TECHNICAL REPORT

# Zapline-plus: a Zapline extension for automatic and adaptive removal of frequency-specific noise artifacts in M/EEG

**Marius Klug*[1] | Niels A. Kloosterman[2,3]**

[1]Biopsychology and Neuroergonomics, Institute of Psychology and Ergonomics, Technische Universität Berlin, Berlin, 10623, Germany

[2]Max Planck UCL Centre for Computational Psychiatry and Ageing Research, Berlin, Germany

[3]Center for Lifespan Psychology, Max Planck Institute for Human Development, Berlin, Germany

**Correspondence**
*Marius Klug, Biopsychology and Neuroergonomics, Technische Universität Berlin, Berlin, 10623, Germany
Email: marius.klug@tu-berlin.de

**Funding information**
This work was supported by the DFG (GR2627/8-1) and USAF (ONR 10024807).

Removing power line and other frequency-specific artifacts

from electrophysiological data without affecting neural signals remains a challenging task. Recently, an approach was introduced that combines spectral and spatial filtering to effectively remove line noise: Zapline (de Cheveigné, 2020). This algorithm, however, requires manual selection of the noise frequency and the number of spatial components to remove during spatial filtering. Moreover, it assumes that noise frequency and spatial topography are stationary over time, which is often not warranted. To overcome these issues, we introduce Zapline-plus, which allows adaptive and automatic removal of frequency-specific noise artifacts from M/EEG data. To achieve this, our extension first searches for peaks in the spectrum, then applies Zapline iteratively to chunks of data to account for non-stationarity of the noise over time. Likewise, the exact noise frequency around the found target frequency is determined separately for every chunk to allow fluctuations of the peak noise frequency. The number of to-be-removed components is automatically determined using an outlier detection algorithm. Finally, the frequency spectrum after cleaning is analyzed for suboptimal cleaning and parameters are adapted accordingly before re-running the process if necessary. The software creates a detailed plot for analysis of the cleaning that can be used to inform manual changes. We showcase the efficacy of the different features of our algorithm by applying it to four openly available data sets, two EEG sets containing both stationary and mobile conditions, and two MEG sets.

**KEYWORDS**
M/EEG, artifact removal, line noise removal, preprocessing

## 1 | INTRODUCTION

The experiment is set up and running, the paradigm is well thought out, a masterpiece, really. Data recording goes well, 30 data sets and more, and finally you have time to plot your first analysis. Then: peaks in your spectra, particularly at 50/60 Hz, but also in other frequencies, right where you want to analyze your data. Removing frequency-specific noise artifacts from electrophysiological data is a key issue in any electroencephalography (EEG) or magnetoencephalography (MEG) experiment. Modern laboratories contain many different electrical devices that all need power, and with great power comes great line noise. But noise is not only limited to the 50/60 Hz power line artifact, but may also

arise from many different sources. Recently, a novel algorithm, Zapline, was introduced that combines spectral and spatial filters to isolate and remove the power line noise (de Cheveigné, 2020). In this paper, we present an adaptive wrapper software for Zapline to enable the fully automatic removal of frequency-specific noise artifacts, including the selection of noise frequencies, chunking the data to alleviate issues of non-stationarity, automatically selecting the number of components to remove with Zapline, as well as a comprehensive analysis and visualization of the cleaning and its impact on the data.

**EEG noise removal is especially difficult in mobile experiments.** Mobile EEG studies require specific treatment to remove noise stemming from muscles and other sources, and often independent component analysis (ICA) can be used for this (Klug and Gramann, 2020). Finding the right way to remove frequency-specific noise from the data, however, is a difficult task, especially since it it is not necessarily spatially stable and thus can have a strong negative impact on ICA. Shielding the laboratory, finding the sources and eliminating them before recording the data help to alleviate the issue, but this is not always feasible, and sometimes the noise goes unnoticed at first.

As recent developments in EEG experimental paradigms show a trend towards measuring the human in its natural habitat - the world (Gramann et al., 2014) - it can become increasingly difficult or impossible to control noise sources. The fields of mobile brain/body imaging (Makeig et al., 2009; Gramann et al., 2011; Jungnickel et al., 2018) and neuroergonomics (Parasuraman and Rizzo, 2007; Dehais et al., 2020) use devices like virtual reality head mounted-displays, motion tracking, eye tracking, treadmills, flight simulators, or actual airplanes, and more. In these experiments, participants move around and interact with the world, including for example navigating through a city (Wunderlich and Gramann, 2018), a virtual maze (Gehrke and Gramann, 2021), or flying an airplane (Dehais et al., 2019). These data sets are almost always riddled with frequency-specific noise, not only stemming from the power line but also from other devices, and often it is just accepted that recordings contain noise.

Removing this noise during processing is especially important when comparing different conditions like seated vs. mobile experiments, as different noise sources may be nearby for the different conditions, and untreated noise can be wrongfully interpreted as an effect of the conditions.

**Line noise artifacts are particularly strong in MEG.** Magnetoencephography (MEG) is a technique closely related to EEG, in which rather than electrical activity itself, its concurrent magnetic fields are recorded (Hämäläinen et al., 1993). Compared to EEG, MEG allows for better spatial specificity of (superficial) sources of neural activity in the brain. Moreover, it does not require extended subject preparation and electrode gel, which makes MEG more feasible for clinical populations as well as children. Magnetic fields are less distorted by the skull than electrical activity, which makes MEG better suited for investigating high-frequency neural activity in the so-called gamma band (although gamma is investigated in EEG as well, e.g. Kloosterman et al. (2019)). However, the gamma band ranges from roughly 40 to 100 Hz (Hoogenboom et al., 2006), which encompasses the 50 or 60 Hz line noise frequency (and its first harmonic), to which MEG is highly sensitive and which can outweigh neural activity by several orders of magnitude. This noise is often removed using strong filters (see next section) which come at the cost of removing true neural activity in this range as well. This approach hampers in-depth investigation of the function of gamma activity in neural processing.

**Noise can be removed with spectral filters, regression, or spatial filters.** Taken together, removing frequency-specific noise is a vital part of data processing. Several methods are available to remove this noise, but these all come with individual drawbacks. Three main approaches can be distinguished: (i) Spectral filters: Filtering the data with a simple low-pass or notch filter is a classic approach. However, a low-pass filter may reduce the quality of decomposing the

data using Independent Component Analysis (ICA; Dimigen, 2020; Hyvärinen and Oja, 1997) and a notch filter must have a steep roll-off to keep the notch small, which comes with the potential of ringing artifacts (Widmann et al., 2015). Additionally, both options remove all information in (or even above) the noise range and will make analysis of these frequencies impossible. (ii) Regression-based approaches: Regressing a target signal out of the data is another often used tool. Examples are the CleanLine plugin of EEGLAB (Delorme and Makeig, 2004), which uses a frequency-domain regression to remove sinusoidal artifacts from the data, or TSPCA, which uses a provided reference signal (de Cheveigné and Simon, 2007). These approaches are dependent on either a provided reference or a successful generation of a target signal in a given frequency. Here, some noise may be left in the data, especially fluctuations in amplitude or phase of the noise can be difficult to remove. (iii) Spatial filters: Spatial filter options like ICA or joint diagonalization (JD, de Cheveigné and Parra, 2014) are widely used and reduce noise by generating their own noise reference signal from a linear combination of all channels. However, noise is not always linearly separable from sources of interest like the brain, and thus removing noise components can inadvertently remove brain signals, too. These methods are also vulnerably to non-stationary noise, which can be particularly problematic in mobile EEG experiments. Finally, removing noise components from the data with a spatial filter relying on linear algebra always reduces the rank of the data and can thus limit further analyses (Cohen, 2021). In sum, all of the above options can be problematic.

