLab | EPOS |
|---|---|
| Visually reject unsuitable portions of the continuous data. | Statistically detect and interpolate channels of low quality. |
| Separate the data into suitable short data epochs. | Separate the data into suitable data epochs. |
| Perform ICA on these epochs | High-pass filtering |
| Semi-automated and visual inspection-based rejection of data epochs on the derived components. | Perform first ICA |
| Visually inspect and select data epochs for rejection. | Detection and deletion of bad segments based on z-value detection on ICs |
| Reject the selected data epochs. | Second ICA |
| Second ICA | Automatic inspection and rejection of the components using ADJUST and MARA with SASICA |
| Inspect and reject the components. | Re-reference (to CSD) |

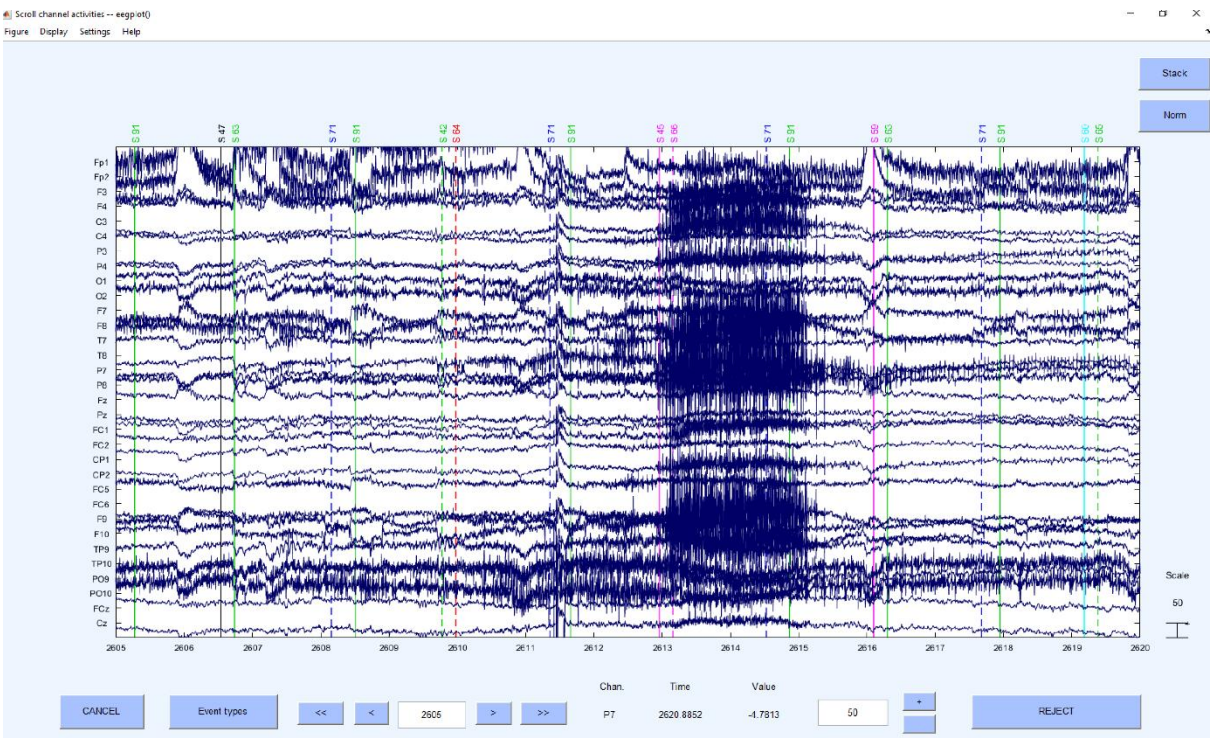


*Figure 2: Raw data example (not too good data).*

**Step 1: Statistically detect and interpolate channels of low quality**

This step is based on the raw data (see example in Figure 2) and detects and excludes channels with a very low signal to noise ratio. These channels will be interpolated and therefore will not contribute with their signal to the signal that will be processed further. Before a detection of the "bad" channels can be done, only the relevant channels have to be selected, Channels that are to be ignored for the following processing steps are for example, heart electrodes or skin conductance measurements, that may be (in-)directly related to brain waves but not of the same structure as the EEG signal. After selecting the EEG electrodes, the online reference should be added back to the data so that this electrode can also be used for further analyses or interpolated in the case of too much noise in this channel. As a reference system is required while also retaining the online reference as a channel, we use average reference to further process the data. The average reverence is very useful, if one has a sufficient amount of electrodes that cover the scalp fields sufficiently(Junghöfer, Elbert, Tucker, & Braun, 1999). If this is not given, one might introduce a bias based on the electrode distribution to the data. If other electrodes are being used as offline reference, one loses the electrode for interpretation in the data. Therefore, we would not recommend this approach, but if not possible otherwise, also other reference electrodes (for example linked mastoids) can be used right away here, losing the respective electrodes. In later steps we also may change from the present reference to the CSD reference, but we may not use this reference for automatic IC detections based on MARA and ADJUST, as they are not trained with these spatially filtered parameters and therefore come to very wrong conclusions. Hence, average reference provides us with the opportunity to assess the quality of the online reference channel along with artifacts only based on this channel, although the average reference may not be a generally preferable reference and can be altered at later stages. Hence as the first preprocessing step the "bad" channels are detected and interpolated using statistical criteria. We use a detection based on *z*-values. The probability, kurtosis and spectrum are detected according to the outlier criterion $z > 3.29$ (Tabachnick & Fidell, 2007, p. 73). For

the spectrum we use as frequency range 1 to 125 as suggested by Makeig and Delorme (https://sccn.ucsd.edu/wiki/Chapter_01:_Rejecting_Artifacts). The interpolation of the bad channels is done instead of a mere exclusion because of the irregularities of the matrices that would be introduced into the data structure and that would interact with later pre-processing and processing steps. Of course, the information of the interpolated channel is lost, but the structure of the data can be retained (see example in Figure 3).

The EPOS uses the EEGLAB *pop_select* function to select the channels, which are to be ignored for the preprocessing. The function *pop_chanedit* makes space for the online reference, which will be added back to the data. Using *pop_reref* the data is re-referenced to the average of all electrodes. The included channels for re-referencing depend on the montage and have to be adjusted in our script. Finally, the EEGLAB function *pop_rejchan* performs the detection of distorted channels according to the statistical criteria and *pop_interp* interpolates the resulting channels of poor data quality.

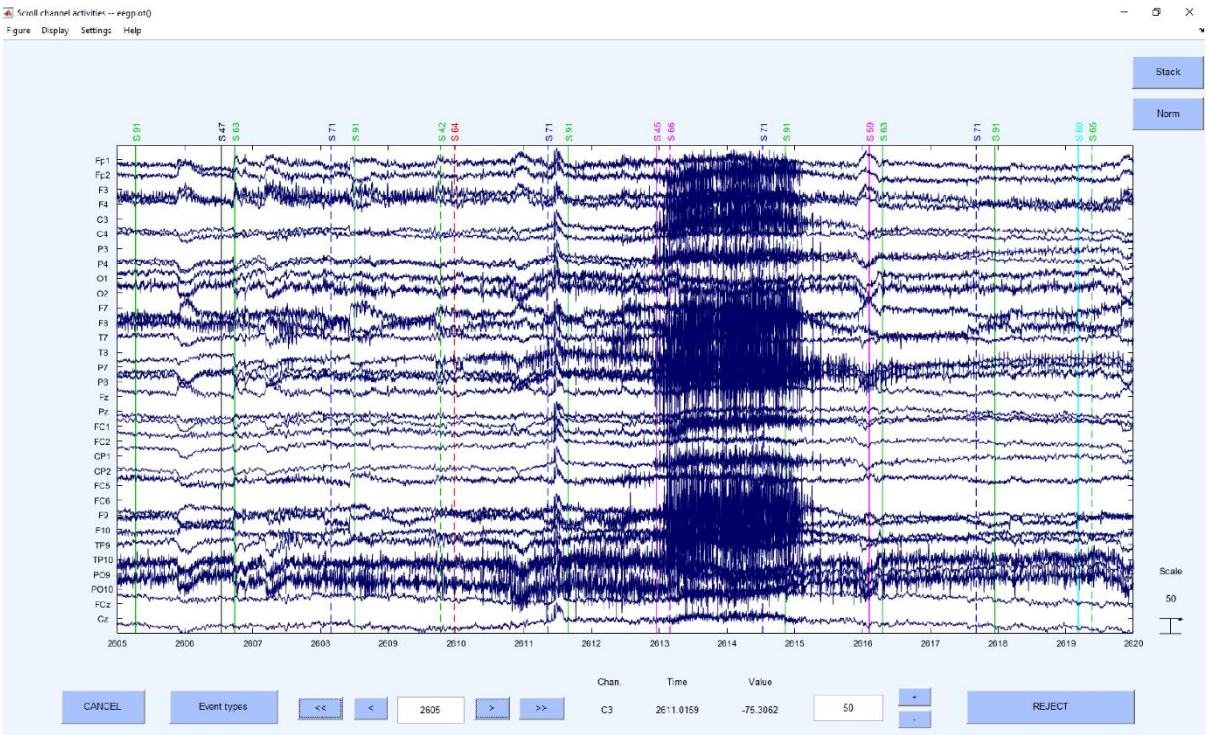


*Figure 3: Interpolated data channels.*

**Step 2: Separate the data into suitable data epochs**

The next step is to slice the data into suitable "first" data epochs which will later be segmented into the “real” segments but should be rather long as their purpose is to be the database for the ICA. Hence, a long segment might be beneficial. The length of a segment should also be chosen depending on the homogeneity of a trial. It is important to choose the data epochs as long as possible, since the following ICA provides a better solution for longer data periods. At the same time, the segments should be as short as possible, since the ICA solution leads to noisier and more unspecific ICs for different tasks, leading to a less sensitive *z*-value based artifact detection. In summary, the signal needs to be long enough to obtain a reliable measure and short enough to account for the rather non-stationary nature of EEG signals (Korats, Le Cam, Ranta, & Hamid, 2012). Therefore, we recommend either to segment the whole trial (if the trial is long enough and not too many different phases are present, or the trials are short and homogeneous) or to segment parts of an experiment. The segments in this phase may have a length of 8-20 (e.g., Möcks & Gasser, 1984) seconds, depending on the data quality and the task. Please keep in mind that a time window for the baseline correction is also necessary and should be included (x seconds before the marker/event of interest). For frequency analysis, more space is needed on both sides of a segment, since “edge effects” (i.e., distortions or transient effects resulting from a time-window larger than the time window for which data are available) can occur (e.g., Debener et al., 2005; Herrmann, Grigutsch, & Busch, 2005; Roach & Mathalon, 2008). The same applies to filters, which can also lead to edge effects. In order to avoid these artifacts later, we would recommend adding 1 second of “buffer time” on the segments on each side, as we will apply a 1 Hz filter in the next step. As the 1 Hz filter may produce “filter rippling” up to one second, this should avoid the artifact in the relevant data parts. In homogeneous data, overlapping segments can easily be used, although one has to take into account that this might also alter the data. An example for the segmented data can be seen in Figure 4.

We use the EEGLAB function *pop_epoch* to slice the data into segments of a suitable length, depending on the experiment and its trial duration.

