Behavioural Brain Research 347 (2018) 425–435

Contents lists available at ScienceDirect

Behavioural Brain Research

journal homepage: www.elsevier.com/locate/bbr

STATSLAB: An open-source EEG toolbox for computing single-subject effects using robust statistics

Allan Campopiano[a,b,*], Stefon J.R. van Noordt[b,c,1], Sidney J. Segalowitz[b,2]

[a] Research and Development Services, Halton Catholic District School Board, Burlington, Ontario, Canada
[b] Cognitive and Affective Neuroscience Laboratory, Department of Psychology, Brock University, St. Catharines, Ontario, Canada
[c] Developmental Electrophysiology Laboratory, Yale Child Study Center, Yale University School of Medicine, New Haven, CT, USA

ARTICLE INFO

Keywords:
Robust estimation
Trimmed mean
Percentile bootstrap
EEG

ABSTRACT

Research on robust statistics during the past half century provides concrete evidence that classical hypothesis tests that rely on the sample mean and variance are problematic. Even seemingly minor departures from normality are now known to create major problems in terms of increased error rates and decreased power. Fortunately, numerous robust estimation techniques have been developed that circumvent the need for strict assumptions of normality and equal variances, leading to increased power and accuracy when testing hypotheses. Two robust methods that have been shown to have practical value across a wide range of applied situations are the trimmed mean and percentile bootstrap test. To facilitate the uptake of robust methods into the behavioural sciences, especially when dealing with trial-based data such as EEG, we introduce STATSLAB: An open-source EEG toolbox for computing single-subject effects using robust statistics. With the STATSLAB toolbox users can apply the percentile bootstrap test, with trimmed means, to a variety of neural signals including voltages, global field amplitude, and spectral features for both scalp channels and independent components. The toolbox offers a range of analytical strategies and is packaged with a fully functional graphical user interface that includes documentation.

1. Introduction

The study of robustness in statistics has led to a number of valuable approaches for comparing groups and measuring associations between variables. The motivation for the development of robust methods was to overcome previously underestimated problems with standard approaches that rely on the sample mean and variance [1–4]. Based on a half-century of statistical theory, there is much evidence to suggest that researchers should move towards resampling methods based on robust estimates of central tendency and scale. Compared to conventional approaches developed before the year 1960, which are still used routinely in the behavioral sciences, robust resampling methods offer a substantial advantage in terms of power and accuracy and do not assume normal distributions or equal variances when comparing groups [5]. In the case of electroencephalography (EEG) data, many computationally intensive statistical procedures are no longer very time consuming, making it feasible to analyze multiple channels and time points rather than reducing the data to a peak measured at a single site, as is traditionally done. In this paper, we introduce the STATSLAB software for analyzing EEG with robust estimation techniques. STATSLAB focuses on the trimmed mean and percentile bootstrap test, which are two complementary methods that are valuable for Null Hypothesis Significance Testing [5].

1.1. The trimmed mean

A common goal when testing hypotheses is to accurately characterize a distribution with some measure of central tendency, which can be difficult to achieve if the chosen estimator is not resistant to extreme values. Resistance can be quantified by considering an estimator's finite-sample breakdown point. For example, consider a set of values $X_1, X_2, \cdots, X_n$. Only one of these values would need to be altered in order to render the sample mean meaningless. That is, if the ith value is set to infinity then the sample mean goes to infinity as well [6]. Thus, the finite-sample breakdown point of the sample mean is $1/n$ because a single value can make the sample mean arbitrarily large or small. The median, however, has a finite-sample breakdown point of 0.5 because more than half of the values would have to be made arbitrarily large to

* Corresponding author at: Research and Development Services, Halton Catholic District School Board, Burlington, ON, L7R 2Y2, Canada.
E-mail addresses: campopianoA@hcdsb.org (A. Campopiano), stefonv0@gmail.com (S.J.R. van Noordt), sid.segalowitz@brocku.ca (S.J. Segalowitz).
[1] Yale Child Study Center, Yale University School of Medicine, 230 South Frontage Rd., New Haven, CT, 06520, USA.
[2] Department of Psychology/Jack and Nora Walker Centre for Lifespan Development Research, Brock University, St. Catharines, ON, L2S 3A1, Canada.

https://doi.org/10.1016/j.bbr.2018.03.025
Received 10 January 2018; Received in revised form 10 March 2018; Accepted 15 March 2018
Available online 21 March 2018

drive the median to infinity. One way to achieve a high finite-sample breakdown point is to use the trimmed mean, which involves calculating the sample mean after removing a proportion of values from each tail of the distribution [4]. In symbols the trimmed mean is expressed as

$$\overline{X}_t = \frac{X_{(g+1)} + ,\cdots,+X_{(n-g)}}{n-2g}$$

where X_1, X_2, $\cdots$,X_n is a random sample and $X_{(1)}$, $\leq X_{(2)}$, $\cdots$,$\leq X_{(n)}$ are the observations in ascending order. The proportion to trim is $\gamma\,(0 \leq \gamma \leq .5)$ and $g = [\gamma n]$ rounded down to the nearest integer. The finite-sample breakdown point of the trimmed mean is γ. For example, if $\gamma = .2$, then more than twenty percent of the values would have to be altered to make the 20% trimmed mean arbitrarily large or small. Thus, the trimmed mean is more robust than the sample mean in terms of its finite-sample breakdown point. Other criteria for judging robustness have been proposed and in all cases the sample mean performs poorly compared to any trimmed mean or the median [6,7]. When testing hypotheses, the trimmed mean appears to be a good general choice if trimming is between 10–20%. In addition to a high finite-sample breakdown point, the 10-20% trimmed mean maintains a lower standard error than the sample mean when sampling from heavy-tailed distributions, and this contributes somewhat to smaller confidence intervals and increased statistical power in such cases (see Table 3.7, [8], page 110; [9–11]). It is worth noting that there are no drawbacks to using a robust estimation approach with trimmed means. In the rare event that data are normally distributed, the trimmed mean will estimate the same parameter as the arithmetic mean. In practice, distributions that will generate outliers are common, therefore researchers should consider using robust estimators, such as the trimmed mean, when measuring central tendency.

1.2. The percentile bootstrap test

Null Hypothesis Significance Testing requires that alpha be specified in order to restrict Type I error (false positives) to a nominal rate, usually 5%. In practice this decision amounts to assuming the shape of the sampling distribution in order to determine critical thresholds that correspond to alpha. For example, if the assumed sampling distribution follows a standard normal curve ($\mu = 0$, $\sigma = 1$), then any obtained test statistic more extreme than $\pm$ 1.96 is declared unlikely under the null hypothesis for a two-tailed test. If the null were true, the chances of obtaining a test statistic more extreme than $\pm$ 1.96 would be less than 0.05 and therefore the null would be rejected. Of course, if the sampling distribution is assumed to have a specific shape, but in reality has some other shape, the 95% probability coverage defined by the critical values is inaccurate and Type I error is not kept at the nominal rate. It turns out that inaccurate probability coverage is to be expected when data are sampled from skewed light-tailed distributions, symmetric heavy-tailed distributions, and even when departures from normality are minor [2–4]. To make this clear, an illustration is given using the one-sample *t*-test where the goal is to test the hypothesis that the population mean is equal to some specified constant. Obtaining accurate probability coverage is then based on the assumption that

$$T = \frac{\overline{X} - \mu}{SD/\sqrt{n}}$$

follows a Student's t distribution. If this assumption is true, then reasonable control over Type I error is achieved [12]. The speed of modern computers makes it easy to test whether or not this assumption holds for T. For example, by using a bootstrap method (which is described in more detail below) it is possible to empirically derive an estimate of the T distribution and compare this to the assumed theoretical shape [13]. Fig. 1 compares the assumed and actual sampling distributions of T when sampling is from a lognormal distribution.