**Zapline is a promising option.** Recently, a promising new method that combines the spectral and spatial filtering approaches has been introduced: Zapline (de Cheveigné, 2020). Zapline first uses a notch filter and its complementary counterpart to split the data into the clean and the noisy part, where summing them together would result in the original data. Then the noisy part is decomposed using JD (de Cheveigné and Parra, 2014) and the first components that carry most of the noise are removed from the noisy data. Last, the now cleaned, previously noisy, data and the clean data are summed together to form the final cleaned data set. This approach has the advantage of not leaving a notch in the spectrum while also not reducing the rank of the data.

**Drawbacks of Zapline.** However, some issues remain. On the one hand, as Zapline makes use of a spatial filter, it assumes stationarity of the noise. But especially in mobile experiments the spatial distribution of the noise can change (proximity changes of devices, orientation changes of the participant, touching cables, etc.). When comparing different conditions, it may even be the case that some noise artifacts are entirely absent in parts of the recording. This issue can lead to insufficient cleaning in some, too much cleaning in other parts of the data, or the need to remove many components, which can distort the data. On the other hand, a key challenge when using Zapline is that it needs to be tuned to each data set: (i) Finding out the correct number of components to remove is not straightforward, recommendations range from two to four (de Cheveigné, 2020) to 25 components (Miyakoshi et al., 2021), depending on the noise level. In our tests with high-density EEG and MEG data, removing ten to fifteen components was usually sufficient. (ii) The noise frequency needs to be chosen. In most cases, choosing the power line frequency is sufficient, but sometimes additional frequencies can be found, like a 90 Hz oscillation of a virtual reality head-mounted display, or other frequencies due to additional devices in the lab. Additionally, in some of our tests Zapline proved to be sensitive to even small changes in the target frequency in the range of 0.1 Hz, which are hard to know in advance, especially if the frequency shifts during the recording. Taken together, Zapline is a powerful tool with some limitations, and using Zapline in an automated analysis pipeline is difficult due to the necessary fine tuning.

**Summary of our approach.** We created Zapline-plus: an adaptive wrapper software for Zapline which allows fully automatic use without parameter tuning. The software searches for outlier peaks in the spectrum and applies Zapline to remove these. To alleviate the stationarity issue, the data is cut into chunks of 150 s duration and within each

chunk, the individual chunk noise peak frequency is detected. Zapline is then applied to each chunk individually, and an adaptive detector removes only the most noisy components. Last, we check if the cleaning was too weak or too strong and adjust the detector accordingly. All used parameters and several performance indicators are stored to enable an understanding and easy replication of the cleaning, and a detailed plot is created to allow inspection of the cleaning performance.

We tested the software on two open EEG and two open MEG data sets with promising results. We discuss limitations and implications for automated processing pipelines. The MATLAB source code of the software is available for download at https://github.com/MariusKlug/zapline-plus.

# 2 | THE SOFTWARE PACKAGE

In this section we describe the different aspects of the adaptive algorithm, the processing flow, as well as the produced plots and the optional parameters in case the default values are suboptimal.

## 2.1 | Algorithm

The algorithm contains several components that are discussed in the following. The processing includes (1) the detection of noise frequencies, (2) the cleaning in chunks, (3) the automatic detection of noise components, and (4) the adaptive changes in order to prevent too weak or too strong cleaning. The processing flow is visualized in figure 1.

### 2.1.1 | Noise frequency detection

Outlier frequencies are defined as having abnormally large power compared to the surrounding frequencies. In order to detect them, a spectrum is computed using Welch's method (Welch, 1967) with a 150s long hanning window. We chose a long window because it enables a high resolution of the frequency, and a hanning window because it resulted in less noisy spectra than the default hamming window for some data sets. The computed power spectral density (PSD) values are then log transformed ($10log_{10}$) and the mean over channels is computed (corresponding to a geometric mean of the spectra that is less outlier-driven). Subsequently, the first outlier frequency withing a minimum (17 Hz) and maximum (99 Hz) frequency is searched with a 6 Hz moving window. If a frequency has a difference >4 of log PSD to the center log PSD (mean of left and right thirds around the current frequency), it is found to be an outlier and the search is stopped. As the input is in $10log_{10}$ space, a difference of 4 corresponds to a ~2.5x increase of the outlier power over the center.

### 2.1.2 | Cleaning in chunks

Zapline uses a spatial filtering approach internally and thus works on the assumption of a stationary spatial noise distribution, but this is not always given. Even small shifts in head orientation or a relocation of the participant due to the experimental paradigm can lead to slightly different noise topography or entirely new noise sources. While this issue is impossible to prevent entirely, we can alleviate the effect by splitting the data in smaller chunks (e.g. 150 s) and running Zapline on these chunks individually. This does not only allow different linear decompositions per chunk, but also allows fine-tuning of the target frequency to the peak in this chunk, improving the cleaning capacity of Zapline further. We chose a chunk length of 150 s because smaller windows resulted in both, a sometimes suboptimal

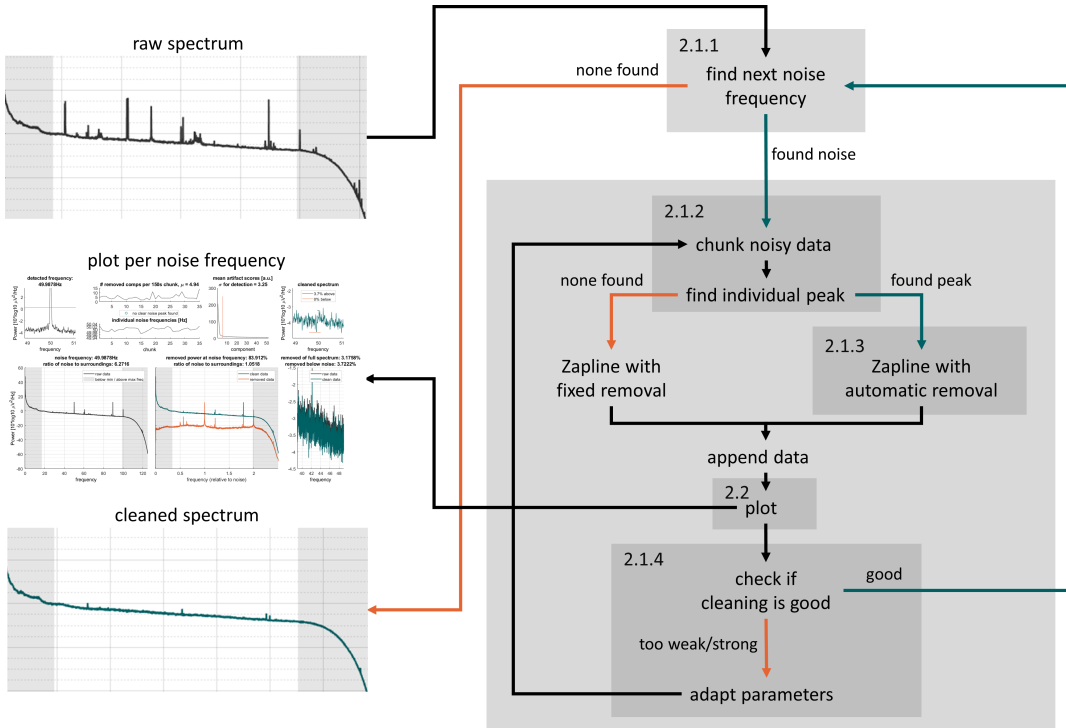


**FIGURE 1** Processing flow of the Zapline-plus algorithm.

decomposition within Zapline and a lower frequency resolution for the chunk noise peak detector.

To detect the chunk noise peak we first search for the peak frequency within a ±0.05 Hz range around the previously detected target frequency. We then determine a fine-grained threshold to define oscillations being present or absent in that chunk: The mean of the two lower 5% log PSD quantiles of the first and last third in a 6 Hz area around the target frequency is computed, and the difference to the center power (mean of left and right third log PSDs around the target frequency) is taken as a measure of deviation from the mean. Both the standard deviation and the median absolute deviation did not lead to good results, as they can be driven by outliers to the top. Finally, the threshold is defined as the center power + 2 $*$ deviation measure, and if the log PSD of the found peak frequency is above this threshold, the chunk is found to have a noise artifact.