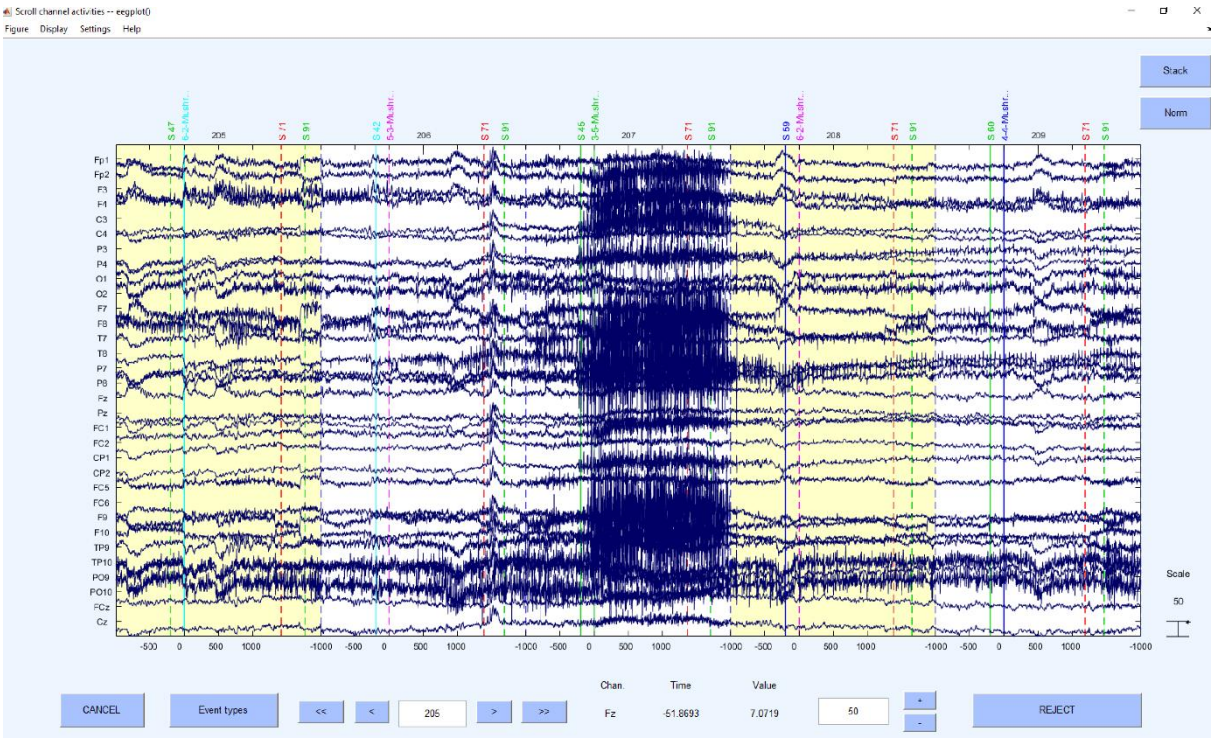


*Figure 4: First segmentation. The marked segments are visual aids to evaluate later steps.*

**Step 3: High-pass filtering**

As a third step we apply a 1 Hz high pass filter to the data. This is done to get a more stable ICA solution as no low frequency shift is present (Winkler, Debener, Müller, & Tangermann, 2015). Furthermore, after extensive testing of the influence of different filters on the performance of MARA (Winkler et al., 2011), we found that MARA works best with only the 1 Hz filter. A 2 Hz filter, for example, will not correct side eye movements as good as if it was filtered with 1 Hz and data filtered to the power spectrum between 2 and 39 Hz (as recommended in the MARA manual) will not correct muscular activation as well as it is done without the 39 Hz filter. At this point we would also like to remind that every filter changes the data, although the filtering at this point is only used to get a better basis for the ICA and artifact rejection based on the automatic artifact IC detection of MARA (Winkler et al., 2011) and ADJUST (Mognon et al., 2011). As previously stated, there might occur edge artifacts (filter

rippling). These artifacts are very prone to occur in short data epochs, as they are on the edges of the filtered data. Because of this problem, one normally recommends filtering unsegmented continuous data, instead of "segmented" data. However, as the "segmentation" we performed in step 2 is basically a selection of a large continuous data part with sufficient edges for the occurring filtering artifacts, instead of only the shortest data part of interest, one may start the filtering at this step, getting a quicker and more efficient filtering process, because only parts of the data need to be filtered and not the entire dataset. Also as mentioned above we recommended 1 second data "buffers" on the large data "segments" in order to avoid filter rippling in the relevant parts of the data with a 1 Hz high-pass filter. Additionally, one has to keep also in mind, that filters only attenuate the frequency bands they are designed to work on and are not built for a complete dampening of the respective frequency responses. This may result in residual artifacts of very large frequency artifacts, if the dampening curve is applied with an inappropriate filter order (i.e., a very large muscular artifact will have effects even after the filtering). To reiterate the different filter types that may be applied: Notch (deletes the target frequency, e.g., 50 Hz (AC in Europe)), low-pass/high-cut (all frequencies below the target frequency will be attenuated), high-pass/low-cut (all frequencies above the target frequency will be attenuated) and bandpass (combines low-pass plus high-pass). It is important to note, that some research interests are in the frequency band below 1 Hz and therefore usually apply much lower high-pass filters (for example .001 Hz if interested in slow waves). However, as there is the opportunity to write the IC solutions back to the unfiltered data, one may use the 1 Hz filtering at this point still, in order to get a good performance of the MARA algorithm.

In the EPOS pipeline, we apply the EEGLAB function *pop_eegfiltnew*, for which the toolbox firfilt (Andreas Widmann) is required. Depending on the data segments, the filter order for the 1 Hz low-pass filter must be adjusted in the script. An example for filtered data can be seen in Figure 5.

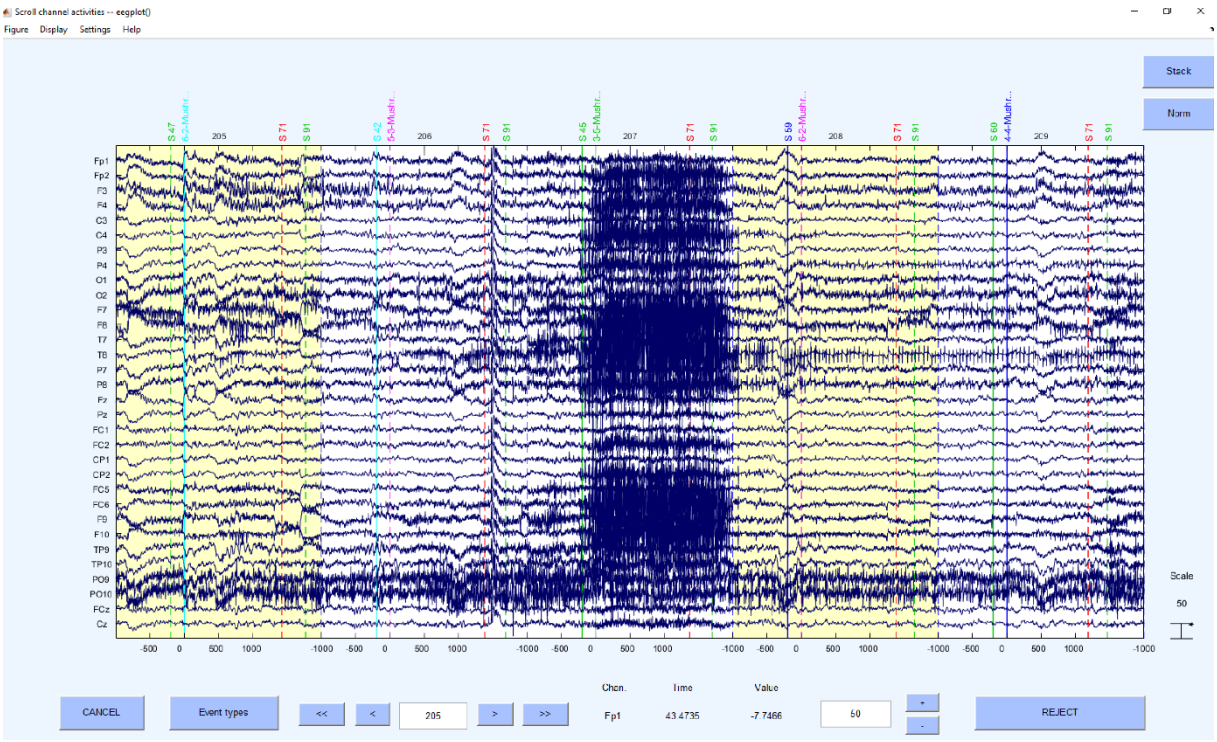


*Figure 5: Filtered data.*

**Step 4: First independent component analysis**

In the next step we apply an independent component analysis to the data. The ICA is a statistical linear decomposition of the signal into independent components, each of which contributes as much specific information as possible to the data (Makeig, Debener, Onton, & Delorme, 2004). Thus, each electrode provides data that is assigned to a source/sensor. The ICA decomposes the linearly mixed sources at the sensor level into independent components (Bell & Sejnowski, 1995) and we get as many components as there are sources (i.e., 64 independent components with 64 electrodes) . As a result, ICA separates the actual electronic brain signal from non-brain artifacts such as eye movements or muscle activity. The ICA is a so-called "blind" separation technique and therefore does not guarantee meaningful results (e.g., Jung et al., 2000). Not every extracted component is equally plausible and depends strongly on the data quality and the specific ICA algorithm used. Depending on the interpolation, however, less information is obtained for each interpolated channel. ICs can be independent as well as dependent on each other, since components included in a source vector can be either correlated or uncorrelated

(Kim, Eltoft, & Lee, 2006). The resulting ICs also have a time course and a frequency distribution as channels have. But as the topographical order of the channels is dissolved a new topographical projection is provided for each IC. Based on these features, artifact detection can be performed, either only using parts of this information (time course, topography, frequency response) or all together. In step 5, we will only use parts of the information neglecting the topography of the ICs to select artifact segments, while in step 7, the automatized machine learning and criteria-based algorithms are using all information in order to select artifact ICs.

To perform the ICA, we use the command *pop_runica* in EEGLAB (Delorme & Makeig, 2004; Makeig et al., 2004).

**Step 5: Detection and deletion of bad segments based on *z*-value detection on ICs**

Now the bad segments are selected and deleted based on a *z*-value detection on the ICs. As in the first step, the criterion of $z > 3.29$ for the probability and kurtosis is applied on the channel basis. The reason for this step is to increase the data quality to be able to clean up artifacts even better with the following second ICA. On global level we used a very high *z*-value threshold of $z = 20$ to only correct for very huge artifacts and to prevent the overcorrection of different signal components. This approach was recommended by Delorme & Makeig, (Delorme & Makeig, 2004) that was provided up to 2019 on https://sccn.ucsd.edu/wiki/Chapter_01:_Rejecting_Artifacts because of the higher sensitivity to "extraordinary" not regularly appearing artifacts (e.g., singular hiccup) than a merely channel *z*-value based artifact detection.

The EEGLAB command *pop_jointprob* was used to reject the probability and *pop_rejkurt* for the kurtosis. The selected segments were removed using *pop_rejepoch*. An example for a rejection of segments can be seen in Figure 6.