As is evident, the assumed probability coverage does not correspond to the 0.025 and 0.975 quantiles of the actual sampling distribution of

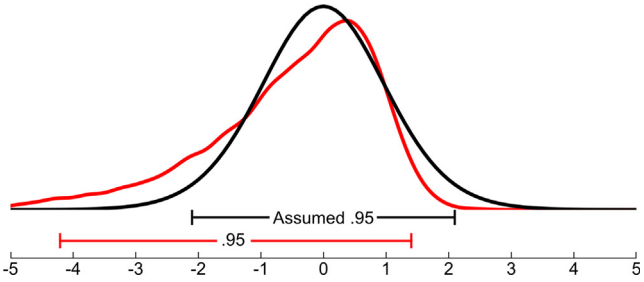

Fig. 1. Assumed (black) and actual (red) probability distributions of T (N = 20, 5000 bootstrap samples) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

T. With $N = 20$, the assumption is that with a probability of 0.95, T will be between -2.09 and 2.09. In reality, however, there is a 0.95 probability that T will be between approximately -4.2 and 1.4 (cf. [8]). In general, violating the assumption that T follows a Student's t distribution manifests itself as increased Type I error when sampling is from skewed light-tailed as well as skewed heavy-tailed distributions. Furthermore, power is markedly reduced when sampling is from symmetric heavy-tailed distributions [12,14].

The problems described above can be minimized by adopting some kind of bootstrap test that utilizes a robust measure of central tendency. Ideally the chosen method should perform well under normality and continue to perform well, compared to other estimators, when sampling is from various non-normal distributions. The percentile bootstrap test and the 20% trimmed mean [9], when combined, satisfy these goals. It has been shown that the percentile bootstrap test with the 20% trimmed mean outperforms Student's *t*-test in terms of maintaining high power when sampling is from heavy-tailed distributions, as well as avoiding increased Type I error when dealing with skewed light-tailed distributions (as do a number of other robust methods not described here) [9]. The percentile bootstrap test based on trimmed means is computed in the following manner: Suppose the goal is to test the null hypothesis that two independent groups have equal trimmed means. In symbols this is specified as

H_0: $\mu_{t1} = \mu_{t2}$

For the jth group (j = 1,2) proceed by

(1) randomly resampling with replacement n_j values from X_{1j}, $\cdots$, X_{nj}, yielding a bootstrap sample

X^*_{1j}, $\cdots$, X^*_{nj}

(2) Compute the difference between the trimmed means of each group based on the bootstrap sample

$D^* = \overline{X}^*_{t1} - \overline{X}^*_{t2}$

(3) Repeat steps 2 and 3 B times yielding the bootstrapped distribution of differences.

(4) Let $D^*_{(1)} \leq ,\cdots, \leq D^*_{(B)}$ be the bootstrapped difference scores in ascending order.

(5) Now determine the quantiles corresponding to alpha. That is, let $l = \alpha B/2$, rounded to nearest integer, and $\mu = B - l$. Thus, a $1-\alpha$ confidence interval for the difference between trimmed means is

$[D^*_{(l+1)}, D^*_{(\mu)}]$

Wilcox [8] describes the extensions to dependent groups as well as extensions to higher-level designs. The process described above can be readily used to compute a *p*-value [15–17].

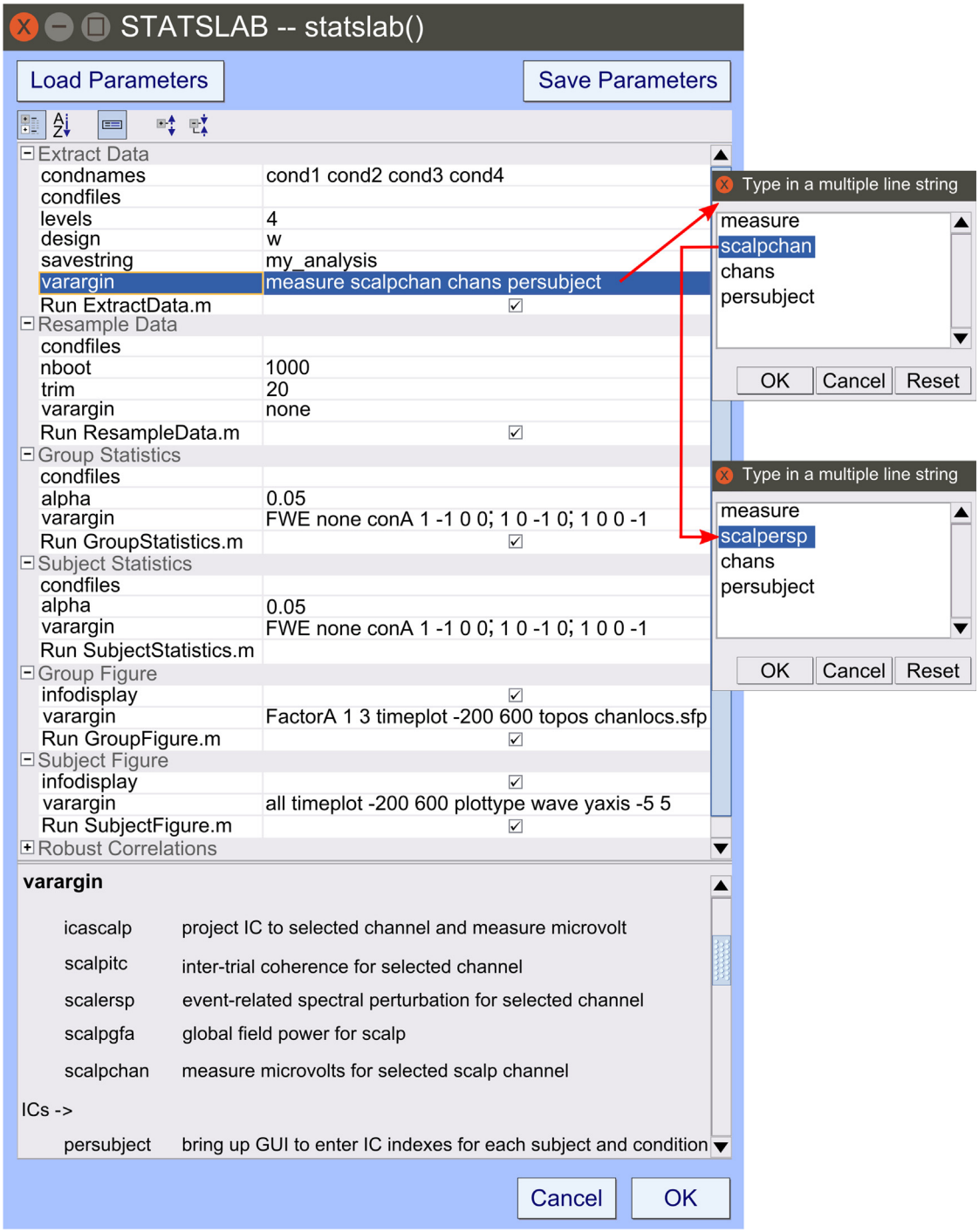

Fig. 2. The STATSLAB GUI. Pop-out windows and drop down lists hold the various parameters. Here the *varargin* pop-out box requires the specification of the dependent variable (*measure*) and the channels to analyze (*chans*). Changing options is done by using different keyword pairs, as demonstrated by pop-out boxes on the right. The documentation pane at the bottom describes the possible parameters.

1.3. Bootstrapping single-trial EEG

In our work we have found it useful to extend the percentile bootstrap method to trial-based data such as EEG. By resampling from a subject's set of single trials it is possible to derive estimates of an individual's sampling distribution of ERP effects [18]. Furthermore, the trimmed mean can be used in place of the sample mean when computing the bootstrapped ERPs in order to minimize the effect of extreme single-trial values at each time point. Applying bootstrapping and the 20% trimmed mean at the single-trial level yields confidence intervals and *p*-values for a single individual ($N = 1$). This procedure has been used in recent EEG studies as a tool to explore individual differences as well as to provide robust estimates of subject-level effects [19–23].