In the next step the chunk is cleaned with Zapline using either the found frequency peak and adaptive removal settings (starting with $3\sigma$, see section 2.1.3, adaptive, see section 2.1.4), or the original noise peak and a fixed number of components to remove (starting at 1, adaptive, see section 2.1.4). We chose to remove a fixed number even when no artifact was found, to make sure even missed artifacts are getting cleaned while also making sure not too many components are removed in case no artifact is actually present in the chunk at that frequency.

### 2.1.3 | Detection of the number of noise components

The core parameter of Zapline is the number of removed components, and so far, this had to be chosen manually. We adapted the function to include a detector for outliers in the computed JD scores that represent the "noisiness" of the

components. To this end, an iterative approach based on a standard mean + x $\sigma$ threshold is used. In each iteration the detector removes outliers, then recomputes mean and $\sigma$, removes outliers again, until none are left. The number of removed outliers is then taken as the number of components to remove in Zapline. We found this iterative approach to be more robust than an approach based on the median absolute deviation in this scenario. In a final step, if the number of found outliers is less than the entered fixed removal, the latter is being used, and, to prevent removing an unreasonable amount of components, the number is capped at 1/5th of the components. We found a value of $3\sigma$ to work well in most cases, but sometimes even this automatic detector removes too many or too few components, which is why the $\sigma$ parameter is adapted in the next step.

### 2.1.4 | Adaptive changes

After each chunk is cleaned, the chunks are concatenated again and the cleaned spectrum is computed as in section 2.1.1. Although the software already contains many steps to find an optimal noise reduction, the cleaning can still be too weak or too strong. We implemented a check for suboptimal cleaning by using the same fine-grained threshold as in section 2.1.2. This check is now applied to search for introduced notches or remaining peaks: If there are 0.5% samples of the spectrum in the range of ±0.05 Hz around the noise frequency above the threshold of center power + 2 $*$ deviation measure, the cleaning is found to be too weak. If there are 0.5% samples of the spectrum in the range of -0.4 to +0.1 Hz around the noise frequency below the threshold of center power - 2 $*$ deviation measure, the cleaning is found to be too strong. In short, if a peak remains or a notch is introduced, the cleaning is deemed to be too weak or too strong, respectively.

If the cleaning was too weak, the $\sigma$ for step 2.1.3 is reduced by 0.25, up to a minimum of 2.5, and the fixed number of removed components (for chunks where no noise was detected) is increased by 1. If the cleaning was too strong, the $\sigma$ for step 2.1.3 is increased by 0.25, up to a maximum of 4, and the fixed number of removed components (for chunks where no noise was detected) is decreased by 1, up to a minimum of the initial fixed removal of 1. A too strong cleaning always takes precedence over too weak cleaning, and if the cleaning was once found to be too strong, it can never become stronger again even after it was weakened and is now found to be too weak.

Using these new values, the entire cleaning of this noise frequency is re-run and evaluated again. This leads to a maximally reduced noise artifact while ensuring minimal impact on any other frequencies. If no further adaptation of the cleaning needs to be performed, this noise frequency is assumed to be cleaned, and the next noise frequency is searched (see section 2.1.1) using the current noise frequency +0.05 Hz as the new minimum frequency. If no more noise frequency is found, the cleaning completes.

## 2.2 | Plots

For every noise frequency that is cleaned, a figure is generated. Example plots can be seen in figures 2 and 3. Importantly, the plot per frequency is being overwritten in case the parameters are adapted in step 2.1.4, so the final plots only show the final values. These plots contain all information that is necessary to determine the success of the cleaning in a colorblind-friendly color scheme. The top row of the figure contains visualizations of the cleaning process, the bottom row contains the final spectra and analytics information.

In the top row, first, the noise frequency of this iteration is shown in a zoomed-in spectrum to ±1.1 Hz around the frequency. The threshold that led to the detection of this frequency is shown in addition, unless the detection is disabled. Next, the cleaning of the individual chunks is visualized in two ways: The number of removed components per chunk, and the individual noise frequency detected for each chunk. Additionally, chunks in which no noise was

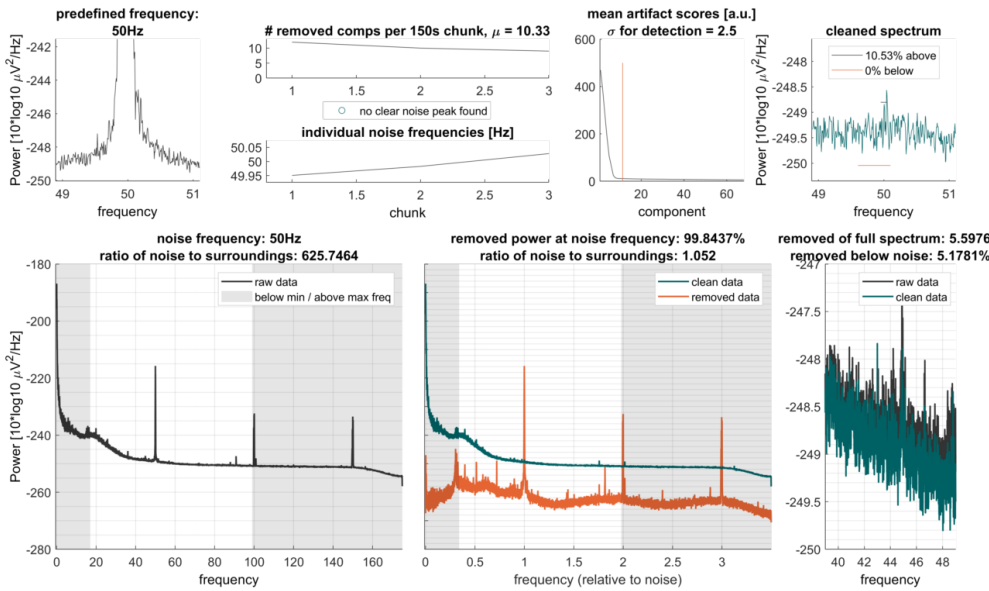


**FIGURE 2** Example plots of a 9 min MEG data set from MEG study I (see section 3.1), 50 Hz was predefined as the noise to remove. For an explanation of the individual subplots, see section 2.2.

detected are marked as such and the mean number of removed components is denoted in the title of the plot. As each chunk contains a set of components and accompanying artifact scores, this is too much to be visualized without cluttering the plot, so we chose to only plot the mean artifact scores over all chunks next. This plot also contains the mean number of removed components again as a vertical line. Ideally, this line should cross the scores around the "elbow" of the curve, which indicates that the outliers, the components which carry the most noise, were detected correctly. The abscissa is cut to one third of the number of components to allow the visualization of the knee point. This is independent of the nkeep parameter that can be set (see section 2.3). The $\sigma$ value that was used for the detector is denoted in the title of this plot. To finalize the visualization of the cleaning process, the zoomed-in spectrum of the cleaned data is shown alongside the thresholds that determine if the cleaning was too strong or too weak with respective horizontal lines. The legend of this plot also contains the proportion of frequency samples that are below or above these thresholds, which determines whether the cleaning needs to be adapted. It may happen that values exceeding these thresholds remain, which can be either due to the minimum or maximum $\sigma$ being reached or due to the fact that the cleaning would to too strong if set to a stronger level.