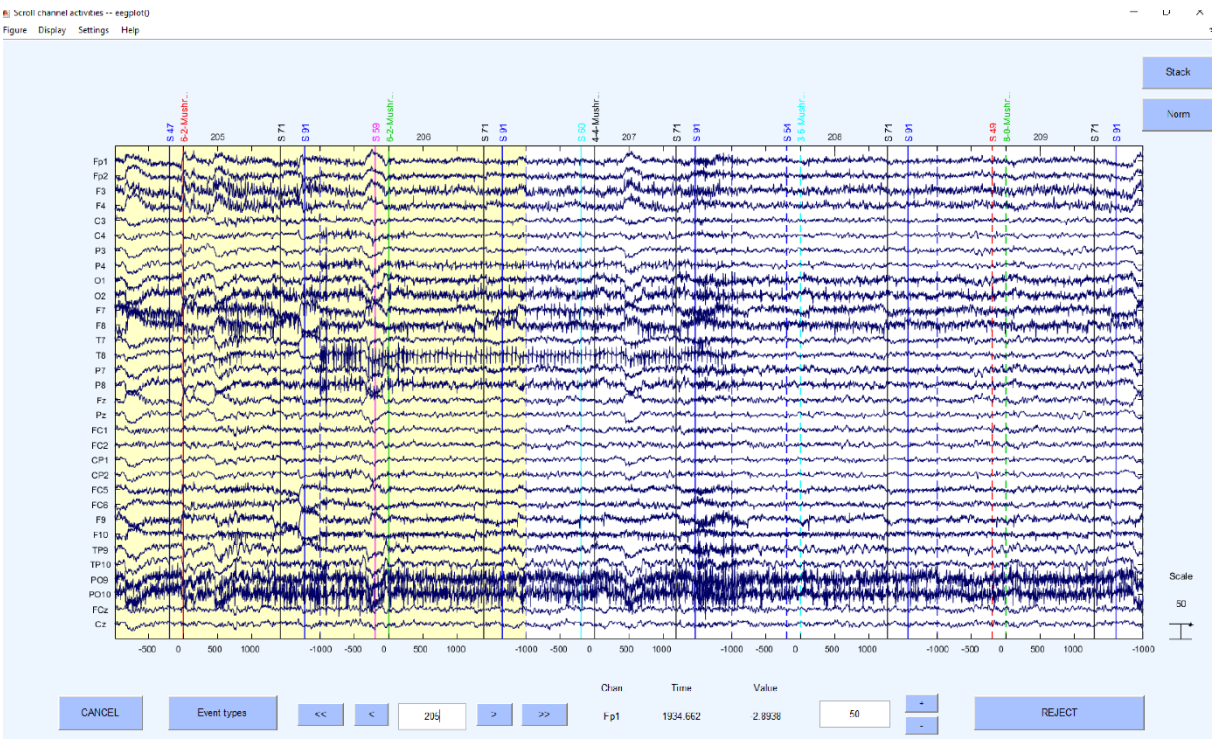


*Figure 6: Bad segments excluded.*

**Step 6: Second independent component analysis**

This second ICA is now performed on the data cleaned for poor segments. This step is only performed if at least one bad segment was detected and rejected. The goal is to achieve a better signal-to-noise-ratio by identifying artifact driven components containing no relevant signal. These artifact components will be deleted to compute signal that only consist of non-artifact data or that is at least less artifact polluted. Therefore, once again a signal decomposition is performed leading to the previously mentioned information in the ICs. After the ICA we select the ICs that represent signal and those that represent noise. Again, we use the EEGLAB function *pop_runica.*

**Step 7: Automatic inspection and rejection of the components using ADJUST and MARA with SASICA**

In the seventh step, the resulting components are automatically inspected and rejected using ADJUST (Automatic EEG artifact Detection based on the Joint Use of Spatial and Temporal features; Mognon et al., 2011) and MARA (Winkler et al., 2011) with SASICA (Chaumon et

al., 2015). SASICA serves as EEGLAB plugin, which contains various artifact correction algorithms from different researchers (e.g., Fully Automated Statistical Thresholding for EEG artifact Rejection (FASTER), Nolan, Whelan, & Reilly, 2010, ADJUST, and MARA).

ADJUST uses algorithms based on temporal and spatial filters to identify mainly (but not exclusively) artifacts caused by eye movements. These include blinks, horizontal and vertical eye movements, but also generic discontinuities. The algorithm uses an expectation-maximization-based approach to automatically detect the threshold of spatio-temporal properties of different artifact types and classify them accordingly (Wu et al., 2018). MARA combines different measures to automatically classify ICs as artifacts via a linear machine learning algorithm. In summary, two spatial, one temporal and three spectral features provide the measures for the best classification results. These different classification components are described in detail in the original paper. MARA is not designed to detect a specific artifact type, but rather is variable to detect eye artifacts, muscle artifacts, the heartbeat, or loose electrodes. Regarding MARA, we have already set the reference to average and filtered the data with 1 Hz. MARA has shown to perform well in the automatic classification of artifacts (85 - 91% accuracy compared to experienced raters, Winkler et al., 2014; Winkler et al., 2011). Although trained and experienced EEG researchers may have an even better ability to distinguish signal from artifact components, automatic artifact correction algorithms have an increased reliability of artifact removal exceeding the human raters.

After we have configured the specific options (i.e., ADJUST or MARA marked it bad) in the script, we perform the automatic removal of the ICs with the EEGLAB command eeg_SASISCA. Using the marking of an artifact by ADJUST or MARA is a very conservative approach because it rejects as many ICs as possible. However, as ADJUST and MARA are selectively strong for certain artifacts (e.g. ADJUST is rather strong in detecting heart-beat related shifting artifacts that MARA tends to miss while MARA is more sensitive in detecting

very noisy components) the selected method leads to less artifact prone data, yet being a very strict approach concerning mixed components. We also provide code that writes the ICA solution back to the unfiltered EEG data. Therefore, we project the ICA solution that is based on the automatic selection by MARA onto the original data without including the necessary preprocessing steps to achieve them. This leads to dropping the preprocessing artifacts that were introduced to gain an optimal performance of the standardized preprocessing. Hence we gain a projected solution of the ICA, which might not be the identical solution that we would have gotten with an ICA on the raw data, but that is a projection of the optimally prepared data matrix for automatic IC selection on the raw data matrix. Please keep in mind, that we only recommend these packages because they are openly available and provide replicable results. The respective setting of these software packages might not fit your need in detail and we highly encourage you to still use other replicable solutions that may select your ICs for you (for instance also a trained neural network for IC detection, mirroring expert opinions for specific tasks, (Rodrigues, Ziebell, Müller, & Hewig, 2020b)), as long as you provide these solutions and the important needed information to understand the procedure to other researchers and therefore guarantee the replicability of your analyses. Examples for the IC cleaned data can be seen in Figure 7 and Figure 8, for the projected unfiltered data and the filtered data solution.

In some (pre-)processing routines (mostly ERP routines), at this point or at the processing step 2 below, there is an additional segmentation in very small data segments and an additional artifact selection is performed after the IC cleaning procedure. We chose to not include it here in this script because of the previous step 5, but if needed in your data, please inform the reader what segmentation steps have been taken and which statistical selection criteria has been used. Please avoid selection “by hand”.

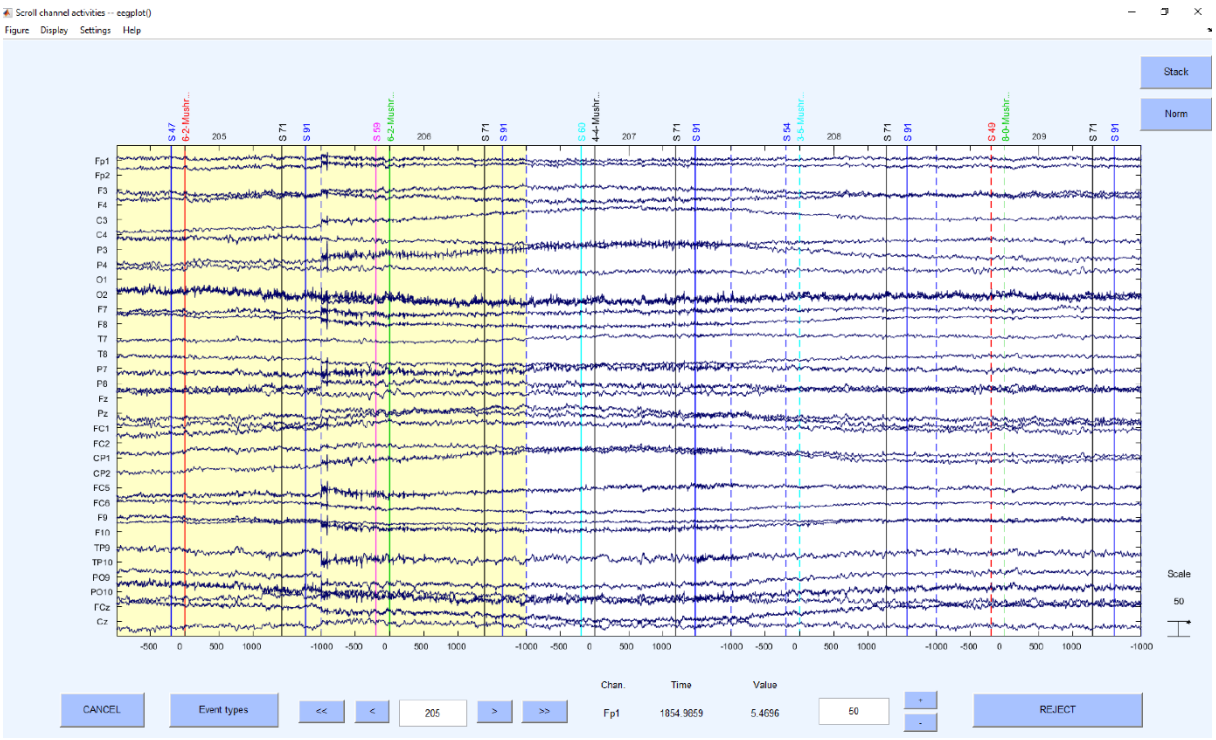


*Figure 7: Data after second ICA and IC cleaning, reprojected on the original, unfiltered data.*

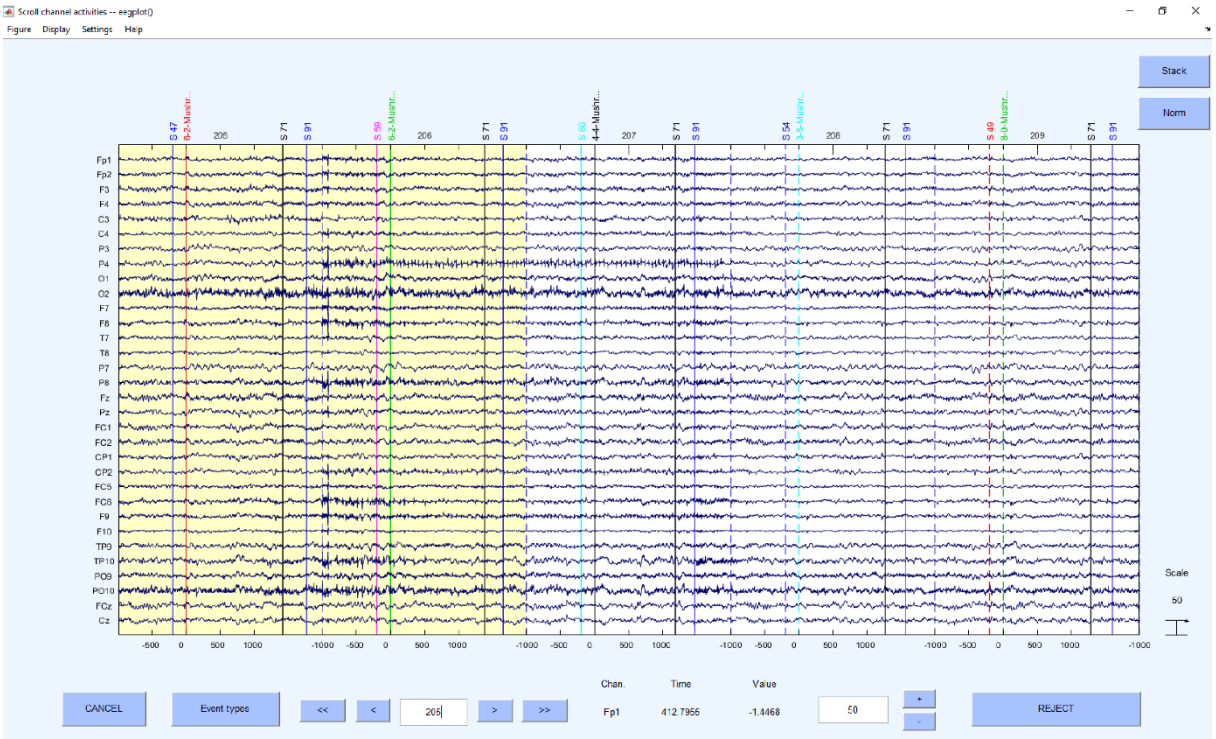


*Figure 8: Data after second ICA and IC cleaning, filtered data solution.*

## Step 8: Transform via Re-reference or Current Source Density (CSD)

Finally, we re-reference our data in the last step of the preprocessing. The selection of reference or montage is paramount to being able to visualize effects of interest, and the choice may be