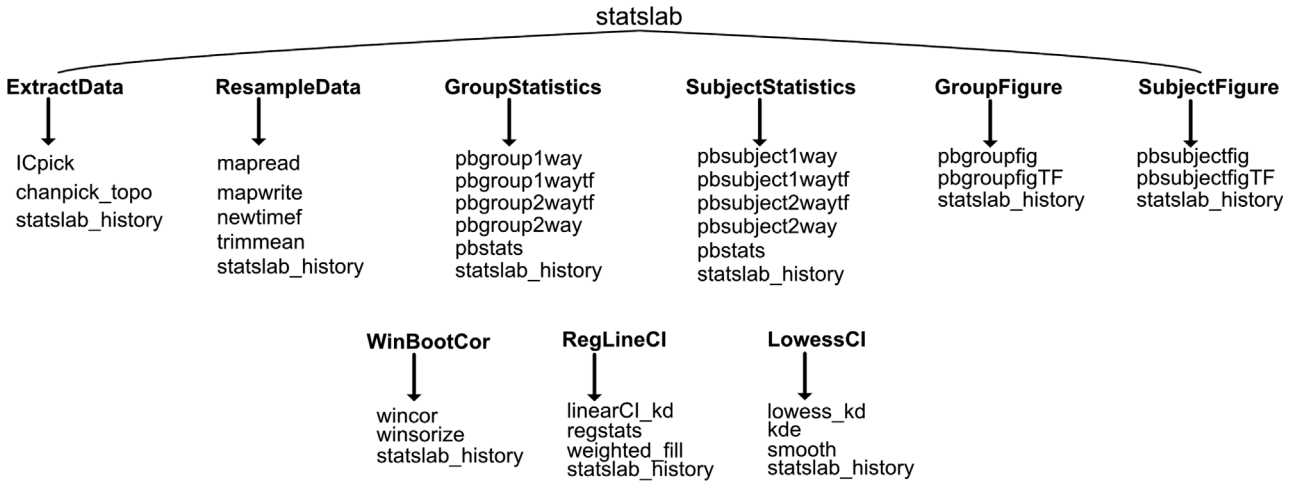

Fig. 3. STATSLAB calling structure. The main function, *statslab*, calls high-level functions which relate to each main module. In turn, the low-level functions are called which relate to the computation of statistics and visualizations, among other operations. The bottom half displays the calling structure of the functions related to correlations and tests of independence.

An animation of the percentile bootstrap procedure for a single subject can be found at https://github.com/Alcampopiano/STATSLAB. The animation provides a clear summary of the bootstrapping procedure. In the following paragraphs we also give a more formal description. Note that matrix notation is used as it is more convenient given the multi-dimensional nature of EEG data.[3]

In the case of EEG, one can test the null hypothesis at many time points and spatial locations. For example, suppose there are two experimental conditions and the goal is to test the null hypothesis that, for all time points in the waveform and at a single location on the scalp, a subject has equal trimmed means in each condition. In symbols this can expressed as

$$H_0: \boldsymbol{\mu}_{t1} = \boldsymbol{\mu}_{t2} = [\mu_{t11} = \mu_{t21},\ \mu_{t12} = \mu_{t22},\ \cdots,\ \mu_{t1m} = \mu_{t2m}]$$

where $\boldsymbol{\mu}_{t1}$ and $\boldsymbol{\mu}_{t2}$ are 1 x m row vectors. That is, we are conducting the hypothesis for all m time points in the waveform.

Consider the following matrix which represents a single subject's data from one electrode in one condition:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & ,\cdots, & a_{1m} \\ a_{21} & & & a_{2m} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & ,\cdots, & a_{nm} \end{bmatrix}$$

where n is equal the number of trials and m is equal to the number of time points. To test the single-subject hypothesis given above with the percentile bootstrap method, proceed in the following way:

(1) Obtain a new n x m matrix by randomly resampling with replacement n rows from matrix $\boldsymbol{A}$ yielding

$$\boldsymbol{A}^* = \begin{bmatrix} a^*_{11} & a^*_{12} & ,\cdots, & a^*_{1m} \\ a^*_{21} & & & a^*_{2m} \\ \vdots & & & \vdots \\ a^*_{n1} & a^*_{n2} & ,\cdots, & a^*_{nm} \end{bmatrix}$$

Note that the asterisk notation is used to indicate that elements in $\boldsymbol{A}^*$ are based on resampling from the original data in $\boldsymbol{A}$.

(2) $\forall_i\ \varepsilon\{1, \cdots, m\}$ let

$$\boldsymbol{p}^*_{ti} = \frac{a^*_{(g+1)i} + ,\cdots,+a^*_{(n-g)i}}{n-2g},$$

where $\forall_i\ \varepsilon\{1, \cdots,m\}$, $a_{(1)i} \leq a_{(2)i}, \cdots, \leq a_{(n)i}$ represents the elements in $\boldsymbol{A}^*$ sorted in ascending order along the first dimension. As stated earlier, the proportion to trim is $\gamma\,(0 \leq \gamma \leq .5)$ and $g = [\gamma n]$ rounded down to the nearest integer. $\boldsymbol{p}^*_t$ is therefore a 1 x m row vector of trimmed means.

(3) Repeat steps 1–3 until B trimmed ERPs are generated for the first condition yielding $\boldsymbol{P}^*_{t1}$.

(4) Repeat steps 1–4 for a second condition, eventually yielding $\boldsymbol{P}^*_{t2}$.

(5) Compute a B x m matrix of difference scores around which CIs can be computed:

$$\boldsymbol{P}^*_{t1} - \boldsymbol{P}^*_{t2} = \boldsymbol{D}^* = \begin{bmatrix} d^*_{11} & d^*_{12} & ,\cdots, & d^*_{1m} \\ d^*_{21} & & & d^*_{2m} \\ \vdots & & & \vdots \\ d^*_{B1} & d^*_{B2} & ,\cdots, & d^*_{Bm} \end{bmatrix}$$

(6) Determine the quantiles corresponding to alpha in the usual way. That is, let $l = \alpha B/2$, rounded to nearest integer, and $\mu = B - l$. $\forall_i\ \varepsilon\{1, \cdots, m\}$, a $1-\alpha$ confidence interval for the difference between trimmed means is given by

$$[d^*_{(l+1)i},\ d^*_{(\mu)i}]$$

where $\forall_i\ \varepsilon\{1, \cdots, m\}$, $d^*_{(1)i} \leq d^*_{(2)i}, \cdots, \leq d^*_{(n)i}$ represents the values in $\boldsymbol{D}^*$sorted in ascending order along the first dimension.

The single-subject statistics can be summarized into group-level results by extending the method just described. Letting $\boldsymbol{P}^*_{t1i}$ represent the bootstrapped waveforms for the ith subject in the first condition, compute

$$\boldsymbol{G}_1 = \frac{\boldsymbol{P}^*_{t11} + \boldsymbol{P}^*_{t12} + ,\cdots,+\boldsymbol{P}^*_{t1N}}{N}$$

where N is equal to the number of subjects. This is repeated for the second condition, yielding $\boldsymbol{G}_2$, and by proceeding along similar lines as in steps 5, compute

$$\boldsymbol{G}_1 - \boldsymbol{G}_2 = \boldsymbol{D}$$

where $\boldsymbol{D}$ is a B x m matrix of difference scores based on the grand averages. Finally, determine the CIs in the usual way by sorting the elements of $\boldsymbol{D}$ along the first dimension and computing the quantiles that correspond to alpha for each time point in the waveform.

[3] This paper follows the convention of representing a matrix with an uppercase letter in boldface font and its elements as lowercase letters with row and column subscripts. A matrix with a single row or column is referred to as a vector and represented by a lowercase letter in boldface font. The elements of a vector are represented by lowercase letters with a single subscript. On occasion, subscripts and superscripts are used to denote additional details (e.g., experimental condition).