The bottom left plot shows the raw spectrum as the mean of the log-transformed channel spectra. Vertical shaded areas denote the minimal and maximal frequency to be checked by the detector, as this can be useful to know in case a spectral peak is present in this area and thus goes undetected. In the bottom mid plot the spectra of the cleaned (green), as well as the removed data (red), are plotted. The abscissa in this plot is relative to the noise frequency which facilitates distinguishing removed harmonics from other frequencies. Last, as it was shown that Zapline can have undesirable effects on the spectrum below the noise frequency (Miyakoshi et al., 2021), the bottom right plot shows the spectra of the raw and cleaned data again zoomed in to the part 10 Hz below the noise frequency to determine if

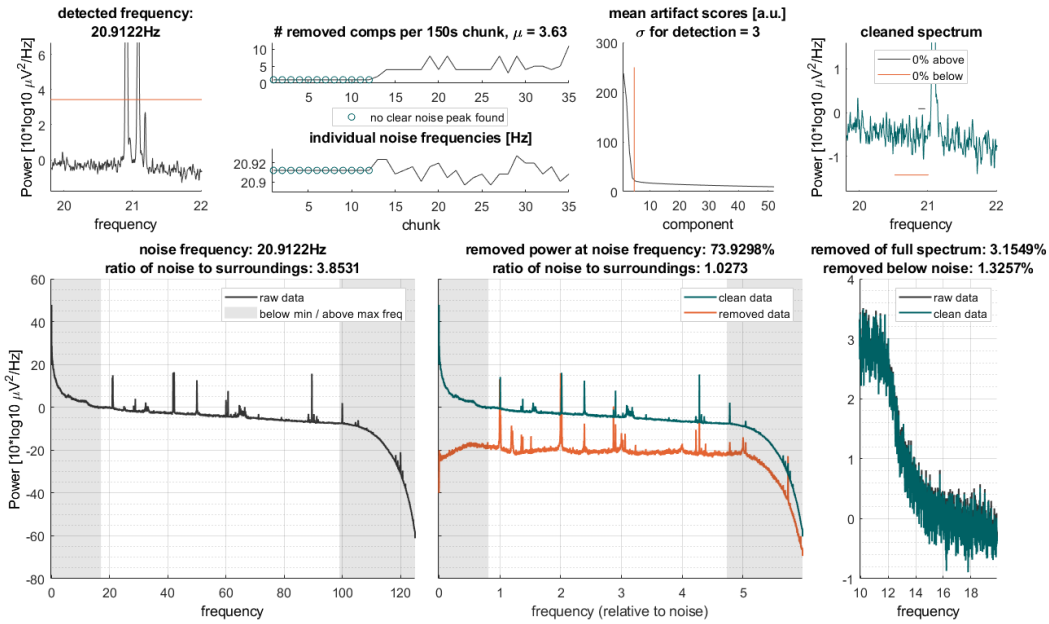


**FIGURE 3** Example plots of a 87 min EEG data set from EEG study II (see section 3.1) containing a mobile and a stationary condition. The plot shows the first of six cleaning steps with automatic noise detection. This noise artifact was present only in the first part of the data. For an explanation of the individual subplots, see section 2.2.

this was the case. In the title of these plots we also denote several analytics values: the proportion of removed power (computed on log-transformed data, corresponding to the geometric mean) of the complete spectrum, of the power ± 0.05 Hz around the noise frequency, and of the power -11 Hz to -1 Hz below the noise frequency, as well as the ratios of power ± 0.05 Hz around the noise frequency to the center power before and after cleaning.

These plots facilitate both, an understanding of the data set itself, as well as the functioning of the cleaning. Although the algorithm is adaptive in many ways and should work "as is", it is naturally possible that the noise has properties that make cleaning with Zapline-plus difficult or impossible. Hence, these plots should always be inspected to determine if the cleaning was successful.

## 2.3 | Parameters and outputs

Although we strive to provide a fully automatic solution with no need for parameter tweaking, we want to provide options for all relevant aspects of the algorithm, including switching adaptations off in case they don't work as intended. Here, we describe the parameters, our reasoning for the default values and reasonable ranges, as well as the output of the cleaning and additional thoughts. The data and sampling rate are required inputs, all additional parameters can be entered either in key-value pairs or as a single struct:

**noisefreqs (default = empty):** Vector with one or more noise frequencies to be removed. If empty or missing, noise frequencies will be detected automatically. Individual chunk peak detection will still be applied if set.

**minfreq (default = 17):** Minimum frequency to be considered as noise when searching for noise frequencies automat-

ically. We chose this default as it is well above the potentially problematic range of alpha oscillations (8 - 13 Hz) and also above the third subharmonics of 50 Hz, which was present in some MEG data sets.

**maxfreq (default = 99):** Maximum frequency to be considered as noise when searching for noise freqs automatically. We chose this default as is is below the second harmonics of the 50 Hz line noise. If the line noise can not be removed successfully in the original frequency, trying to remove the harmonics can potentially lead to overcleaning.

**adaptiveNremove (default = true):** Boolean if the automatic detection of number of removed components (see section 2.1.3) should be used. If set to false, a fixed number of components will be removed in all chunks. As this is a core feature of the algorithm it is switched on by default.

**fixedNremove (default = 1):** Fixed number of removed components per chunk. If adaptiveNremove is set to true, this will be the minimum. Will be automatically adapted if "adaptiveSigma" is set to true. We chose this default to remove at least one component at all times, no matter whether or not a noise oscillation was detected per chunk, as the detector can fail to find an oscillation that should be removed, and removing a single component does not lead to a large effect if no oscillation was present in the chunk.

**detectionWinsize (default = 6):** Window size in Hz for the detection of noise peaks. As the detector uses the lower and upper third of the window to determine the center power (see section 2.1.1) this leaves a noise bandwidth of 2 Hz. In our tests, some data sets indeed had such a large bandwidth of line noise, which can occur if the noise varies in time.

**coarseFreqDetectPowerDiff (default = 4):** Threshold in $10log_{10}$ scale above the center power of the spectrum to detect a peak as noise frequency. If this is too high, weaker noise can go undetected and thus uncleaned. If it is too low, spurious peak oscillations can be wrongfully classified as noise artifacts. This default corresponds to a 2.5 x increase of the noise amplitude over the center power in the detection window which worked well in our tests.

**coarseFreqDetectLowerPowerDiff (default = 1.76):** Threshold in $10log_{10}$ scale above the center power of the spectrum to detect the end of a noise artifact peak. This is necessary for the noise frequency detector to stop. This default corresponds to a 1.5 x increase of the noise amplitude over the center power in the detection window which worked well in our tests.

**searchIndividualNoise (default = true):** Boolean whether or not individual noise peaks should be applied on the individual chunks instead of the noise frequency specified or found on the complete data (see section 2.1.2). As this is a core feature of the algorithm it is switched on by default.

**freqDetectMultFine (default = 2):** Multiplier for the 5% quantile deviation detector of the fine noise frequency detection for adaption of sigma thresholds for too strong/weak cleaning (see section 2.1.2). If this value is lowered, the adaptive changes of section 2.1.4 are stricter, if it is increased, these adaptations happen more rarely.

**detailedFreqBoundsUpper (default = [-0.05 0.05 ):]** Frequency boundaries for the fine threshold of too weak cleaning. This is also used for the search of individual chunk noise peaks as well as the computation of analytics values of removed power and ratio of noise power to surroundings. Low values mean a more direct adaptation to the peak, but too low values might mean that the actual noise peaks are missed.

**detailedFreqBoundsLower (default = [-0.4 0.1 ):]** Frequency boundaries for the fine threshold of too strong cleaning. Too strong cleaning usually makes a notch into the spectrum slightly below the noise frequency which is why these boundaries are not centered around the noise peak.

**maxProportionAboveUpper (default = 0.005):** Proportion of frequency samples that may be above the upper threshold before cleaning is adapted. We chose this value since it allows a few potential outliers before adapting the cleaning.

**maxProportionBelowLower (default = 0.005):** Proportion of frequency samples that may be below the lower threshold before cleaning is adapted. We chose this value since it allows a few potential outliers before adapting the cleaning.

**noiseCompDetectSigma (default = 3):** Initial sigma threshold for iterative outlier detection of noise components to be removed (see section 2.1.3). Will be automatically adapted if "adaptiveSigma" is set to 1. This value led to the fewest adaptations in our tests.

**adaptiveSigma (default = 1):** Boolean if automatic adaptation of noiseCompDetectSigma should be used. Also adapts fixedNremove when cleaning becomes stricter (see section 2.1.4). As this is a core feature of the algorithm it is switched on by default.

**minsigma (default = 2.5):** Minimum when adapting noiseCompDetectSigma. We found that a lower sigma than 2.5 usually resulted in removing too many components and a distortion of the data.