determined, in part, based on the standard of practice in a given research domain and the specific question of interest. To the extent that spatially-specific effects are important, the current source density (CSD) transformation is preferred and suggested by us (for a comparison of different reference schemes concerning alpha-band activity and frontal-asymmetry, see Hagemann (2004). Nevertheless, one may also choose another reference like linked mastoids or other reference schemes that are suitable and common in the respective field of EEG research. CSD provides an estimation of relative current at a point on the scalp surface as a function of the surrounding points. The distances are weighted with the relative activity on the electrode: The surface is estimated as a sphere, the signal differences to the adjacent are measured and the weighting of each difference is performed for each distance. Thus, a reference without reference is obtained and any electrode can be used. This results in a spatial filter that sharpens the topography of the (in)activation. There are two readily-available possibilities for doing this, either the CSD toolbox from Kayser (Kayser, 2009; Kayser & Tenke, 2006a; Kayser & Tenke, 2006b) or the *laplacian_perrinX* function provided by Cohen (2014), based on the approach of Perrin, Pernier, Bertrand, and Echallier (1989; 1990). We perform this step at this rather late point in the preprocessing chain, because an average reference is favorable for ICA and component classification using MARA and ADJUST (steps 4 and 5), since all the algorithms of MARA and ADJUST have not been trained with CSD transformed data and their selection accuracy concerning artifacts is not good if CSD data is used. Also, with average reference, ICA scalp topographies have a zero-total potential (i.e., red and blue are balanced). If no average reference is chosen, some IC scalp topographies could be completely red or blue, leading to visually odd topographies that are hard to interpret.

In this step, either the code provided by Kayser (Kayser, 2009; Kayser & Tenke, 2006a; Kayser & Tenke, 2006b) or the *laplacian_perrinX* function from Cohen (2014) is used. These two tools provide comparable results, although the latter is substantially faster in execution; nonetheless,

we highly recommend visiting the website provided by Kayser (http://psychophysiology.cpmc.columbia.edu/software/CSDtoolbox/tutorial.html) in order to get more information about how CSD transformation is implemented and what it does Alternatively, any other reference can be used with the command *pop_reref* (e.g., linked mastoids). Examples of the unfiltered and filtered solutions for CSD and linked mastoid reference can be seen in Figure 9, Figure 10, Figure 11 and Figure 12.