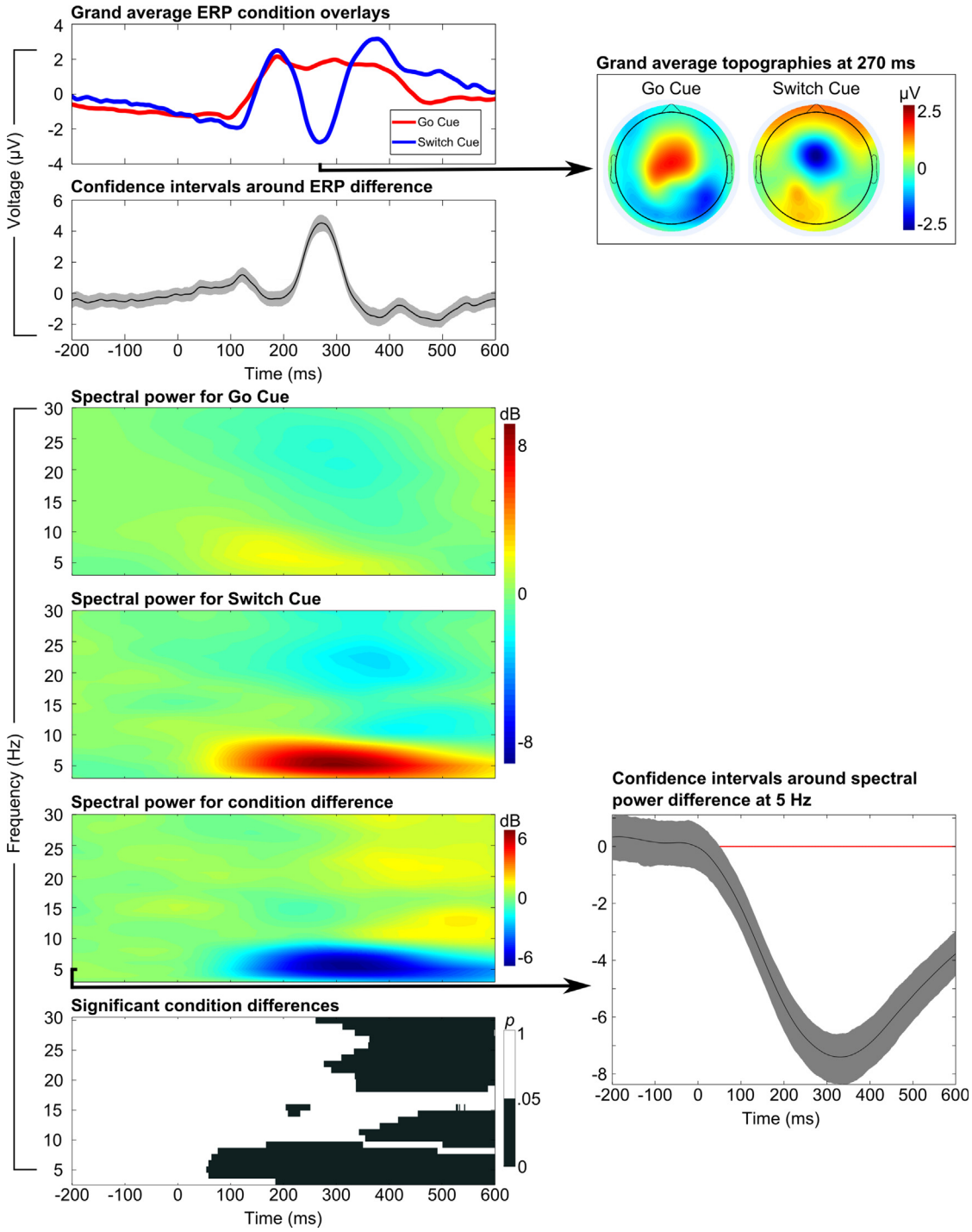

Fig. 4. Group-level waveforms from two conditions (red and blue) and the corresponding difference wave with bootstrapped CI (top left). Topographies generated from clicking waveforms at desired latency (top right). ERSP for each condition and the difference. Black areas in the bottom subplot denote statistical differences as a function of time and frequency band. The CI is generated by clicking on the ERSP condition difference plot at the desired frequency (bottom right) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

2. STATSLAB

Here we introduce STATSLAB, an open-source MATLAB toolbox for computing the percentile bootstrap test based on trimmed means, at the group and single-subject level. The software is primarily designed to analyze EEG/MEG, and depends on functions contained in the EEGLAB toolbox [24]. STATSLAB provides an option-rich environment for robust hypothesis testing and data visualizations, and supports common hierarchal analyses for dependent, independent, and mixed designs. Operations are specified within a graphical user interface (GUI) and, for more advanced users, the command line interface. The GUI contains built-in documentation so that users can easily access instructions and examples. STATSLAB may be used to compute statistics for all latencies and channels in both the time and time-frequency domains. Furthermore, users who have previously run Independent Component Analysis (ICA) may test hypotheses on latent factors representing cortical source projections. The STATSLAB software, tutorial, and a sample data set can be downloaded from

https://github.com/Alcampopiano/STATSLAB. The following sections describe generally how the software is used to test hypotheses and visualize results.

2.1. The STATSLAB GUI

Typing *statslab* into the MATLAB command line interface will bring up the GUI. The GUI stores all of the parameters needed for computing each stage of an analysis (see Fig. 2). Parameters are inputted by the user and provide information such as the alpha level, condition labels, and the type of experimental design type that is being used, as well as many other details that relate to how an analysis is carried out and how the results should be displayed. As the parameters are populated they can be saved and subsequently loaded. This saves time as previously used parameters often require only minimal alterations to be compatible with a new analysis. Fig. 2 shows an example of a fully populated GUI, but highlights the single alteration necessary to switch the dependent measure from microvolts to Event-Related Spectral Perturbation (ERSP).

The interface is separated into modules including *Extract Data*, *Resample Data*, *Group Statistics*, *Subject Statistics*, *Group Figure*, and *Subject Figure*. Each module is associated with a function by the same name, which in turn makes calls to many lower-level functions (see Fig. 3). The standard pipeline consists of running the modules sequentially, beginning at *Extract Data,* and continuing down until *Group Figure* and *Subject Figure* are completed. All figures produced in STATSLAB can be exported and saved as scalable vector graphics (SVG) so that users can customize them for publication format outside of MATLAB. Additional modules located at the bottom of the GUI compute robust measures of association including Winsorized correlation and LOWESS (Locally Weighted Scatterplot Smoothing), as well as a heteroscedastic test of independence (described below).

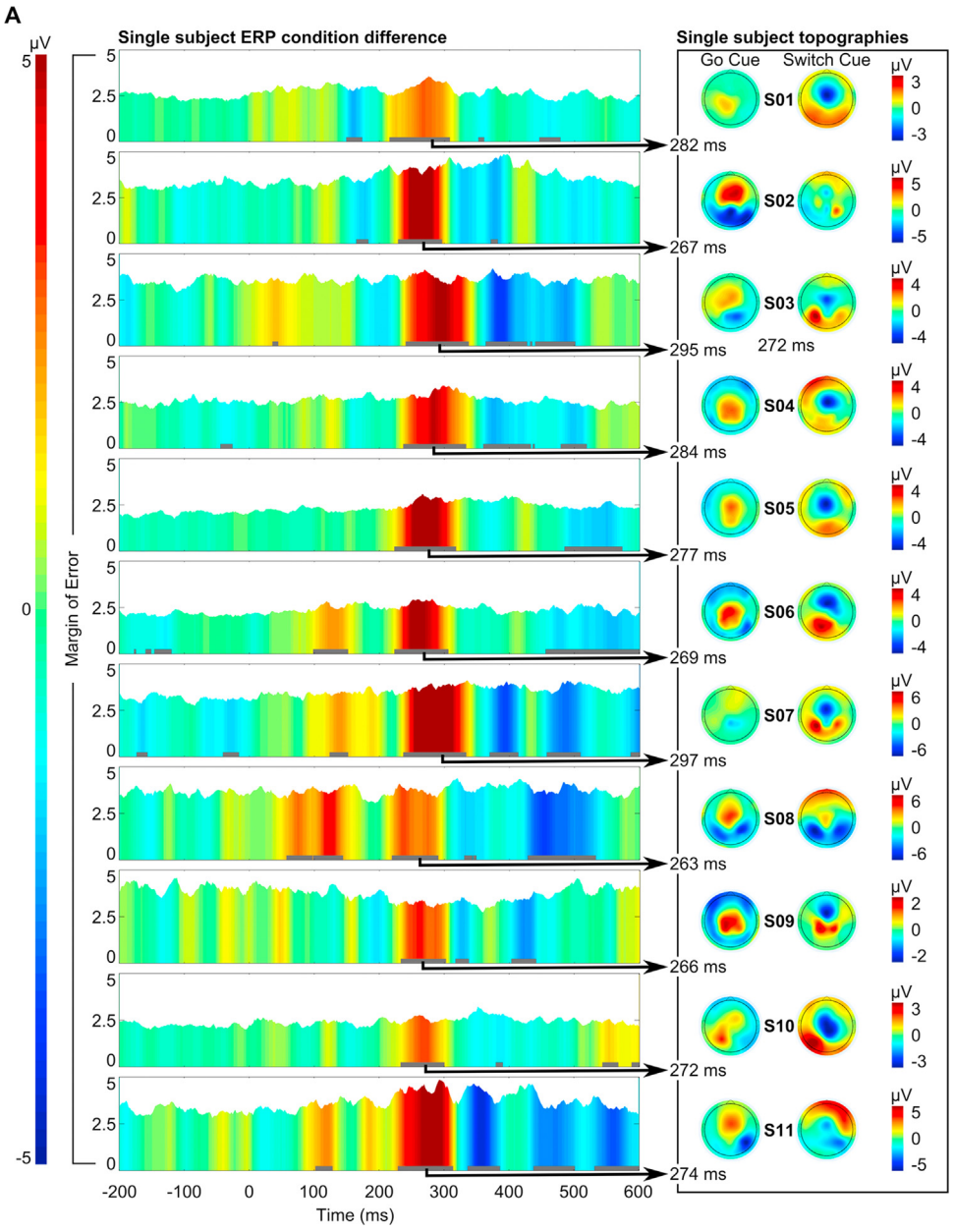

Fig. 5. A. Difference waves (as a colorbar) for each subject. The height of the colorbar represents the margin-of-error (one half of the CI). The topography for each condition and subject is generated by clicking the colorbars. Topographies are scaled to each subject's minimum and maximum values. B. ERSP plots identical to those in Fig. 4, however bottom subplot shows the proportion of subjects who show a statistical effect at a given time and frequency. The CI for each subject is generated by clicking on the ERSP condition difference plot at the desired frequency (right).