**maxsigma (default = 4):** Maximum when adapting noiseCompDetectSigma. We found a high sigma than 4 usually didn't relax the cleaning meaningfully anymore.

**chunkLength (default = 150):** Length of chunks to be cleaned in seconds. If set to 0, no chunks will be used. We chose a chunk length of 150s because smaller windows resulted in both, a sometimes suboptimal decomposition within Zapline and a lower frequency resolution for the chunk noise peak detector. Smaller chunks result in better adaptation to non-stationary noise, but also potentially worse decomposition within Zapline. The necessary chunk length for ideal performance may also depend on the sampling rate.

**winSizeCompleteSpectrum (default = 0):** Window size in samples of the pwelch function to compute the spectrum of the complete data set for detecting the noise frequencies. If 0, a window length of sampling rate x chunkLength is used. This parameter mainly adjusts the resolution of the computed spectrum. We chose relatively long windows to ensure a high resolution for the noise frequency detector.

**nkeep (default = 0):** Principal Component Analysis dimension reduction of the data within Zapline. If 0, no reduction will be applied. This can be useful for extremely high number of channels, but in our tests even on high-density EEG and MEG data it did not lead to better results.

**plotResults (default = 1):** Boolean if plot should be created.

After completing the cleaning, Zapline-plus passes out the complete configuration struct including all adaptations that happened during the cleaning. This allows a perfect replication of the cleaning when applying the configuration to the same raw data again and facilitates reporting the procedure. Additionally, the generated analytics values that can be found in the plot are also passed out as a struct: raw and final cleaned log spectra of all channels, sigma used for detection, proportion of removed power of the complete spectra, the noise frequencies, and below noise frequencies, ratio of noise powers to surroundings before and after cleaning per noise frequency, proportion of spectrum samples above/below the threshold for each frequency, matrices of number of removed components per noise frequency and chunk, of artifact component scores per noise frequency and chunk, of individual noise peaks found per noise frequency and chunk, and whether or not the noise peak exceeded the threshold, per noise frequency and chunk. These values allow an easy check of the complete Zapline-plus cleaning both for each subject and on the group-level.

## 2.4 | A note on the sampling rate of the data

Modern M/EEG setups typically record data at high sampling rates of at least 500 Hz (1200 Hz is common for MEG), which allows for high temporal resolution and investigation of very high frequencies. However, brain activity is typically not quantified beyond 100 Hz, and lower sampling rates such as 250 Hz are typically deemed sufficient for

ERP studies investigating the onset of neural responses. Importantly, the presence of high frequencies poses a major challenge for line noise removal with Zapline, because Zapline also needs to handle the harmonics (multiples of the line noise frequency) that emerge with frequency-specific noise. For example, at a sample rate of 1200 Hz, Zapline will try to remove line noise at 50 Hz also at multiples of 50 Hz all the way up to 600 Hz (Nyquist frequency), yielding as many as twelve harmonics. We noticed that Zapline performed worse with higher sampling rates. Thus, to make Zapline's task easier, it is advisable to downsample the data prior to running Zapline-plus. For the MEG data analyzed here, we down-sampled to 350 Hz, for the EEG data to 250 Hz, such that only 50 and 100 Hz, and 150 Hz for the MEG data, are considered for noise removal. Indeed, we found that Zapline-plus performed much better at lower sampling rates.

# 3 | EXAMPLE APPLICATION

## 3.1 | Data sets

In order to test the efficacy of the Zapline-plus algorithm we ran it on four different open data studies, two EEG data sets containing both stationary and mobile conditions, and two stationary MEG data sets.

**EEG study I** This is an open data set available at `https://openneuro.org/datasets/ds003620/versions/1.0.2` (Liebherr et al., 2021). Data of 41 participants (aged 18-39 years, $\mu$ = 23.1 years, 26 female and 15 male) is available, of which we only used 24 sets for technical reasons. The experiment consisted of an auditory oddball task which was administered either in a laboratory environment, or on a grass field, or on the campus of the University of South Australia. Continuous EEG data was recorded with a 500 Hz sampling rate using 32 active Ag/AgCl electrodes and the BrainVision LiveAmp (Brain Products GmbH, Gilching, Germany). Electrode impedances were kept below 20kΩ and channels were referenced to the FCz electrode. See Liebherr et al. (2021) for detailed information.

**EEG study II** This is an open data set available at `http://dx.doi.org/10.14279/depositonce-10493` (Gramann et al., 2021). Data of 19 participants (aged 20-46 years, $\mu$ = 30.3 years, 10 female and 9 male) is available, of which we used all. The experiment consisted of a rotation on the spot, which either happened in a virtual reality environment with physical rotation or in the same environment on a 2D monitor using a joystick to rotate the view. EEG data for each condition was recorded with a 1000 Hz sampling rate using 157 active Ag/AgCl electrodes (129 on the scalp in a custom equidistant layout, 28 around the neck in a custom neck band) and the BrainAmp Move System (Brain Products GmbH, Gilching, Germany). Electrode impedances were kept below 10kΩ for scalp electrodes and below 50kΩ for neck electrodes, and channels were referenced to the FCz electrode. See Gramann et al. (2021) for detailed information.

**MEG study I** This open data set is available at `https://data.donders.ru.nl/collections/di/dccn/DSC_3011020.09_236?0`. We randomly selected 12 of the 204 subjects to test Zapline-plus. Subjects performed a language task, during which they had to process linguistic utterances that either consisted of normal or scrambled sentences. Four of the analyzed subjects were reading the stimuli (subject IDs V1001, V1012, V1024, V1036), the other eight listened to the stimuli (subject IDs A2027, A2035, A2051, A2064, A2072, A2088, A2101, A2110). Magnetoencephalographic data were collected with a 275-channel axial gradiometer system (CTF). The MEG recording for each subject lasted ca. 45 minutes. The signals were digitized at a sampling frequency of 1200 Hz (cutoff frequency of the analog anti-aliasing low pass filter was 300 Hz). See Schoffelen et al. (2019) for details.

**MEG study II** This data set comprises open MEG data from the Cam-CAN set of the Cambridge Centre for Ageing and Neuroscience, available at `http://www.mrc-cbu.cam.ac.uk/datasets/camcan/` (Taylor et al., 2017; Shafto et al., 2014). We randomly selected 23 of the 647 participants. Participants performed a sensory motor task on audio-visual stimuli (bilateral sine gratings and concurrent audio tone). Participants were asked to respond each time a stimulus was presented. The task lasted for 8 minutes and 40 seconds. Magnetoencephalographic data were collected with a 306-channel Elekta Neuromag Vectorview (102 magnetometers and 204 planar gradiometers) at a sampling rate of 1000 Hz (bandpass 0.03-330 Hz). Only planar gradiometers were used in the analysis.

## 3.2 | Processing

No preprocessing was applied except for the removal of excess channels, resampling to 250/350 Hz (for the EEG and MEG sets, respectively), and merging of all conditions per study (EEG study II). First, to test the different elements of the algorithm, we ran it seven times on EEG study II (which contained complex artifacts that differed between the two conditions):

1. Using a fixed removal of 3 components and no chunks.
2. Using a fixed removal, but chunking the data into 150s segments.
3. Using the automatic detector of noise components, but no chunks.
4. Combining chunks and automatic noise component detector.
5. Using chunks with individual peak detection and automatic noise component detector.
6. Using chunks without peak detection and automatic noise component detection with adaptive changes for over- or undercleaning.
7. Using all features (defaults): chunks with individual peak detection, as well as automatic detection with adaptive changes.

All conditions used the automatic detector of noise frequencies. With this approach we tried to mimic the creation of the algorithm with successive improvements.

Subsequently, we ran Zapline-plus additionally on EEG study I and on the MEG studies. For EEG study I we used only default values, for the MEG studies we set 'noisefreqs' to 50 as we expected only line noise and wanted to prevent false positive noise frequency detection due to very strong subharmonics or higher harmonics of the line frequency.