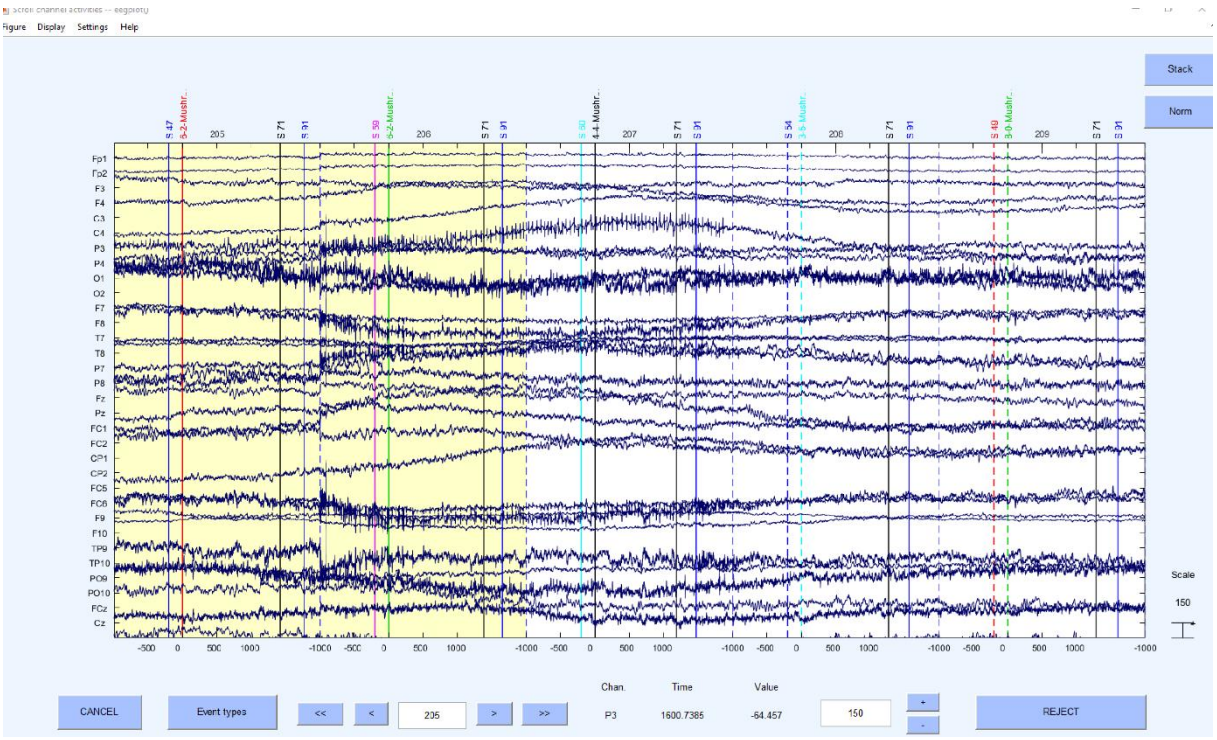


*Figure 9: Unfiltered CSD transformed data solution.*

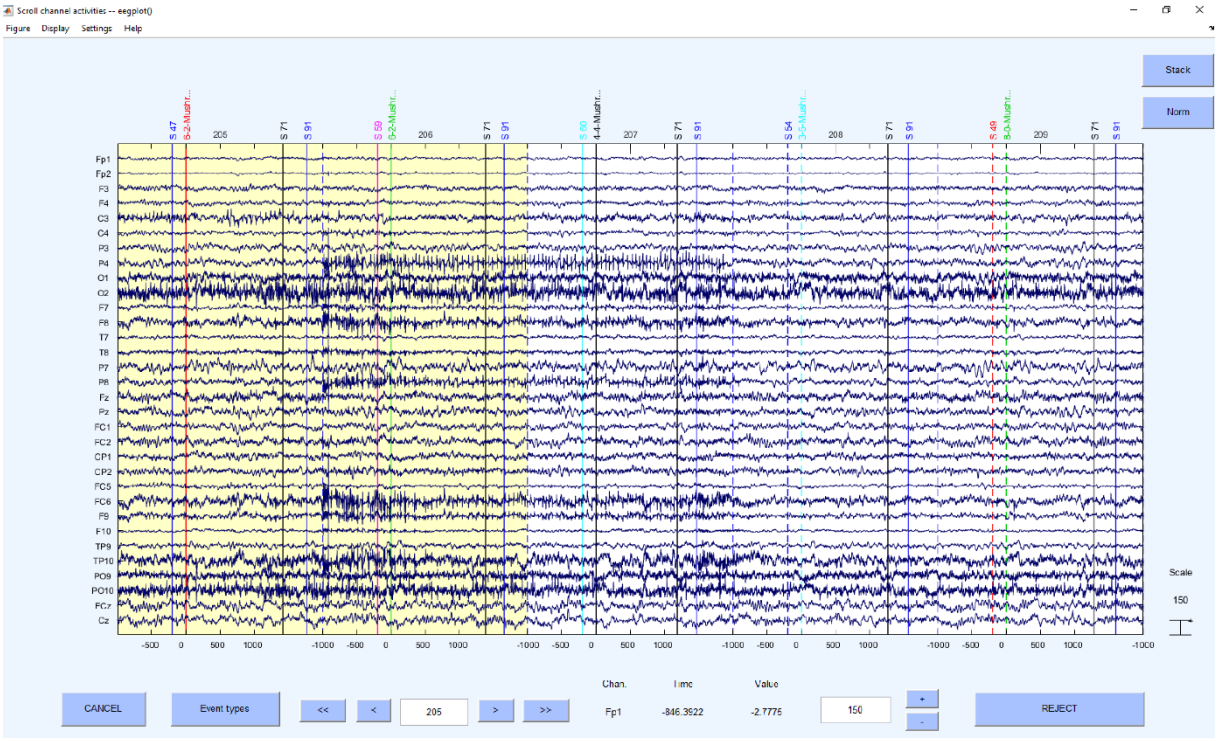


*Figure 10: Filtered CSD transformed data solution.*

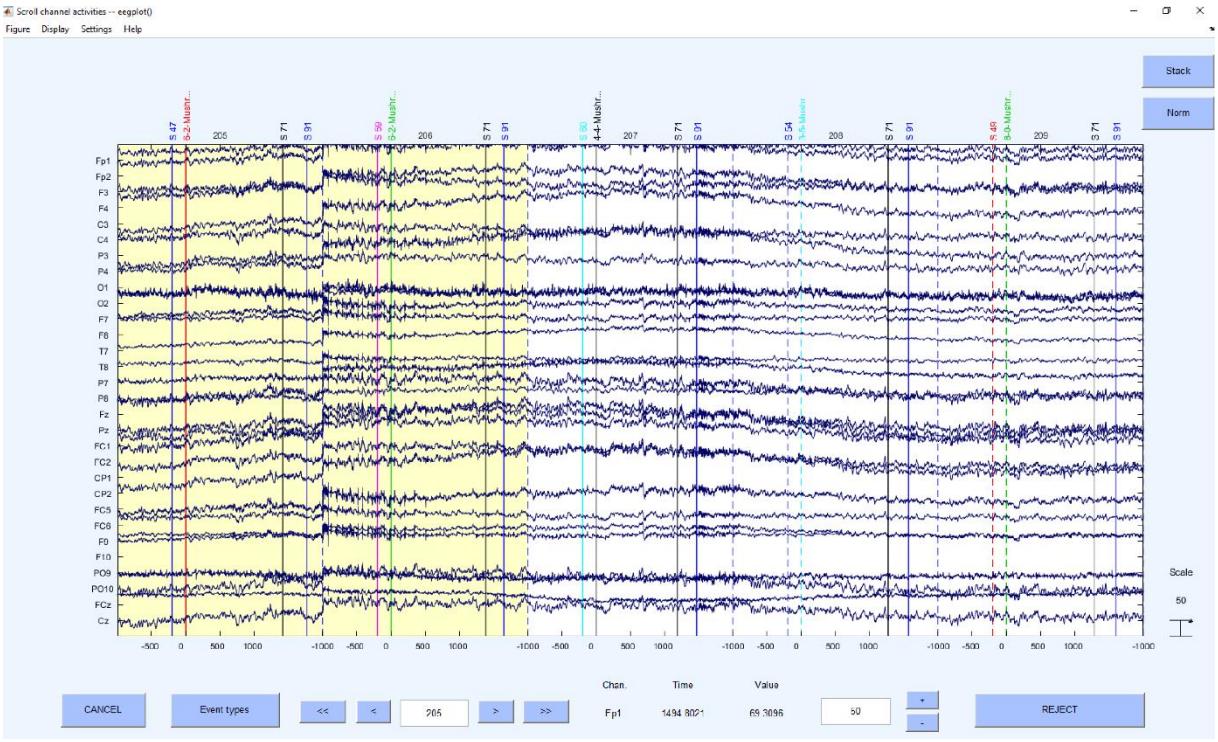


*Figure 11: Unfiltered linked mastoid referenced data solution.*

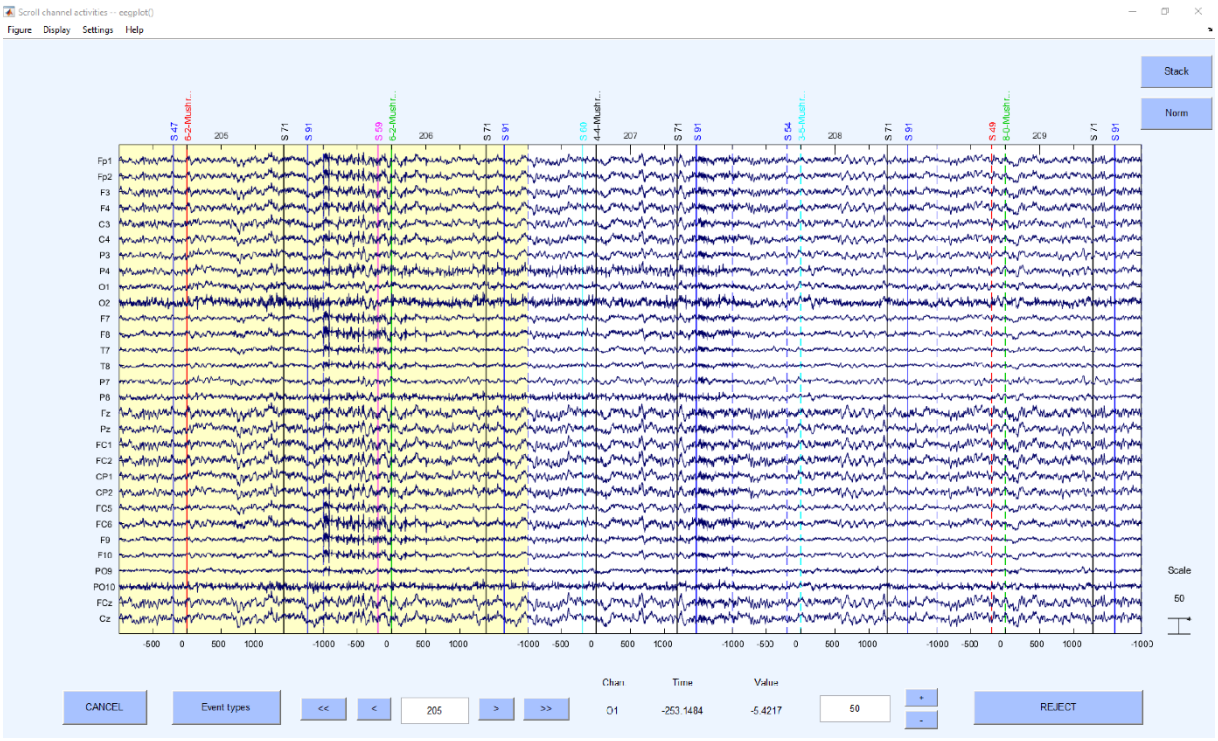


*Figure 12: Filtered linked mastoid referenced data solution.*

## Processing according to EPOS

After the pre-processing of the EEG data has been completed at this point in a standardized and automated procedure, we will describe the further processing of the data in the following steps (see Figure 2). For this purpose, nine steps are performed, some of which are optional, depending on the experiment, data set and personal preferences. The MATLAB addons *boundedline* (Kearney, 2020) and *export_fig* (Altman, 2020) are required for creating graphics and exporting data. Also, the wavelet function based on the code provided by Cohen (2014) and edited by John J.B. Allen and Johannes Rodrigues is required to analyze time frequency results. To perform single trial analyses for frequencies later, an adjustment was made to the frequency extraction functions of Cohen (2014). Otherwise these functions were implemented as described by Cohen (2014).

1. Segment the data for analysis
2. Drop the cases that are not present (for example in free choice paradigms)
3. Automatic peak detection in a given time-window in EEG signal

4. Compute and visualize event-related potentials
5. Topographical maps (Topoplots) in the time-domain
6. Automatic peak detection in a given time window in frequency responses
7. Topographical maps for frequency responses
8. Time-frequency plot for a specific electrode in a broad frequency window
9. Export the data to statistical software

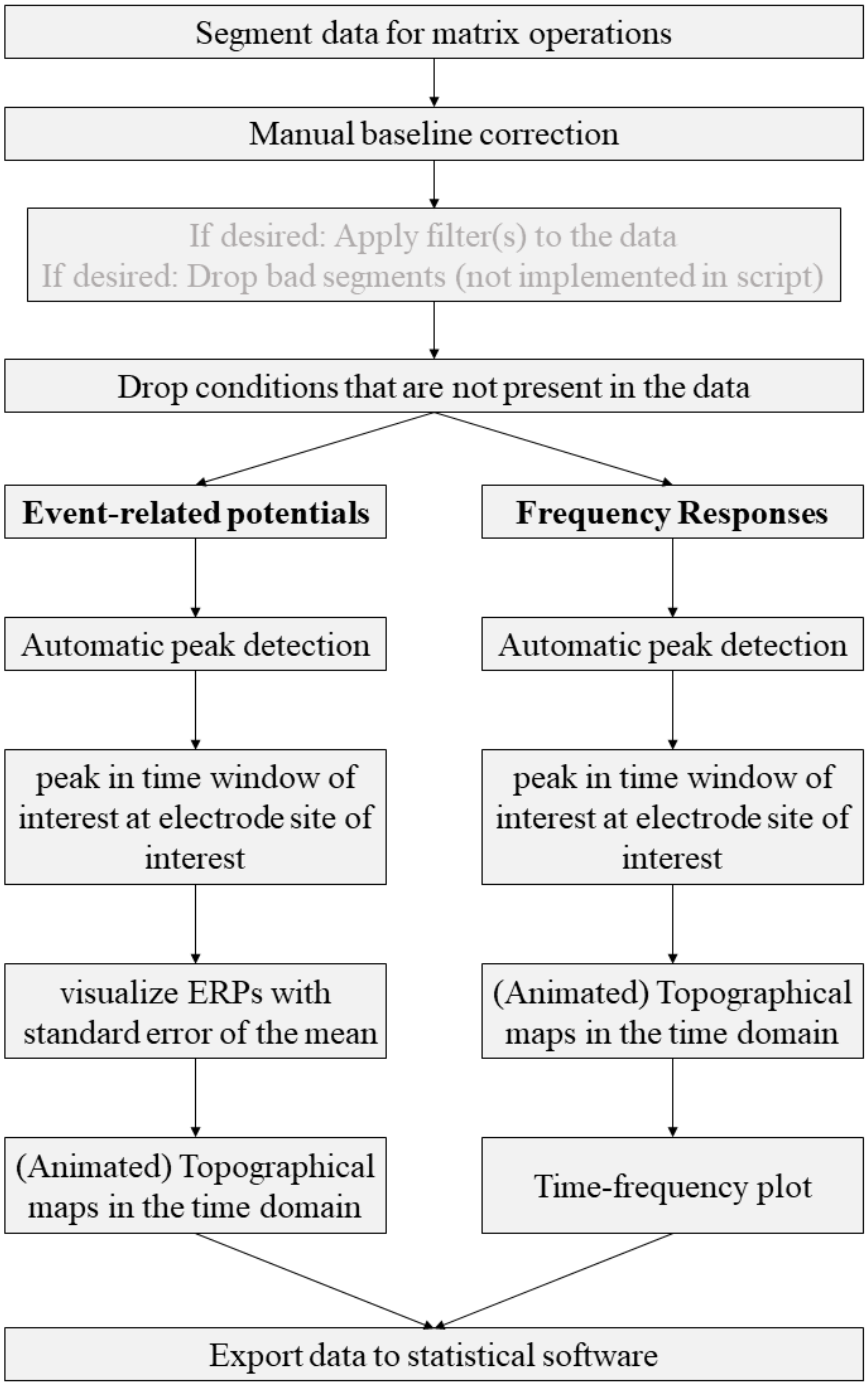


*Figure 13 Schematic representation of the processing steps as recommended by the EPOS pipeline.*

**Step 1: Segment the data for analysis**

The goal of this step is to create a 4-dimensional matrix for the signal and each frequency for data analysis and to generate a 5-dimensional matrix for single trial analysis (of course, other dimensions in addition to the following dimensions can be generated). Depending on the size of the matrix, however, generating this matrix can lead to memory problems. A solution for this is to create only one matrix at a time or to resample the data during pre-processing as a very first step. To avoid estimated values based on interpolation, we advise resampling only to a new sampling rate that is a divisor of the previous one. Also, it is necessary to use anti-aliasing filters prior to resampling to avoid the introduction of aliased frequencies. To perform the resampling, there is for example the functions *pop_resample.* Generally, we would recommend recording the data only with the required sampling rate for all the planned frequency analysis and filtering (normally 250 Hz is sufficient) instead of the highest available recording frequency, to avoid resampling. Higher sampling rates, in unique cases, may help to overcome very specific data corruption problems, but normally they just take recording resources as well as lead to down-sampling of the data later, which could be avoided if sampled in a lower frequency right away.

At this point the data can be segmented again according to the relevant markers for the respective task to allow smaller segments to be extracted from the existing epochs if desired. As the first “segmentation” in the preprocessing was made with the intend to capture segments that are very fitting for IC decomposition and artifact detection and therefore might be overly long for a frequency response of interest or an event related potential, a second segmentation can be performed, now with the goal of getting a fitting epoch for data analysis. These markers must be selected, for which a separate segmentation script is recommended. An example of such a segmentation script is also provided along with the (pre-)processing chain. In this script, the variable "casearray" contains all relevant condition triggers for this experiment, grouped by condition. As mentioned above, in some processing pipelines (mostly ERP related) there is a

second bad segment detection step at this point (see Step 5 pre-processing). Feel free to execute this step if needed, but we did not include it here or in the scripts. Please also mention this step if in detail, if you decide to include it here. As a next step, the baseline correction is calculated manually and not with the *pop_rmbase* function from EEGLAB. The reason for this is that we have encountered problems with the *round* function in different MATLAB versions, which does not (or did not) work properly with some EEGLAB versions, leading to wrong baseline applications. Therefore, we avoid such compatibility issues by manual calculation. Next, the frequencies of interest are defined. We assume that only specific time-frequency windows and ERP components will be analyzed in a hypothesis-driven fashion for the particular research question (of course, other frequency bands and ERP components can be considered for exploratory purposes). If desired, a filter with specific characteristics can be applied to the data depending on the ERP of interest. Finally, the user decides whether to look at single-trial data. We recommend using multilevel models for such a single-trial analysis. We also recommend using frequency bands instead of pure ERP-related data, since they have a higher reliability in single trial EEG analyses compared to similar single trial ERPs (e.g., see Rodrigues, Liesner, Reutter, Mussel, & Hewig, 2020a).

**Step 2: Drop conditions that are not present**

This step is short and simple. We recommend excluding those conditions (to reduce the amount of data) that are not contained in the data set but were present in the segmentation file. This is especially relevant for free choice paradigms, as some participants may have chosen not to act in a specific manner. Therefore, these cases can be dropped from the segmentation file for this person.

**Step 3: Automatic peak detection in a given time-window in EEG signal**

In this step, a peak is searched for in a time window of interest at an electrode position of interest via the averaged signal or the average over distinct conditions, leading to averages over trials

in the respective condition instead of a total average. Please note, that the peak is not taken as a single value, but a time window is defined around the peak in order to avoid biases due to peak latency or artifacts and therefore capitalization on noise (see Luck, 2005). You will later be able to visualize and export and analyze the average of the trials in a condition or even the single trial values of the data if you intend to perform single-trial analysis, if you also preprocessed the single trial matrices (see e.g., Rodrigues et al., 2020a for an example of both approaches). The corresponding parameters (search window, electrode) depend on the ERP of interest. For this purpose, search a priori in the literature for the recommendations but critically evaluate guidelines and research propositions (e.g., it should be questioned whether the FRN is considered only at Fz, as is often the case in the literature, although FCz is also available and the respective topography also indicates that the component is rather mid-fronto-central than only limited to frontal regions. Hence a "standard" might be questioned by an informed decision. This "standard" however was partly built in times where the electrode position FCz was not available due to few electrodes. In must current setups the electrode position FCz is often available and can also be used as in many studies the topography of the FRN and the midfrontal theta reaction has its´ peak on FCz).