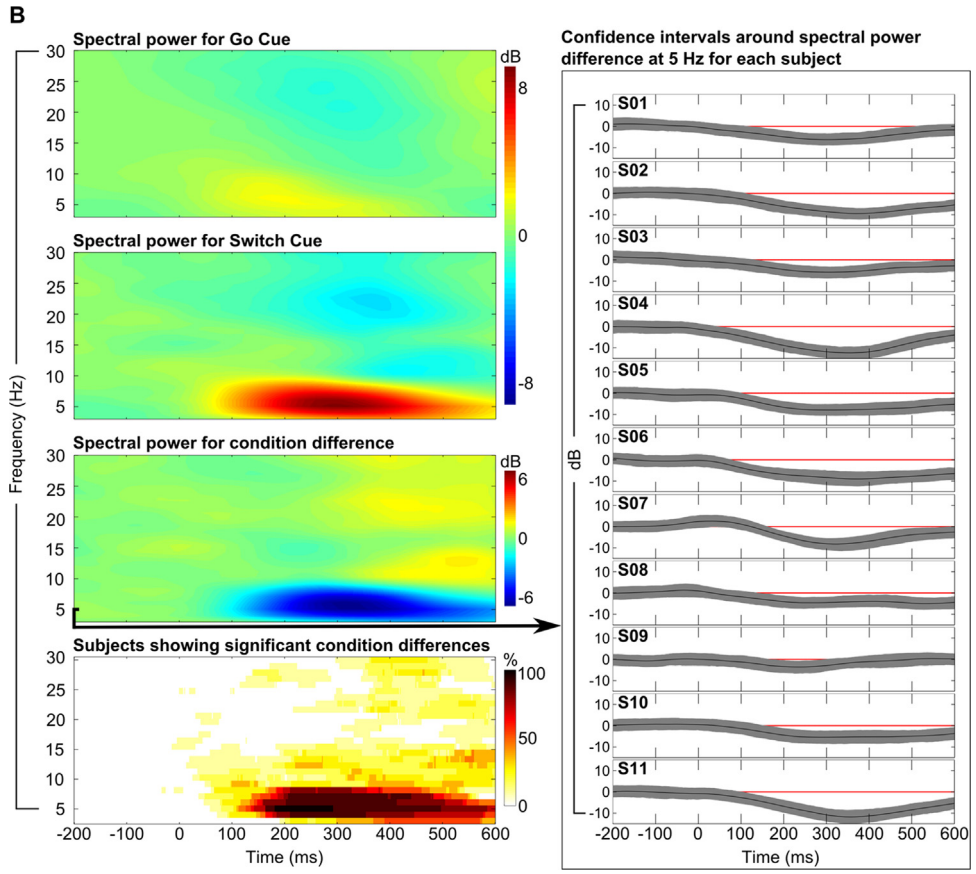

Fig. 5. (*continued*)

Clicking on the various fields in the GUI will reveal the corresponding documentation in the bottom pane which explains the required information. At this point it is convenient to briefly explain the purpose of each module.

2.1.1. Extract data

The purpose of the *Extract Data* module is to extract from the EEGLAB file only those data that are necessary for the current analysis. For example, if the analysis is being conducted on only one channel, then *Extract Data* will extract the specified channel from all subjects' EEGLAB files, producing a number of smaller output files in the native MATLAB format (i.e., *.mat). In addition, *Extract Data* collects parameters such as condition names, design specifications (e.g., within- or between-subjects), and the dependent measure (e.g., global field power, microvolts, time-frequency measures). All of these options are specified by the user in plain text or via drop-down lists.

The *varargin* field specifies the channels to extract as well as the dependent measure to use. If the user prefers to graphically choose channels for each subject, *varargin* options can be given which bring up a topographical channel-selection interface. Furthermore, *varargin* allows the user to specify whether or not hypothesis testing is to be computed on Independent Components (ICs) or scalp-level EEG. For example, the *varargin* options specified in leftmost pop-out box in Fig. 2 indicate the following information: (1) channels are to be selected graphically on a per-subject basis (indicated by the *persubject* keyword), and (2) the dependent variable is measured in microvolts at the scalp (indicated by the *scalpchan* keyword). The documentation pane lists all other available options and provides many examples.

2.1.2. Resample data

The Resample Data module takes the output of Extract Data and computes the bootstrap samples (sometimes referred to as bootstrap surrogates). Specifically, each subject's extracted data (files ending in *extracted.mat*) are bootstrapped by randomly resampling from the trials, calculating the trimmed ERP, and storing the result. This process is repeated *nboot* times (e.g., 1000) for each input file. Here, one can also specify the amount of trimming that is used when calculating the trimmed mean. The output files from *Resample Data* contain the *nboot* surrogates based on the dependent variable specified in the *Extract Data* module. For example, if the chosen dependent measure was in the time-frequency domain (specified via the *scalpersp*, *scalpitc*, *icaersp,* or *icaitc* keywords), then the output files from *Resample Data* would contain the bootstrapped time-frequency decomposition, which is three dimensional (frequency by time by bootstrapped surrogates). In all other cases the output files will be two dimensional (bootstrap surrogates by time).

2.1.3. Group and subject statistics

The statistics modules take the output from *Resample Data* (files ending in *bootstrapped.mat* or *bootstrapped.map*) in order to compute *p*-values, confidence intervals (CIs), and test statistics using low-level functions (e.g., *pbstats.m*). The user must specify the alpha level and a method for controlling family-wise error. Further, the *varargin* field requires contrast coefficients in order to control how conditions (or groups) are to be compared via the *con1way.m* or *con2way.m* functions. For readability when using the GUI, the contrast coefficients in STATSLAB are entered on a single line delimited by a semicolon. Formally, however, contrast coefficients are arranged as matrices where each column in the matrix corresponds to a particular comparison and the number of rows corresponds to the number of conditions in the design (this arrangement is necessary for the matrix multiplication that happens in the background). To demonstrate, consider a 2×2 design where the goal is to compute simple effects for each factor as well as the interaction. The following set of contrast matrices accomplish this goal [5]. Note that for Factor A, conditions one and three may be compared,