# 4 | RESULTS

The results for analytics (the number of removed components per cleaning step, the ratio of noise/surroundings after cleaning, the proportion of removed power below noise, and the proportion of frequency samples below and above the adaptation threshold) of the cleaning of the successively enabled features while cleaning EEG study II can be found in table 1. The results of the cleaning of all example studies are visualized in figure 4. Only EEG study II had noise frequencies different from line, which is why we specifically show the raw and clean 50 Hz / surroundings power ratios. Overall, the cleaned spectra show that zapline-plus successfully removed the strong line noise peaks while introducing only minimal notches. Table 2 shows the analytics results (the final sigma value for detection, the number of removed components per cleaning step, the ratio of noise/surroundings before and after cleaning, and the

**TABLE 1** Analytics (mean and standard deviation) when using varying features enabled during cleaning of EEG study II. The removed power below noise refers to -11 Hz to -1 Hz below the detected noise frequency, the percentage below/above thresholds refer to the proportion of samples in the spectrum exceeding the thresholds for fine-grained adaptation. Although they were not always used, they are always available for analysis. The values are first averaged over all detected noise frequencies per subject. N = 19.

| | fixed | fixed chunks | auto | auto & chunks |
|---|---|---|---|---|
| final # of removed components | 3 (0) | 3 (0) | 4.55 (1.93) | 5.76 (1.61) |
| clean ratio noise/surroundings | 1.22 (0.21) | 1.09 (0.12) | 1.08 (0.04) | 0.97 (0.07) |
| % removed power below noise | 1.73 (0.57) | 1.87 (0.46) | 3.05 (2.10) | 4.30 (1.36) |
| % below lower threshold | 0 (0) | 3.36 (6.67) | 0.20 (0.63) | 20.08 (16.78) |
| % above upper threshold | 23.13 (11.37) | 16.79 (13.67) | 14.74 (9.94) | 6.82 (7.06) |
| | **auto & chunks & peaks** | **auto & adaptive & chunks** | **full** | |
| final # of removed components | 4.08 (1.09) | 3.63 (0.88) | 3.31 (0.92) | |
| clean ratio noise/surroundings | 1.00 (0.05) | 1.01 (0.05) | 1.03 (0.04) | |
| % removed power below noise | 3.01 (0.95) | 2.41 (0.72) | 2.20 (0.69) | |
| % below lower threshold | 9.31 (12.84) | 5.26 (11.01) | 2.94 (6.51) | |
| % above upper threshold | 7.25 (7.90) | 8.33 (6.85) | 8.37 (7.47) | |

proportion of removed power below noise) of the three example applications.

**Suboptimal case results** Viewing only the average results of the final cleaning, however, yields only a limited understanding of the detailed processes. Some data sets had less-than-ideal results, for example they showed a distortion of the spectrum below noise such that the power was actually increased. This could be seen mostly in data sets with particularly strong noise contamination, especially in MEG study I where four data sets had more than 800 times stronger power at noise frequency than surroundings, up to almost 7000 times for the noisiest data set. All these four data sets, but only them, exhibited a negative removal of power below noise, i.e. an increase, and they drive the average as can be see in figure 4. Also, while all data sets saw a reduction in noise power, some of them had comparably strong residual noise peaks (ratios of noise/surroundings above 1.2, these usually also had very high ratios before cleaning), indicating that Zapline-plus could not fully clean these data sets. In rare cases it happened that the noise/surroundings ratio fell below 1, which means a notch was introduced into the spectrum.

## 5 | DISCUSSION

In this work, we created a wrapper of Zapline to allow fully automatic removal of line noise and other spectral peaks, while giving the user a maximum of flexibility and information, as well as allowing complete replicability of the processing. We evaluated the algorithm on two EEG and two MEG data sets. First, we checked whether the different parts of the algorithm improved the cleaning on one EEG study, then we applied the final default values to the three other datasets. Taken together, the results show that the new features allow for fully automatic noise removal and make the algorithm applicable for different kinds of data, resulting in a substantial decrease of frequency-specific noise with

**TABLE 2** Analytics results of the cleaning of all example data sets (mean and standard deviation). The removed power below noise refers to -11 Hz to -1 Hz below the detected noise frequency. For EEG study II the values are first averaged over all detected noise frequencies per subject, the other studies had only 50 Hz line noise removed.

| | EEG study I (N=24) | EEG study II (N=19) | MEG study I (N=12) | MEG study II (N=23) |
|---|---|---|---|---|
| final sigma of detector | 2.67 (0.22) | 3.33 (0.51) | 3.31 (0.73) | 2.86 (0.34) |
| final # of removed components | 2.65 (1.08) | 3.29 (2.69) | 17.77 (7.95) | 7.88 (1.22) |
| raw ratio noise/surroundings | 6.49 (4.93) | 2.67 (2.08) | 1096.2 (2052.4) | 246.38 (378.27) |
| clean ratio noise/surroundings | 1.1 (0.17) | 1.03 (0.10) | 1.36 (0.59) | 1.03 (0.02) |
| % removed power below noise | 6.20 (3.17) | 2.20 (2.05) | -49.64 (110.26) | 3.95 (1.41) |

minimal negative impact on true neural activity.

**Efficacy of the algorithm** The examination of the algorithm components on EEG study II showed that they do improve the results. However, the improvement is not a simple linear relationship: Both, using chunks, and using automatic detection of to-be-removed components improved the clean ratio of noise/surrounding power similarly over using the standard fixed approach. In doing so, using auto detection affected the power below noise frequencies (-11 to -1 Hz) more than chunks did, but chunks had a larger proportion of samples below the threshold directly at the noise frequency, meaning chunks introduced a slight notch into the spectrum, whereas auto cleaning without chunks distorted the spectrum more generally.

Interestingly, combining these two approaches led to the lowest ratio of noise/surroundings power while also introducing substantial amounts of overcleaning, both in terms of general distortion (% removed below noise) and a notch (% below lower threshold). This combination also had the fewest samples above the adaptation threshold, corresponding to the low noise/surroundings ratio. The strong overcleaning effect can be explained by the fact that not all noise oscillations were present in all chunks. Although the automatic detection of components to remove should be able to select fewer samples with less noise, it requires some sort of 'knee-point' or 'corner' in the artifact scores. In chunks with no oscillation in the given noise frequency, the scores exhibit an almost linear decrease, which can lead to erroneously removing large numbers of components.

This negative interaction effect can be fixed by either adapting the sigma level the detector uses, or by simply not using auto detection when no noise is present. Using either improvement alone led to similar levels of cleaning in terms of noise/surroundings power as well as % of samples above threshold, while the adaptive cleaning had a slightly reduced impact on the spectrum below noise and a reduced notch.

Last, combining all options, chunks with individual peak detection, as well as automatic detection with adaptation, led to the best results overall. This combination had a lower % of samples below the threshold and a lower impact to the spectrum below noise, which is 'paid for' by a slight increase in noise/surroundings power. Overall, the combination of all features of the algorithm successfully cleans the data, while keeping the distortions to the spectrum as low as possible.