**Step 4: Compute and visualize event-related potentials**

In this part of the script we offer the possibility to generate different forms of ERP graphs. First of all, an ERP can be plotted as is traditionally plotted in older manuscripts, which only consists of one waveform per condition (see Figure 14 upper panel). Next, we offer ERPs with shaded error lines (Kearney, 2020), which in addition to the course of the ERP also provide information about the precision of the estimate of the mean value (ERP). In this step, between errorlines (between standard error) are added to the figure (see Figure 14 middle panel). Alternatively, we provide code to add the mean within error lines (mean within standard error, being mean within standard errors of the differences of relevant conditions) to the figure (see Figure 14 middle

panel). For the latter, it is important that the researcher is aware that meaningful conditions should be taken, or meaningful clusters of conditions should be calculated (only a short example in the script, but simply use the “nanmean” command).

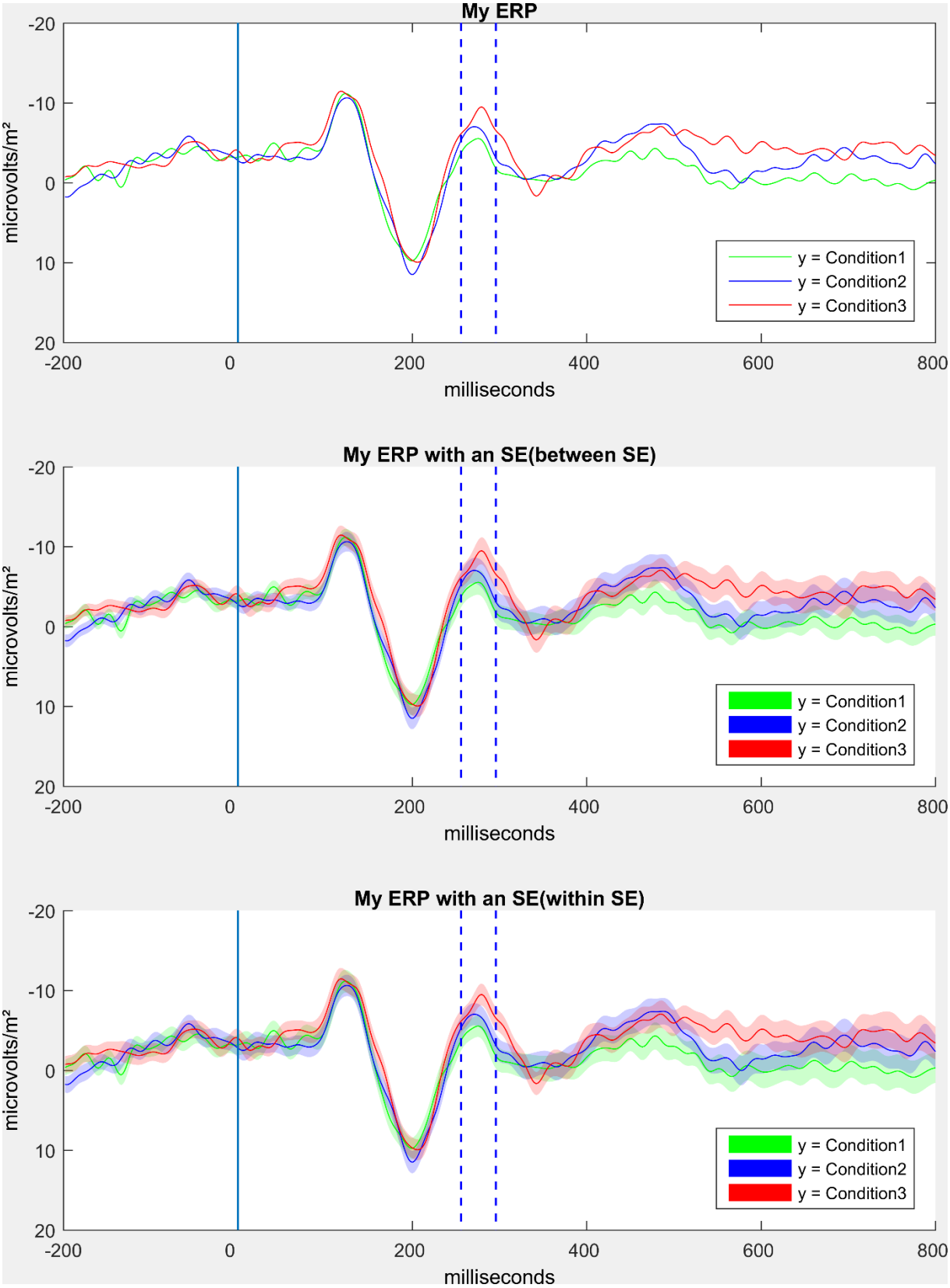

*Figure 14: Comparison of the three different ERP plot options.*

**Step 5: Topographical maps (Topoplots) in the time-domain**

In the next step we provide code to create topographic maps. We include the option to generate a topoplot for a time window of interest (peak-window) for ERP (see Figure 15), but also to create an animated graphics interchange format (GIF). This GIF depicts different time intervals to show the dynamic changes in the topography and to verify the selected time-window of interest as correct for the corresponding electrodes. Hence, you are able to validate your choice if time window as well as your choice of electrode of interest.

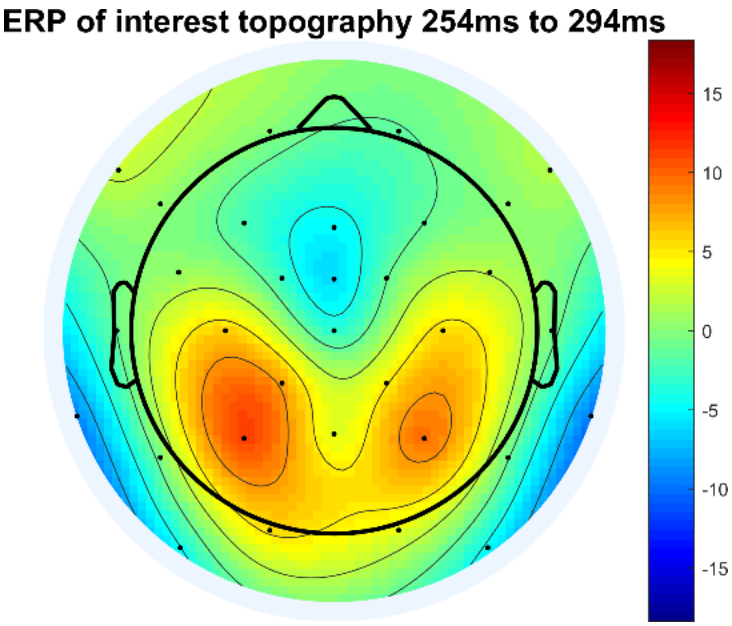


*Figure 15: Topographical plot of ERP.*

**Step 6: Automatic peak detection in a given time window in frequency band**

We are using morlet wavelets to perform time frequency decomposition. Our processing approach assumes the user has an a priori frequency band of interest and that the analysis focusses on said frequency band to avoid capitalization on chance findings.

Similar to the peak-detection in ERPs, the peak is searched for in the time-window of interest at the electrode of interest for the frequency band of interest. The corresponding parameters can

be set in the same way as those used for the ERPs. Again, check the evidence in the literature and remain critical (e.g., it should be questioned whether the peak of midfrontal theta is considered only at Fz, as is often the case in the literature, although FCz is also available and the respective topography also indicates that the component is rather a mid-fronto-central frequency response). Depending on the task, it might be useful to look at the average of all conditions (as pointed out by Cohen, 2014) or at the peak in certain conditions that differ from others, which, however, biases the chance to find significance. The attached script contains examples, but they must be adapted for different experiments.

**Step 7: Topographical maps for frequency responses**

This step is identical to step 5, both the topographic maps for the peak-window of the frequency response and a GIF for the time course are implemented in the script. Here you may validate your choice of electrode of interest and time window (see Figure 16).

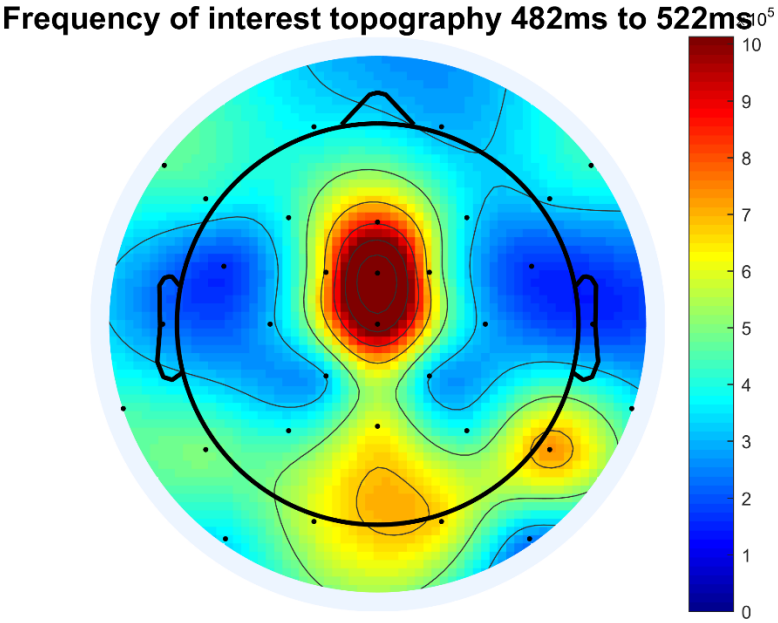


*Figure 16: Topographical plot of frequency of interest, here example theta frequency and display of raw power.*

**Step 8: Time-frequency plot for a specific electrode in a broad frequency window**

In the last step of the graphical illustration of electrophysiological data, we implemented code for the creation of time-frequency plots. For this time-frequency plot we use the plot function

based on the code provided by Cohen (2014) and edited by John J.B. Allen and Johannes Rodrigues (see Figure 17). It provides either a log transformed power output, a raw data output or the recommended dB change to baseline output that corrects for the power law that affects the display of different frequency bands together. The result is a time-frequency plot, which not only shows the frequency response limited to the desired functional frequency but can also display larger frequency windows in order to not only validate your selection of the electrode of interest and time window, but also your frequency of interest. As a recommendation, we suggest the spectral range of 1-30 Hz if you are not particularly interested in gamma frequencies. We also recommend using the dB to baseline change setting as mentioned above.