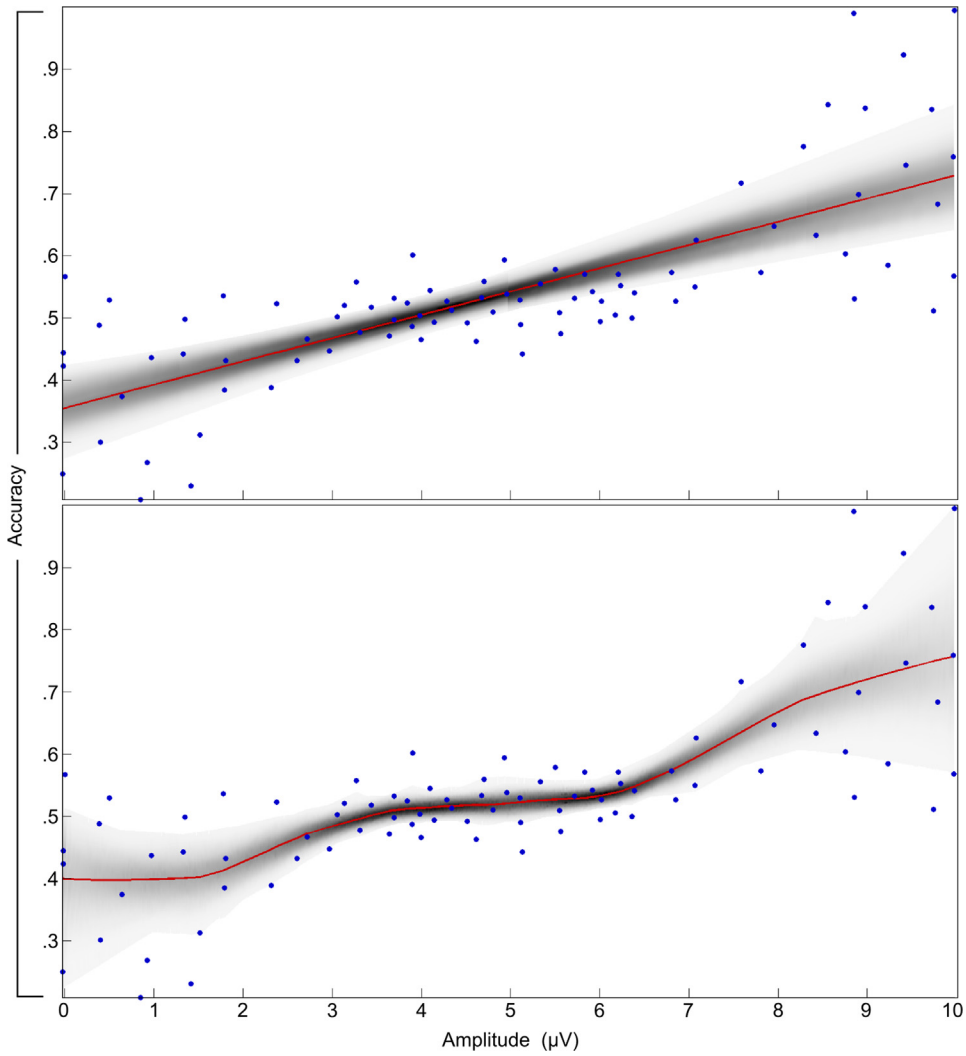

Fig. 6. Bootstrapped confidence bands around the regression line (top) and Cleveland's (1979) smoother (bottom). The shading of the confidence bands are visually weighted, meaning that they are darkest when they contain a high density of bootstrap surrogates [31].

followed by a comparison between conditions two and four, like so:

$$\begin{matrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{matrix}$$

For Factor B, conditions one and two may be compared, followed by a comparison between conditions three and four:

$$\begin{matrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{matrix}$$

Finally, the interaction is given by:

$$\begin{matrix} 1 \\ -1 \\ -1 \\ 1 \end{matrix}$$

In Fig. 2, the contrasts given by the parameters in the GUI indicate that the first condition (with the label *cond1*) is to be compared to all other conditions. Given in formal notation, the contrast matrix appears as follows:

$$\begin{matrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{matrix}$$

2.1.4. Group and subject figures

The figure plotting modules specify options relating to categorical visualizations. In the *varargin* field, users can specify which results to visualize based on the contrasts that were set in the *Group* and *Subject Statistics* modules. For example, the parameters in Fig. 2 specify that only the first and third contrast should be visualized when the *Group Figure* module is run (via the *FactorA* keyword). Various aesthetic options can be specified such as the window length (in milliseconds), axis range, and the style of graph. When computing statistics in the time domain, the *Subject Figure* module allows users to choose from the following three graph styles (via the *plottype* keyword): standard waveforms, horizontal colorbars, and colorbars which include information about the margin-of-error (i.e., the width of one half of the CI). There is also an option to compute topographies. If this option is selected, topographies are generated by clicking on the waveforms at a particular latency of interest. In the case of *Subject Figure*, topographies are produced for each subject. Figs. 4 and 5 show examples of group- and subject-level visualizations, respectively.[4]

2.1.5. Measures of association

STATSLAB includes a module for computing and visualizing robust measures of association between external variables and (1) the condition waveform in the design, or (2) the difference waves given by the

[4] The data used to produce the figures are from van Noordt et al. [22].

chosen contrasts. The robust correlation module computes Pearson's r using the Winsorized marginal distributions. Winsorizing is similar to trimming with the exception being that, instead of removing data, the extreme values are set to the nearest untrimmed value. Given a random sample that is sorted in ascending order, $X_{(1)} \leq X_{(2)}, \cdots, \leq X_{(n)}$, Winsorizing is expressed in symbols as

$$W_i = \begin{cases} X_{(g+1)}, \text{ if } X_i \leq X_{(g+1)} \\ X_i, \text{ if } X_{(g+1)} < X_i < X_{(n-g)}, \\ X_{(n-g)}, \text{ if } X_i \geq X_{(n-g)} \end{cases}$$

where the proportion to Winsorize is $y\ (0 \leq y \leq .5)$ and $g = [yn]$ rounded down to the nearest integer. Pearson's r is not resistant because it has a finite sample breakdown point of $1/n$. When Winsorizing the marginal distributions prior to computing Pearson's r, a higher finite sample breakdown point of y is achieved [8].

Although resistant to extreme values in the marginal distributions, proceeding in the usual way by computing the classic test of independence is still problematic when data are heteroscedastic. For example, if the goal is to test

H_0: $\rho = 0$

that is, that the population correlation is zero, the classic test of independence is given by

$$T = r\sqrt{\frac{n-2}{1-r^2}}.$$

When data are heteroscedastic, however, type I error rates are much higher than assumed when using this method, even in situations when the population correlation is zero [8]. In other words, the reason for rejecting the null may be unclear to us when relying on the classic test of independence if heteroscedasticity is present. The percentile bootstrap method described above provides a heteroscedastic method for testing independence. For example, when using Winsorized correlation, a heteroscedastic test of independence can be computed as follows:

(1) Given the bivariate data, $(Y_{11}, Y_{12}), \cdots, (Y_{n1}, Y_{n2})$, generate a bootstrap sample by randomly resampling with replacement n pairs yielding $(Y_{11}^*, Y_{12}^*), \cdots, (Y_{n1}^*, Y_{n2}^*)$
(2) Winsorize each marginal distribution by applying the formula above to each variable separately and compute

$$r_w = \frac{\Sigma(Y_{i1}-\overline{Y}_1)(Y_{i2}-\overline{Y}_2)}{\sqrt{\Sigma(Y_{i1}-\overline{Y}_1)^2(Y_{i2}-\overline{Y}_2)^2}}$$

(3) Label the result r_w^*, and repeat steps 1 and 2 B times
(4) Sort the B values in ascending order and let $l = \alpha * B/2$ rounded to nearest integer, and $\mu = B-l$
(5) A $1-\alpha$ confidence interval for r_w is:

$$[r_{w(l+1)}^*, r_{w(u)}^*]$$

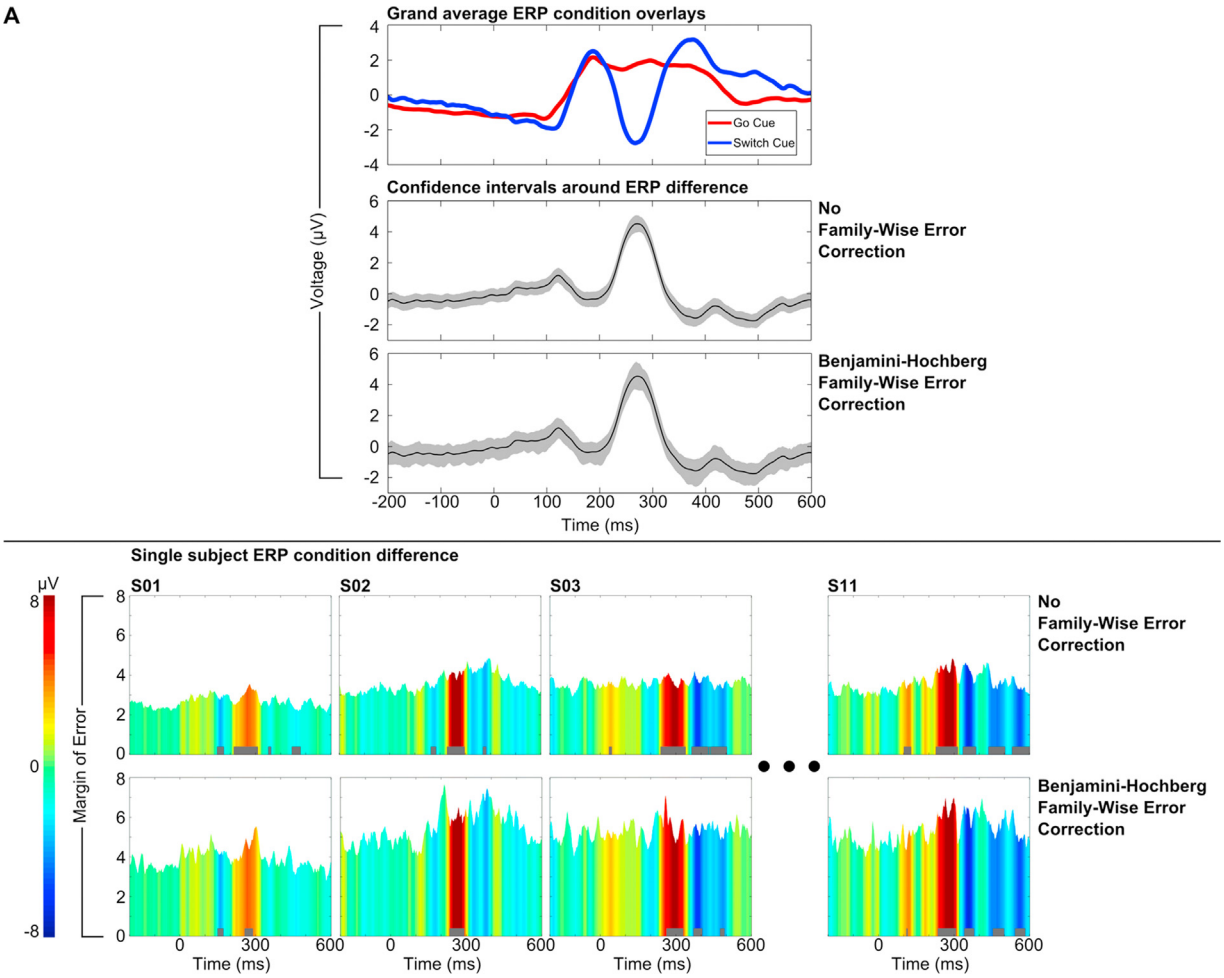

Fig. 7. A.Top panel shows group-level waveforms from two conditions (red and blue) with uncorrected (middle) and corrected (bottom) CIs. Bottom panel shows uncorrected (top) and corrected (bottom) margin of error figures for a sample of single subjects. B. Top panel shows ERSP plots identical to those in Figs. 4 and 5B. Bottom panel shows masked significant condition differences (left) and proportion of subjects who show a statistical effect at a given time and frequency (right), when data are uncorrected (top) and corrected (bottom). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

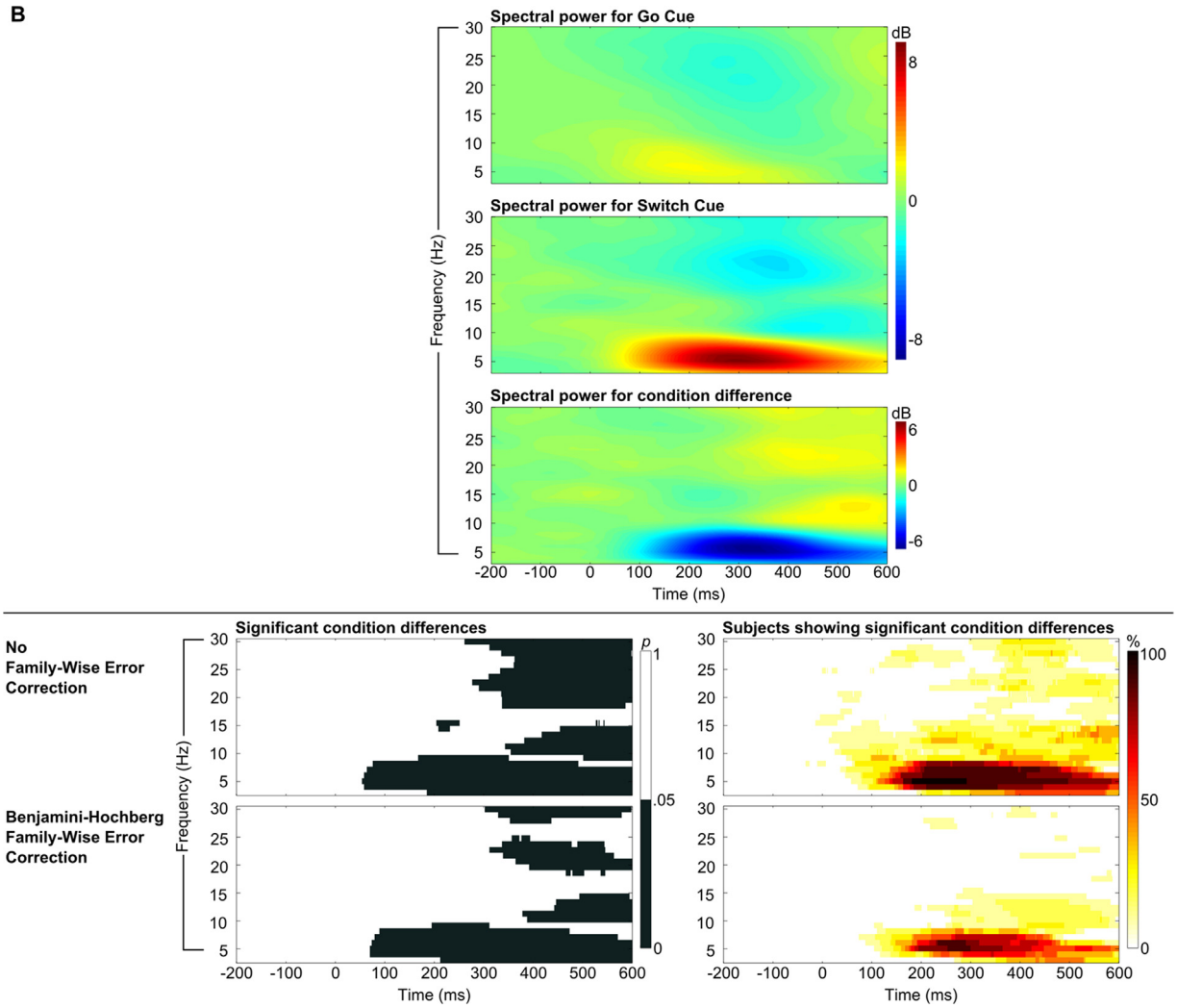

Fig. 7. (*continued*)

As described in the documentation pane in STATSLAB, once the required parameters are entered, the Winsorized correlation and bootstrapped test of independence are computed for all latencies. In addition to visualizing the r_w values, CIs, and *p*-values as a function of time, the *Measures of Association* module allows users to generate scatterplots along with bootstrapped confidence bands around the typical regression line. When dealing with curvature, Locally Weighted Scatterplot Smoothing (LOWESS; Cleveland, 1979) may also be visualized. The usual bootstrap principles apply when characterizing a cloud of points on a scatterplot. Proceed by resampling with replacement pairs of points as was done for the heteroscedastic test of independence above and, with each resample, compute the regression line or any other measure of interest. STATSLAB uses the resulting bootstrapped distribution to compute confidence bands around the parameter estimate. Fig. 6 shows examples of the output from the *Measures of Association* module.

2.1.6. Multiple comparison and error rate correction

To deal with the issue of multiple comparisons across the time course of an effect, STATSLAB uses the Benjamini-Hochberg FDR correction [25], as described in Groppe et al. [26], which corrects alpha by considering all time-points in the family of tests. In STATSLAB, we consider a family of tests to be any set of contrasts belonging to the *FactorA*, *FactorB*, or *FactorAB* options. Thus, if two tests are specified under the *FactorA* option, p-values for both contrasts are concatenated and corrected as a whole with the Benjamini-Hochberg method. This FDR correction is available for all dependent measures offered in STASTLAB. See Fig. 7 for examples of dependent measures that have been calculated with and without FDR correction applied.