Applying this final combination to all example data sets led to substantial improvements of the spectra. In EEG study I, there was 50 Hz line noise present in the data, and an unknown oscillation at around 7 Hz, plus harmonics. The former was detected and successfully cleaned by Zapline-plus, whereas the latter was too small to be detected. EEG study II is a particularly heavily contaminated study, as can be seen by the various peaks in the spectrum. How-

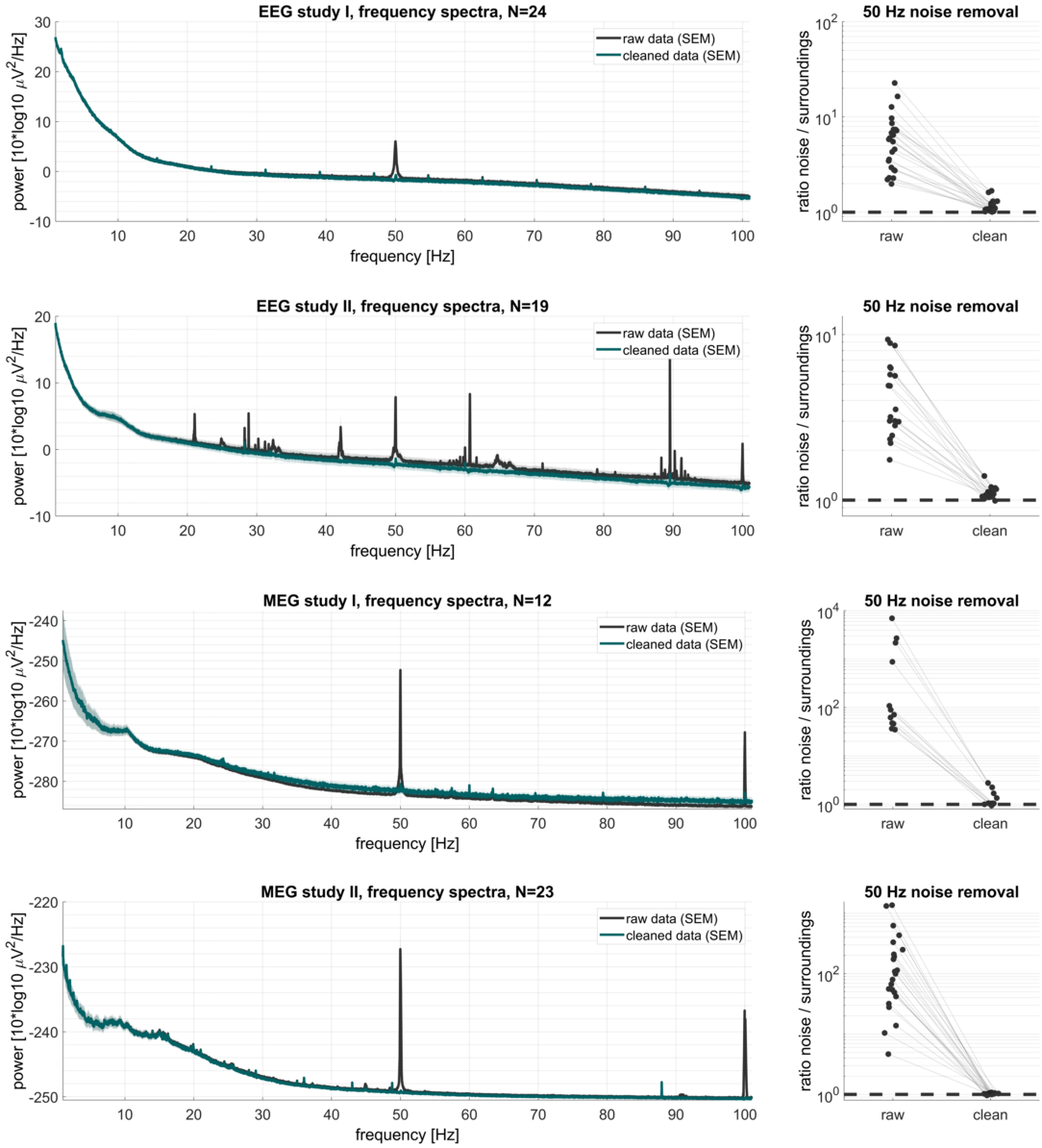


**FIGURE 4** Visualizations of the frequency spectra and the 50 Hz noise removal of the example application data sets. Left: frequency spectra before and after cleaning. Right: ratio of power at noise / surrounding frequency for raw and cleaned data.

ever, Zapline-plus successfully cleans this data, not only at line noise, but also all other strong peaks. This example emphasizes the importance of the automatic noise frequency detector, as these oscillations are difficult to anticipate. Applying Zapline-plus on the MEG studies shows that even extremely noisy data is successfully cleaned. It can be seen in MEG study I, however, that Zapline-plus may have an impact on the overall spectrum by increasing the power.

This effect is driven by four of the twelve data sets, which show extreme levels of noise before cleaning, the other six do not show such an increase. In these cases the actual impact of the cleaning on final measures must be closely examined in order to decide whether the trade-off of reduced noise vs. spectrum distortions is worth it in this particular analysis or if the cleaning must be adapted.

**Other notes** In EEG study II, it was clearly visible that some noise frequencies were only present in the first or second part of the data (body vs joystick rotation, see figure 3 for an example of a noise frequency only present in the second half). This underlines the importance of the chunking and individual frequency detection, as this allows checking whether the oscillation is actually present in that chunk and prevent overcleaning. We would also like to point out the importance of fine-tuned noise frequency detection for some frequencies, especially the one seen in figure 3. The separation of 20.9 Hz and, subsequently, 21.1 Hz noise is important as the two frequencies can not be cleaned together. This would be impossible to see without a high resolution of the frequency spectrum, and simply cleaning with a fixed 21 Hz setting does lead to subpar results. Also, as can be seen in figure 2, the peak frequency of the line noise is not always stationary and Zapline-plus is able to detect these variations.

**Limitations** As we showed, the cleaning is not always perfect. Especially with data that is heavily contaminated with noise, it is possible to 1) change the spectrum below the noise frequency such that the power is actually increased, 2) residual noise is left in the data, or 3) after cleaning, there is a small notch present in the spectrum. Although the default values of the algorithm are chosen to fit most of the data sets, in some cases it might be better to adjust them according to the results obtained from the automatic cleaning and then re-run Zapline-plus. The user is strongly advised to always check the resulting analytics plots after applying Zapline-plus.

**Future directions** It might also be that no matter the parameter adjustment, the cleaning will not work. In these cases it could be useful to combine Zapline-plus with CleanLine, as it was shown that this can improve the cleaning on heavily contaminated data (Miyakoshi et al., 2021). Examining this possibility of an automatic extra CleanLine step if Zapline-plus alone yielded suboptimal results would be an option for future investigations.

Another interesting possibility is to visualize the topographies of the removed artifacts. As Zapline internally uses spatial filters, these can be visualized like any other spatial filter and be added to the analytics information feedback for the user. However, this is not straightforward as Zapline-plus specifically uses different spatial decompositions and different number of removed components for each chunk. Still, if the filters vary only slightly, visualizing the average of the removed topographies could be valuable feedback.

Lastly, it could be explored whether Zapline-plus can also be used for other applications. For example, some of our tests suggest that one could remove very regular mechanical walking artifacts in mobile EEG studies, or the steps could be extracted to create events for subsequent analysis. Another option would be to extract alpha oscillations (8-13 Hz) that exceed the 1/f background activity. This topic has already been mentioned in the original Zapline paper (de Cheveigné, 2020), but with a focus on removing alpha for other analysis. Extracting only the oscillatory alpha time series by switching the "clean" with the "noise" data could result in more specific alpha signals than using a standard band-pass filter. In sum, Zapline-plus is essentially a tool created for noise removal, but it can also be used to extract all kinds of oscillatory activity to be used in other analyses, which makes it a versatile tool in any analysis pipeline.

**Implications for the field** Removing line noise is an undeniably important part of electrophysiological data processing, and having the option to do so without risking the analysis of potentially important frequencies while retaining full data rank is a valuable tool. The newly added features of fully automatic and documented processing including the

detection of noise oscillations are especially important considering the current trend towards complete automatic processing pipelines (Bigdely-Shamlo et al., 2015; Pedroni et al., 2019; da Cruz et al., 2018; Gabard-Durnam et al., 2018) and the need for more rigorous methods in neurophysiological analysis (Cohen, 2017) due to the replication crisis (Open Science Collaboration, 2015). Also, although the impact of preprocessing has been investigated in parts (Robbins et al., 2020), and some pipelines create comprehensive documentation of their processes, a documentation of the line noise removal as detailed as provided by Zapline-plus is lacking thus far. Zapline-plus hence makes a strong contribution to the field by making the removal of line noise and other oscillation artifacts in large data sets automatic, easy, transparent, and reproducible, while limiting its potential negative impact on downstream analysis. It can easily be integrated in any automatic processing pipeline.