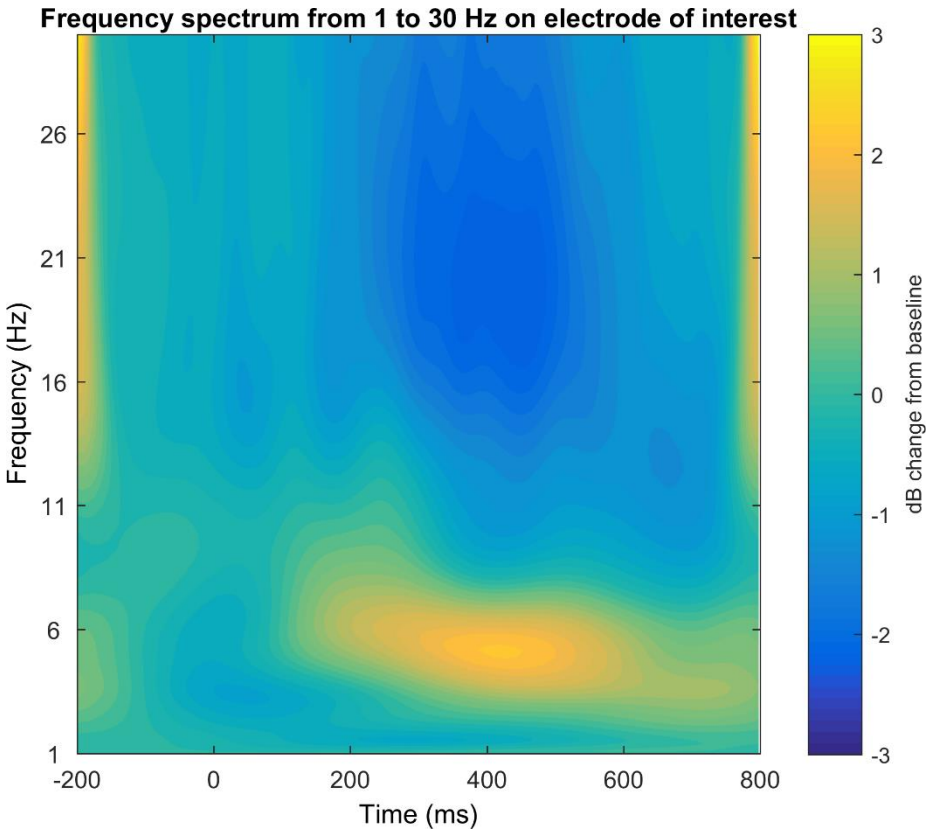


*Figure 17: Time-frequency plot on the electrode of interest in the frequency spectrum 1-30 Hz.*

But concerning the gamma frequency band, we are rather cautious in interpretation and try to avoid it as there is evidence that microsaccades (e.g., Dimigen, Valsecchi, Sommer, & Kliegl,

2009; but also Hipp & Siegel, 2013) and electrical muscular activity (e.g., Whitham et al., 2007) may drive these frequency spectrum responses.

**Step 9: Export the data to statistical software**

In the last step of the processing "chain", we offer code for exporting the EEG data into different statistical programs using the Excel xlsx or the MATLAB mat data formats. We support the export into the long format (i.e., each row represents a data-point in a specific condition combination per participant, with columns indicating the data as well as the conditions and the participants, resulting in multiple rows per participant) for the mean signals/frequencies. This format is for example required by many R (R Core Team, 2020) packages to calculate analysis of variance or multilevel analysis. SPSS (IBM, Armonk, NY) also requires long formatted data if a multilevel analysis should be performed. Furthermore, we offer the export into the wide format (i.e., all responses of a participant are in one row and each column represents the condition combination of the relevant data variable) for the mean signals/frequencies, which is used for example by Jamovi (The jamovi project, 2020) or in SPSS (IBM, Armonk, NY) to calculate analysis of variance. Finally, we also offer the export of single-trial signal/frequency data in long format, which is required by R or SPSS to calculate multilevel mixed models.

**Note concerning single-trial analysis**

In this (pre-) processing chain, the opportunity of processing single trial data is provided although it is not the default option, but rather commented out to be used if activated. This was done to have some decent management over the necessary resources when using the processing pipeline the first time. Nevertheless, we want to encourage further exploration of your data and encourage the analysis of single-trial EEG responses because of the interesting time dynamics that may happen in your data that are mostly hidden if you just look at the mean responses over all trials. These trial-level responses may provide information about learning, boredom as well as surprise and fatigue. They are very helpful in understanding the data and its implications in

a better and maybe more precise way than just looking at the means. Also, inter-individual differences may hide in the variance of the responses, showing persons that are rather prone to be bored or similar reactions concerning the mentioned variables. Exploratory data analysis has an important role, but it is of course important to let it be guided by your hypotheses and for you to state explicitly what you had been looking for that motivated the exploration.

**Note concerning missing interesting analyses in this chain**

In this (pre-) processing chain, only a few analyses are provided and many interesting analyses like evoked and induced frequency responses (David, Kilner, & Friston, 2006; Galambos, 1992; Tallon-Baudry & Bertrand, 1999), cross-frequency coupling (e.g., with phase-amplitude coupling, Canolty & Knight, 2010), frequency phase distributions (e.g., Busch, Dubois, & VanRullen, 2009) and deeper source analysis with LORETA (Pascual-Marqui, Michel, & Lehmann, 1994; Pascual-Marqui, 1999) or similar algorithms are not included. Also, PCA based ERP peak detection (e.g., Dien, 2010; Kayser & Tenke, 2003) is not included. But why did we not include all these interesting and fancy methods? One reason was that we aimed to establish a very basic pre-processing chain, i.e., a standardized beginning on which anyone may build on. This chain can hopefully also be used by novices who try to get in touch with EEG and get inspired by the analyses they see to understand and get to know their data based on hypotheses, but also based on exploratory validation of established criteria. Of course, we also like to encourage you to explore your data in other and newer ways, but please keep in mind that a standardized result for hypothesis testing should be the first step, after which the data exploration follows. Another reason for not including some of the techniques mentioned above was the concern of introducing methods that are not that easy to understand and that may just be used as a "black box" without trying to think about them and to validate the results. After having seen many "odd" topographies, frequency response patterns and ERPs with questionable time-windows, we wanted to provide a standardized script that everyone is able to understand

and is able to validate their results quite quickly. Nevertheless, all suggestions that have been made in these scripts should also be seen with caution, as they may not be the perfect matching decisions for your specific data. The idea that is behind this standardized approach is to provide researchers with a decent starting point of a standardized approach, that can be modified and adjusted to their need, as long as they report their changes in order to get to replicable and transparent analyses.

## Conclusion

We presented a standardized, automated open-source processing pipeline for EEG data. In times where replicability and standardization are becoming more and more important to increase the robustness of research results, the pipelines necessary to fulfill this requirement are still not overly present, especially in the psychophysiological research domain. Therefore, we have presented a suggestion of a pipeline for (pre-)processing of EEG data as well as for detecting and graphically illustrating measured values, as a way to check the integrity of the processing results. We hope that the scripts included here will provide a basis to easily understand and replicate the analysis of future studies, as well as encourage people to explore their data and validate their results. In addition, an open and replicable pipeline may ensure that data sets from different sources could be transferred more easily into a joint analysis. The presented pipeline is not limited to ERP or frequency analysis but offers necessary code for both analyses and even single trial analysis. Nevertheless, it should still be mentioned, that this pipeline is merely a suggestion and may be adjusted to the respective needs of the data and paradigm. With this tool to get started with EEG data processing, one might hopefully develop a standardized and inspired way into analyzing the data and present valuable results to the scientific community.

**Acknowledgements**

The authors want to thank Prof. Patrick Mussel, Prof. Mike X Cohen, Prof. Edmund Wascher, the Wintersymposium Montafon and the training award from the Society of Psychophysiological Research.

## References

Altman, Y. (2020). export_fig (https://www.github.com/altmany/export_fig). *GitHub*.

Baldwin, S. A. (2017). Improving the rigor of psychophysiology research. *International Journal of Psychophysiology, 111*, 5-16. https://doi.org/10.1016/j.ijpsycho.2016.04.006

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation, 7*(6), 1129-1159. https://doi.org/10.1162/neco.1995.7.6.1129

Berger, H. (1929). Über das Elektrenkephalogramm des Menschen [On the electroencephalogram in humans]. *Archiv für psychiatrie und nervenkrankheiten, 87*(1), 527-570. https://doi.org/10.1007/BF01797193.

Bigdely-Shamlo, N., Mullen, T., Kothe, C., Su, K. M., & Robbins, K. A. (2015). The PREP pipeline: standardized preprocessing for large-scale EEG analysis. *Frontiers in Neuroinformatics, 9*, 16. https://doi.org/10.3389/fninf.2015.00016

Bishop, D. V. (2007). Using mismatch negativity to study central auditory processing in developmental language and literacy impairments: where are we, and where should we be going? *Psychological Bulletin, 133*(4), 651-672. https://doi.org/10.1037/0033-2909.133.4.651

Botvinik-Nezer, R., Holzmeister, F., Camerer, C. F., Dreber, A., Huber, J., Johannesson, M., . . . Schonberg, T. (2020). Variability in the analysis of a single neuroimaging dataset by many teams. *Nature*, 1-7. https://doi.org/10.1038/s41586-020-2314-9

Busch, N. A., Dubois, J., & VanRullen, R. (2009). The phase of ongoing EEG oscillations predicts visual perception. *Journal of Neuroscience, 29*(24), 7869-7876. https://doi.org/10.1523/JNEUROSCI.0113-09.2009

Canolty, R. T., & Knight, R. T. (2010). The functional role of cross-frequency coupling. *Trends in Cognitive Sciences, 14*(11), 506-515. https://doi.org/10.1016/j.tics.2010.09.001

Chaumon, M., Bishop, D. V., & Busch, N. A. (2015). A practical guide to the selection of independent components of the electroencephalogram for artifact correction. *Journal of Neuroscience Methods, 250*, 47-63. https://doi.org/10.1016/j.jneumeth.2015.02.025

Clayson, P. E., & Miller, G. A. (2017). ERP Reliability Analysis (ERA) Toolbox: An open-source toolbox for analyzing the reliability of event-related brain potentials. *International Journal of Psychophysiology, 111*, 68-79. https://doi.org/10.1016/j.ijpsycho.2016.10.012

Cohen, M. X. (2014). *Analyzing Neural Time Series Data: Theory and Practice*. Cambridge, Massachusetts, London, England: MIT press.

Cohen, M. X. (2017a). Rigor and replication in time-frequency analyses of cognitive electrophysiology data. *International Journal of Psychophysiology, 111*, 80-87. https://doi.org/10.1016/j.ijpsycho.2016.02.001

Cohen, M. X. (2017b). Where Does EEG Come From and What Does It Mean? *Trends in Neurosciences, 40*(4), 208-218. https://doi.org/10.1016/j.tins.2017.02.004

Cowley, B. U., Korpela, J., & Torniainen, J. (2017). Computational testing for automated preprocessing: a Matlab toolbox to enable large scale electroencephalography data processing. *PeerJ Computer Science, 3*, e108. https://doi.org/10.7717/peerj-cs.108

Cuevas, K., Cannon, E. N., Yoo, K., & Fox, N. A. (2014). The infant EEG mu rhythm: methodological considerations and best practices. *Developmental Review, 34*(1), 26-43. https://doi.org/10.1016/j.dr.2013.12.001

da Cruz, J. R., Chicherov, V., Herzog, M. H., & Figueiredo, P. (2018). An automatic pre-processing pipeline for EEG analysis (APP) based on robust statistics. *Clinical Neurophysiology, 129*(7), 1427-1437. https://doi.org/10.1016/j.clinph.2018.04.600

David, O., Kilner, J. M., & Friston, K. J. (2006). Mechanisms of evoked and induced responses in MEG/EEG. *Neuroimage, 31*(4), 1580-1591. https://doi.org/10.1016/j.neuroimage.2006.02.034

Dawes, R. M., Faust, D., & Meehl, P. E. (1989). Clinical versus actuarial judgment. *Science, 243*(4899), 1668-1674. https://doi.org/10.1126/science.2648573

Debener, S., Ullsperger, M., Siegel, M., Fiehler, K., von Cramon, D. Y., & Engel, A. K. (2005). Trial-by-trial coupling of concurrent electroencephalogram and functional magnetic resonance imaging identifies the dynamics of performance monitoring. *Journal of Neuroscience, 25*(50), 11730-11737. https://doi.org/10.1523/JNEUROSCI.3286-05.2005

Debnath, R., Buzzell, G. A., Morales, S., Bowers, M. E., Leach, S. C., & Fox, N. A. (2020). The Maryland analysis of developmental EEG (MADE) pipeline. *Psychophysiology, 57*(6), e13580. https://doi.org/10.1111/psyp.13580

Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods, 134*(1), 9-21. https://doi.org/10.1016/j.jneumeth.2003.10.009

Dien, J. (2010). The ERP PCA Toolkit: an open source program for advanced statistical analysis of event-related potential data. *Journal of Neuroscience Methods, 187*(1), 138-145. https://doi.org/10.1016/j.jneumeth.2009.12.009

Dimigen, O., Valsecchi, M., Sommer, W., & Kliegl, R. (2009). Human microsaccade-related visual brain responses. *Journal of Neuroscience, 29*(39), 12321-12331. https://doi.org/10.1523/JNEUROSCI.0911-09.2009

Eaton, J. W. (2002). GNU Octave Manual. Network Theory Limited. Retrieved from http://www.octave.org

Gabard-Durnam, L. J., Mendez Leal, A. S., Wilkinson, C. L., & Levin, A. R. (2018). The Harvard Automated Processing Pipeline for Electroencephalography (HAPPE): Standardized Processing Software for Developmental and High-Artifact Data. *Frontiers in Neuroscience, 12*, 97. https://doi.org/10.3389/fnins.2018.00097

Galambos, R. (1992). A comparison of certain gamma band (40-Hz) brain rhythms in cat and man. In E. Başar & T. H. Bullock (Eds.), *Induced rhythms in the brain*. Boston, MA: Birkhäuser.