2.1.7. Robust estimation toolboxes

There are several powerful open-source toolboxes available for analyzing and computing robust statistics for EEG data, which may be more appropriate for users depending on the nature of their data and goals of the research. The ERP PCA Toolkit is one such MATLAB toolbox that offers a variety of preprocessing options (e.g., filtering, data quality metrics) as well as the ability to compute robust ANOVAs based on Principal Components Analysis [27]. These methods are gaining attention, and guidelines regarding best practices for their use in EEG have been documented, including assessing p-value stability across bootstrap re-samples [28]. With more complex multifactorial designs, users can utilize the open-source LIMO EEG MATLAB toolbox [29]. LIMO implements a hierarchical linear modeling approach that allows users to include a wide range of categorical and/or continuous variables, including single trial data and user-specified covariates (e.g., noise). After setting up the appropriate ANCOVA design matrix, users can perform robust statistics and implement a variety of state-of-the-art correction procedures for multiple comparisons. The ERP PCA toolkit and LIMO are far more fully-featured and mature than STATSLAB and we encourage users to add them to their repertoire of statistical

software. This is especially the case for larger and more complex designs (e.g., ANCOVA, 3-factor ANOVAS), as well as when other types of bootstrapping tests and analytic procedures are required or preferred (e.g., the Welch-James ADF, robust ANOVAs and ANCOVAs, cluster-based correction for multiple comparisons). Our software differs in that users can compute robust statistics on a variety of dependent signals (e.g., EEG microvolts, GFA, ITC, ERSP, and signals derived from scalp or independent component analysis) for *both* groups and single-subjects. In addition, STATSLAB allows robust single-subject statistics to be compared to individual difference measures via winsorized correlation and LOWESS. Currently, our toolbox is limited to 2-factor designs and deals exclusively with pair-wise comparisons of main effects and interactions.

3. Conclusions

In this paper, we have outlined some of the advantages to using robust estimation techniques, including the trimmed mean and percentile bootstrap test, with a particular focus on single-subject statistics. Robust methods overcome the well-documented limitations with classical approaches and offer a multitude of alternative strategies to data analysis that are easily computed by modern day computers. Given that these methods have been around for decades and rest on a solid mathematical foundation, we encourage applied researchers in the neurosciences, and behavioural science more generally, to incorporate these techniques as a part of their research repertoire. Robust methods are certainly not a panacea and we are not suggesting that they be used in place of all other methods. Our perspective is that if robust methods perform well when data are sampled from a range of probability distributions, and continue to perform well when distributions are normal, then from a mathematical standpoint there is nothing to lose by switching to modern strategies. There is already a great lag between what statisticians and many behavioural scientists consider to be evidence-based analytic strategies. The classical set of statistical tools are based on developments prior to the year 1960 and since then many examples of how these methods can fail have been documented. These failures have given way to the development of robust methods and these should be incorporated into what is considered standard practice in the behavioural sciences. To this end, we have developed STATSLAB, which is a MATLAB toolbox for computing the percentile bootstrap test using trimmed means on EEG/MEG data. With the development of STATSLAB, and other toolboxes that focus on robust statistics [27,29,30], it is our hope that researchers will start to implement robust estimation techniques to probe the neural correlates of behaviour at both the group and single-subject level. These techniques have a wealth of statistical and theoretical advantages that could lead to a more nuanced understanding of brain function and the development of individual profiles in experimental, clinical, and treatment domains.

Acknowledgements

Funding was provided by grants to SJS from the Natural Sciences and Engineering Research Council of Canada (Grant number: 122222-2013) and the Canadian Foundation for Innovation (Grant number: 8780), and Postdoctoral Fellowship from the Natural Sciences and Engineering Research Council of Canada to SJRV.

References

[1] F.R. Hampel, Contributions to the Theory of Robust Estimation. Unpublished PhD Thesis, University of California, Berkeley, 1968.
[2] P.J. Huber, Robust estimation of a location parameter, Annal. Math. Stat. 35 (1) (1964) 73–101.
[3] P.J. Huber, John W. Tukey's contributions to robust statistics, Ann. Stat. 30 (6) (2002) 1640–1648.
[4] J.W. Tukey, A survey of sampling from contaminated distributions, Contrib. Probab. Stat. Essays Honor Harold Hotelling 2 (1962).
[5] R. Wilcox, Introduction to Robust Estimation and Hypothesis Testing Introduction to Robust Estimation and Hypothesis Testing, 4th edition, Academic Press, San Diego, CA, 2017.
[6] P.J. Huber, E. Ronchetti, Robust Statistics, Wiley, 2009.
[7] R.R. Wilcox, Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy, Springer, 2010.
[8] R. Wilcox, Introduction to Robust Estimation and Hypothesis Testing Introduction to Robust Estimation and Hypothesis Testing, 3rd edition, Academic Press, San Diego, CA, 2012.
[9] R.R. Wilcox, Pairwise comparisons of trimmed means for two or more groups, Psychometrika 66 (3) (2001) 343–356.
[10] M. Hill, W.J. Dixon, Robustness in real life: a study of clinical laboratory data, Biometrics (1982) 377–396.
[11] P.C. Wu, Central Limit Theorem and Comparing Means, Trimmed Means One-step M-estimators and Modified One-step M–estimators Under non-normality (Doctoral dissertation), University of Southern California, 2002.
[12] R.R. Wilcox, How many discoveries have been lost by ignoring modern statistical methods? Am. Psychol. 53 (3) (1998) 300–314.
[13] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall/CRC, 1993.
[14] R.R. Wilcox, H.J. Keselman, Modern robust data analysis methods: measures of central tendency, Psychol. Methods 8 (3) (2003) 254–274.
[15] P. Hall, On symmetric bootstrap confidence intervals, J. R. Stat. Soc. 50 (1) (1988) 35–45.
[16] P. Hall, Theoretical comparison of bootstrap confidence intervals, Annal. Stat. 16 (3) (1988) 927–953.
[17] R. Liu, K. Singh, Notions of limiting P values based on data depth and bootstrap, J. Am. Stat. Assoc. 92 (437) (1997) 266–277.
[18] I. Oruç, O. Krigolson, K. Dalrymple, L.S. Nagamatsu, T.C. Handy, J.J.S. Barton, Bootstrap analysis of the single subject with event related potentials, Cognit. Neuropsychol. 28 (5) (2011) 322–337.
[19] J. Desjardins, S. Segalowitz, Deconstructing the early visual electrocortical responses to face and house stimuli, J. Vis. 13 (2013) 1–18.
[20] G.A. Rousselet, J.S. Husk, P.J. Bennett, A.B. Sekuler, Time course and robustness of ERP object and face differences, J. Vis. 47 (27) (2007) 3350–3359.
[21] G.A. Rousselet, C.R. Pernet, Quantifying the time course of visual object processing using ERPs: it's time to up the game, Front. Psychol. 2 (2011) 1–6.
[22] S.J.R. van Noordt, J.A. Desjardins, S.J. Segalowitz, Watch out! medial frontal cortex is activated by cues signaling potential changes in response demands, NeuroImage 114 (2015) 356–370.
[23] S.J.R. van Noordt, A. Campopiano, S.J. Segalowitz, A functional classification of medial frontal negativity ERPs: Theta oscillations and single subject effects, Psychophysiology 53 (9) (2016).
[24] A. Delorme, S. Makeig, EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis, J. Neurosci. Methods 134 (1) (2004) 9–21.
[25] Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing, J. R. Stat. Soc. 57 (1) (1995) 289–300.
[26] D.M. Groppe, T.P. Urbach, M. Kutas, Mass univariate analysis of event-related brain potentials/fields I: a critical tutorial review, Psychophysiology 48 (2011) 1726–1737.
[27] J. Dien, The ERP PCA Toolkit: an open source program for advanced statistical analysis of event-related potential data, J. Neurosci. Methods 187 (1) (2010) 138–145.
[28] J. Dien, Best practices for repeated measures ANOVAs of ERP data: reference, regional channels, and robust ANOVAs, Int. J. Psychophysiol. 111 (2017) 42–56.
[29] C.R. Pernet, N. Chauveau, C. Gaspar, G.A. Rousselet, LIMO EEG: a toolbox for hierarchical LInear MOdeling of ElectroEncephaloGraphic data, Comput. Intell. Neurosci. 2011 (2011) 831409.
[30] R. Wilcox (2016). Retrieved from https://github.com/nicebread/WRS.
[31] S.M. Hsiang, Visually-Weighted Regression, Available at SSRN: (2013) https://ssrn.com/abstract=2265501.