## Acknowledgements

We are thankful to the researchers who made the data sets freely available and to Alain de Cheveigné who kindly allowed the adaptation and re-hosting of parts of the original Zapline code. We would also like to thank the members of the Berlin Mobile Brain/Body Imaging Lab of Prof. Klaus Gramann for valuable discussions throughout the development of the algorithm. We acknowledge the support of this work by the DFG (GR2627/8-1) and USAF (ONR 10024807).

## Conflict of interest

The authors declare no conflict of interest.

## Data availability statement

The data used in this study is available for download as laid out in section 3.1. The MATLAB source code of the software is available for download at https://github.com/MariusKlug/zapline-plus.

## References

Bigdely-Shamlo, N., Mullen, T., Kothe, C., Su, K.-m. and Widmann, A. (2015) The PREP pipeline: standardized preprocessing for large-scale EEG analysis. **9**, 1–20.

de Cheveigné, A. (2020) ZapLine: a simple and effective method to remove power line artifacts. *NeuroImage*, **1**, 1–13.

de Cheveigné, A. and Parra, L. C. (2014) Joint decorrelation, a versatile tool for multichannel data analysis. *NeuroImage*, **98**, 487–505.

de Cheveigné, A. and Simon, J. Z. (2007) Denoising based on time-shift PCA. *Journal of Neuroscience Methods*, **165**, 297–305.

Cohen, M. X. (2017) Rigor and replication in time-frequency analyses of cognitive electrophysiology data. *International Journal of Psychophysiology*, **111**, 80–87.

— (2021) *Linear Algebra: Theory, Intuition, Code*. Sincxpress Bv.

da Cruz, J. R., Chicherov, V., Herzog, M. H. and Figueiredo, P. (2018) An automatic pre-processing pipeline for EEG analysis (APP) based on robust statistics. *Clinical Neurophysiology*, **129**, 1427–1437.

Dehais, F., Duprès, A., Blum, S., Drougard, N., Scannella, S., Roy, R. and Lotte, F. (2019) Monitoring Pilot's Mental Workload Using ERPs and Spectral Power with a Six-Dry-Electrode EEG System in Real Flight Conditions. *Sensors*, **19**, 1324.

Dehais, F., Lafont, A., Roy, R. N. and Fairclough, S. H. (2020) A neuroergonomics approach to mental workload, engagement and human performance. *Frontiers in Neuroscience*.

Delorme, A. and Makeig, S. (2004) EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, **134**, 9–21.

Dimigen, O. (2020) Optimizing the ICA-based removal of ocular EEG artifacts from free viewing experiments. *NeuroImage*, **207**, 116117.

Gabard-Durnam, L. J., Leal, A. S., Wilkinson, C. L. and 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**.

Gehrke, L. and Gramann, K. (2021) Single-trial regression of spatial exploration behavior indicates posterior EEG alpha modulation to reflect egocentric coding. *European Journal of Neuroscience*, 1–18.

Gramann, K., Ferris, D. P., Gwin, J. and Makeig, S. (2014) Imaging natural cognition in action. *International Journal of Psychophysiology*, **91**, 22–29.

Gramann, K., Gwin, J. T., Ferris, D. P., Oie, K., Jung, T. P., Lin, C. T., Liao, L. D. and Makeig, S. (2011) Cognition in action: Imaging brain/body dynamics in mobile humans. *Reviews in the Neurosciences*, **22**, 593–608.

Gramann, K., Hohlefeld, F. U., Gehrke, L. and Klug, M. (2021) Human cortical dynamics during full-body heading changes. *Scientific Reports*, **11**, 18186.

Hämäläinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J. and others (1993) Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Rev. Mod. Phys.*

Hoogenboom, N., Schoffelen, J.-M., Oostenveld, R., Parkes, L. M. and Fries, P. (2006) Localizing human visual gamma-band activity in frequency, time and space. *Neuroimage*, **29**, 764–773.

Hyvärinen, A. and Oja, E. (1997) A fast fixed-point algorithm for independent component analysis. *Neural Computation*, **9**, 1483–1492.

Jungnickel, E., Gehrke, L., Klug, M. and Gramann, K. (2018) MoBI-Mobile Brain/Body Imaging. In *Neuroergonomics: The Brain at Work and in Everyday Life* (eds. H. Ayaz and F. Dehais), chap. 10, 59–63. London: Elsevier, 1 edn.

Kloosterman, N. A., de Gee, J. W., Werkle-Bergner, M., Lindenberger, U., Garrett, D. D. and Fahrenfort, J. J. (2019) Humans strategically shift decision bias by flexibly adjusting sensory evidence accumulation. *Elife*, **8**.

Klug, M. and Gramann, K. (2020) Identifying key factors for improving ICA-based decomposition of EEG data in mobile and stationary experiments. *European Journal of Neuroscience*, 1–15.

Liebherr, M., Corcoran, A. W., Alday, P. M., Coussens, S., Bellan, V., Howlett, C. A., Immink, M. A., Kohler, M., Schlesewsky, M. and Bornkessel-Schlesewsky, I. (2021) EEG and behavioral correlates of attentional processing while walking and navigating naturalistic environments. *bioRxiv*.

Makeig, S., Gramann, K., Jung, T. P., Sejnowski, T. J. and Poizner, H. (2009) Linking brain, mind and behavior. *International Journal of Psychophysiology*, **73**, 95–100.

Miyakoshi, M., Schmitt, L. M., Erickson, C. A., Sweeney, J. A. and Pedapati, E. V. (2021) Can We Push the "Quasi-Perfect Artifact Rejection" Even Closer to Perfection? *Frontiers in Neuroinformatics*, **14**, 1–5.

Open Science Collaboration (2015) Estimating the reproducibility of psychological science. *Science*, **349**.

Parasuraman, R. and Rizzo, M. (eds.) (2007) *Neuroergonomics - The brain at work*. New York: Oxford University Press.

Pedroni, A., Bahreini, A. and Langer, N. (2019) Automagic: Standardized preprocessing of big EEG data. *NeuroImage*, **200**, 460–473.

Robbins, K. A., Touryan, J., Mullen, T., Kothe, C. and Bigdely-Shamlo, N. (2020) How Sensitive Are EEG Results to Preprocessing Methods: A Benchmarking Study. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **28**, 1081–1090.

Schoffelen, J. M., Oostenveld, R., Lam, N. H., Uddén, J., Hultén, A. and Hagoort, P. (2019) A 204-subject multimodal neuroimaging dataset to study language processing. *Scientific Data*, **6**, 1–13.

Shafto, M. A., Tyler, L. K., Dixon, M., Taylor, J. R., Rowe, J. B., Cusack, R., Calder, A. J., Marslen-Wilson, W. D., Duncan, J., Dalgleish, T., Henson, R. N., Brayne, C., Matthews, F. E. and Cam-CAN (2014) The cambridge centre for ageing and neuroscience (Cam-CAN) study protocol: a cross-sectional, lifespan, multidisciplinary examination of healthy cognitive ageing. *BMC Neurol.*, **14**, 204.

Taylor, J. R., Williams, N., Cusack, R., Auer, T., Shafto, M. A., Dixon, M., Tyler, L. K., Cam-Can and Henson, R. N. (2017) The cambridge centre for ageing and neuroscience (Cam-CAN) data repository: Structural and functional MRI, MEG, and cognitive data from a cross-sectional adult lifespan sample. *Neuroimage*, **144**, 262–269.

Welch, P. (1967) The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Trans. Audio Electroacoust.*, **15**, 70–73.

Widmann, A., Schröger, E. and Maess, B. (2015) Digital filter design for electrophysiological data – a practical approach. *Journal of Neuroscience Methods*, **250**, 34–46.

Wunderlich, A. and Gramann, K. (2018) Electrocortical evidence for long-term incidental spatial learning through modified navigation instructions. In *Spatial Cognition XI* (eds. S. Creem-Regehr, J. Schöning and A. Klippel), 261–278. Cham: Springer International Publishing.