Grove, W. M., Zald, D. H., Lebow, B. S., Snitz, B. E., & Nelson, C. (2000). Clinical versus mechanical prediction: a meta-analysis. *Psychological Assessment, 12*(1), 19-30. https://doi.org/10.1037/1040-3590.12.1.19

Hagemann, D. (2004). Individual differences in anterior EEG asymmetry: methodological problems and solutions. *Biological Psychology, 67*(1), 157-182. https://doi.org/https://doi.org/10.1016/j.biopsycho.2004.03.006

Herrmann, C., Grigutsch, M., & Busch, N. (2005). Event-related potentials: a methods handbook. *Todd Handy (Ed.)*, 229-259.

Hipp, J. F., & Siegel, M. (2013). Dissociating neuronal gamma-band activity from cranial and ocular muscle activity in EEG. *Frontiers in Human Neuroscience, 7*, 338. https://doi.org/10.3389/fnhum.2013.00338

Jung, T. P., Makeig, S., Humphries, C., Lee, T.-W., Mckeown, M. J., Iragui, V., & Sejnowski, T. J. (2000). Removing electroencephalographic artifacts by blind source separation. *Psychophysiology, 37*(2), 163-178. https://doi.org/10.1111/1469-8986.3720163

Junghöfer, M., Elbert, T., Tucker, D. M., & Braun, C. (1999). The polar average reference effect: A bias in estimating the head surface integral in EEG recording. *Clinical Neurophysiology, 110*(6), 1149-1155. https://doi.org/10.1016/S1388-2457(99)00044-9

Kayser, J. (2009). Current source density (CSD) interpolation using spherical splines-CSD Toolbox (Version 1.1). *New York State Psychiatric Institute: Division of Cognitive Neuroscience*.

Kayser, J., & Tenke, C. E. (2003). Optimizing PCA methodology for ERP component identification and measurement: theoretical rationale and empirical evaluation. *Clinical*

*Neurophysiology, 114*(12), 2307-2325. https://doi.org/10.1016/S1388-2457(03)00241-4

Kayser, J., & Tenke, C. E. (2006a). Principal components analysis of Laplacian waveforms as a generic method for identifying ERP generator patterns: I. Evaluation with auditory oddball tasks. . *Clinical Neurophysiology, 117*(2), 348-368. https://doi.org/10.1016/j.clinph

Kayser, J., & Tenke, C. E. (2006b). Principal components analysis of Laplacian waveforms as a generic method for identifying ERP generator patterns: II. Adequacy of low-density estimates. *Clinical Neurophysiology, 117*(2), 369-380. https://doi.org/10.1016/j.clinph.2005.08.033

Kearney, K. (2020). boundedline.m (https://www.github.com/kakearney/boundedline-pkg). *GitHub*.

Keil, A., Debener, S., Gratton, G., Junghöfer, M., Kappenman, E. S., Luck, S. J., . . . Yee, C. M. (2014). Committee report: publication guidelines and recommendations for studies using electroencephalography and magnetoencephalography. *Psychophysiology, 51*(1), 1-21. https://doi.org/10.1111/psyp.12147

Kim, T., Eltoft, T., & Lee, T.-W. (2006). *Independent vector analysis: An extension of ICA to multivariate components.* Paper presented at the International conference on independent component analysis and signal separation.

Korats, G., Le Cam, S., Ranta, R., & Hamid, M. (2012). *Applying ICA in EEG: choice of the window length and of the decorrelation method.* Paper presented at the International Joint Conference on Biomedical Engineering Systems and Technologies.

Larson, M. J., & Carbine, K. A. (2017). Sample size calculations in human electrophysiology (EEG and ERP) studies: A systematic review and recommendations for increased rigor. *International Journal of Psychophysiology, 111*, 33-41. https://doi.org/10.1016/j.ijpsycho.2016.06.015

Levin, A. R., Méndez Leal, A. S., Gabard-Durnam, L. J., & O'Leary, H. M. (2018). BEAPP: the batch electroencephalography automated processing platform. *Frontiers in Neuroscience, 12*, 513. https://doi.org/10.3389/fnins.2018.00513

Luck, S. (2005). An introduction to the event-related potential technique. In. Cambridge, Massachusetts, London, England: MIT Press.

Makeig, S., Debener, S., Onton, J., & Delorme, A. (2004). Mining event-related brain dynamics. *Trends in Cognitive Sciences, 8*(5), 204-210. https://doi.org/10.1016/j.tics.2004.03.008

MATLAB. (2011). Natick, Massachusetts: The MathWorks Inc.

Möcks, J., & Gasser, T. (1984). How to select epochs of the EEG at rest for quantitative analysis. *Electroencephalography and Clinical Neurophysiology, 58*(1), 89-92. https://doi.org/10.1016/0013-4694(84)90205-0

Mognon, A., Jovicich, J., Bruzzone, L., & Buiatti, M. (2011). ADJUST: An automatic EEG artifact detector based on the joint use of spatial and temporal features. *Psychophysiology, 48*(2), 229-240. https://doi.org/10.1111/j.1469-8986.2010.01061.x

Mullen, T. (2012). CleanLine EEGLAB plugin. *San Diego, CA: Neuroimaging Informatics Toolsand Resources Clearinghouse (NITRC)*.

Nolan, H., Whelan, R., & Reilly, R. B. (2010). FASTER: fully automated statistical thresholding for EEG artifact rejection. *Journal of Neuroscience Methods, 192*(1), 152-162. https://doi.org/10.1016/j.jneumeth.2010.07.015

Pascual-Marqui, R., Michel, C. M., & Lehmann, D. (1994). Low-resolution electromagnetic tomography–a new method for localizing electrical activity in the brain. *International Journal of Psychophysiology, 18*, 49-65.

Pascual-Marqui, R. D. (1999). Review of methods for solving the EEG inverse problem. *International Journal of Bioelectromagnetism, 1*(1), 75-86.

Pedroni, A., Bahreini, A., & Langer, N. (2019). Automagic: Standardized preprocessing of big EEG data. *Neuroimage, 200*, 460-473. https://doi.org/10.1016/j.neuroimage.2019.06.046

Perrin, F., Pernier, J., Bertrand, O., & Echallier, J. (1989). Spherical splines for scalp potential and current density mapping. *Electroencephalography and Clinical Neurophysiology, 72*(2), 184-187. https://doi.org/10.1016/0013-4694(89)90180-6

Perrin, F., Pernier, J., Bertrand, O., & Echallier, J. (1990). Corrigenda EEG 02274. *Electroencephalography and Clinical Neurophysiology, 76*, 565–566.

Picton, T. W., Bentin, S., Berg, P., Donchin, E., Hillyard, S. A., Johnson, R., . . . Taylor, M. J. (2000). Guidelines for using human event-related potentials to study cognition: Recording standards and publication criteria. *Psychophysiology, 37*(2), 127-152. https://doi.org/10.1111/1469-8986.3720127

Pivik, R. T., Broughton, R. J., Coppola, R., Davidson, R. J., Fox, N., & Nuwer, M. R. (1993). Guidelines for the recording and quantitative analysis of electroencephalographic activity in research contexts. *Psychophysiology, 30*(6), 547-558. https://doi.org/10.1111/j.1469-8986.1993.tb02081.x

R Core Team. (2020). R: A language and environment for statistical computing. *Vienna: R Foundation for Statistical Computing.*

Roach, B. J., & Mathalon, D. H. (2008). Event-related EEG time-frequency analysis: an overview of measures and an analysis of early gamma band phase locking in schizophrenia. *Schizophrenia Bulletin, 34*(5), 907-926. https://doi.org/10.1093/schbul/sbn093

Rodrigues, J., Liesner, M., Reutter, M., Mussel, P., & Hewig, J. (2020a). It's costly punishment, not altruistic: Low midfrontal theta and state anger predict punishment. *Psychophysiology*, e13557. https://doi.org/10.1111/psyp.13557

Rodrigues, J., Ziebell, P., Müller, M., & Hewig, J. (2020b). Tell me what I told you Mini-Me: Constructing and providing two layer feedforward networks for classification of continuous data in (and) a virtual T-maze. *Manuscript submitted for publication.*

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-Positive Psychology: Undisclosed Flexibility in Data Collection and Analysis Allows Presenting Anything as Significant. *Psychological Science, 22*(11), 1359–1366. https://doi.org/10.1177/0956797611417632

Tabachnick, B. G., & Fidell, L. S. (2007). *Using Multivariate Statistics* (Vol. 5). Boston, MA: Pearson.

Tallon-Baudry, C., & Bertrand, O. (1999). Oscillatory gamma activity in humans and its role in object representation. *Trends in Cognitive Sciences, 3*(4), 151-162. https://doi.org/10.1016/s1364-6613(99)01299-1

The jamovi project. (2020). jamovi [Computer Software]. Retrieved from https://www.jamovi.org.

Whitham, E. M., Pope, K. J., Fitzgibbon, S. P., Lewis, T., Clark, C. R., Loveless, S., . . . Willoughby, J. O. (2007). Scalp electrical recording during paralysis: quantitative evidence that EEG frequencies above 20 Hz are contaminated by EMG. *Clinical Neurophysiology, 118*(8), 1877-1888. https://doi.org/10.1016/j.clinph.2007.04.027

Winkler, I., Brandl, S., Horn, F., Waldburger, E., Allefeld, C., & Tangermann, M. (2014). Robust artifactual independent component classification for BCI practitioners. *Journal of Neural Engineering, 11*(3), 035013. https://doi.org/10.1088/1741-2560/11/3/035013

Winkler, I., Debener, S., Müller, K.-R., & Tangermann, M. (2015). *On the influence of high-pass filtering on ICA-based artifact reduction in EEG-ERP.* Paper presented at the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).

Winkler, I., Haufe, S., & Tangermann, M. (2011). Automatic classification of artifactual ICA-components for artifact removal in EEG signals. *Behavioral and Brain Functions, 7*(1), 30. https://doi.org/10.1186/1744-9081-7-30

Wu, W., Keller, C. J., Rogasch, N. C., Longwell, P., Shpigel, E., Rolle, C. E., & Etkin, A. (2018). ARTIST: A fully automated artifact rejection algorithm for single-pulse TMS-EEG data. *Human Brain Mapping, 39*(4), 1607-1625. https://doi.org/10.1002/hbm.23938