

UC San Diego

UC San Diego Previously Published Works

Title

From BIDS-Formatted EEG Data to Sensor-Space Group Results: A Fully Reproducible Workflow With EEGLAB and LIMO EEG

Permalink

<https://escholarship.org/uc/item/31h2c83v>

Authors

Pernet, Cyril R

Martinez-Cancino, Ramon

Truong, Dung

et al.

Publication Date

2021

DOI

10.3389/fnins.2020.610388

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed



From BIDS-Formatted EEG Data to Sensor-Space Group Results: A Fully Reproducible Workflow With EEGLAB and LIMO EEG

Cyril R. Pernet^{1*}, Ramon Martinez-Cancino², Dung Truong², Scott Makeig² and Arnaud Delorme^{2,3,4}

¹ Centre for Clinical Brain Sciences, The University of Edinburgh, Edinburgh, United Kingdom, ² Swartz Center for Computational Neurosciences, University of California, San Diego, San Diego, CA, United States, ³ Centre de Recherche Cerveau et Cognition, Université Toulouse III – Paul Sabatier, Toulouse, France, ⁴ Centre National de la Recherche Scientifique, Centre de Recherche Cerveau et Cognition, Toulouse, France

OPEN ACCESS

Edited by:

Dezhong Yao,
University of Electronic Science
and Technology of China, China

Reviewed by:

Tibor Auer,
University of Surrey, United Kingdom
Annalisa Pascarella,
Istituto per le Applicazioni del Calcolo
“Mauro Picone” (IAC), Italy

*Correspondence:

Cyril R. Pernet
cyril.pernet@ed.ac.uk

Specialty section:

This article was submitted to
Brain Imaging Methods,
a section of the journal
Frontiers in Neuroscience

Received: 25 September 2020

Accepted: 11 December 2020

Published: 11 January 2021

Citation:

Pernet CR, Martinez-Cancino R, Truong D, Makeig S and Delorme A (2021) From BIDS-Formatted EEG Data to Sensor-Space Group Results: A Fully Reproducible Workflow With EEGLAB and LIMO EEG. *Front. Neurosci.* 14:610388. doi: 10.3389/fnins.2020.610388

Reproducibility is a cornerstone of scientific communication without which one cannot build upon each other's work. Because modern human brain imaging relies on many integrated steps with a variety of possible algorithms, it has, however, become impossible to report every detail of a data processing workflow. In response to this analytical complexity, community recommendations are to share data analysis pipelines (scripts that implement workflows). Here we show that this can easily be done using EEGLAB and tools built around it. BIDS tools allow importing all the necessary information and create a study from electroencephalography (EEG)-Brain Imaging Data Structure compliant data. From there preprocessing can be carried out in only a few steps using EEGLAB and statistical analyses performed using the LIMO EEG plug-in. Using Wakeman and Henson (2015) face dataset, we illustrate how to prepare data and build different statistical models, a standard factorial design (faces * repetition), and a more modern trial-based regression approach for the stimulus repetition effect, all in a few reproducible command lines.

Keywords: brain imaging data structure, preprocessing algorithm, linear models, reproducibility and tools, EEGLAB toolbox, LIMO EEG

INTRODUCTION

As data analyses become more and more complex, it has been advocated that clear workflows and all of the parameters used in their implementation should be reported in order to increase reproducibility (Pernet et al., 2020). It is, however, difficult to concisely report a workflow and maybe even impossible given hidden parameters built into the algorithms we use. One solution is to report in detail the workflow and share the corresponding pipelines—thus only having to communicate key algorithm details. Such pipelines and/or tools to build pipelines have been developed in recent years (see, e.g., Bigdely-Shamlo et al., 2015; Andersen, 2018; Jas et al., 2018; Niso et al., 2019; Meunier et al., 2020) and here we describe tools developed around EEGLAB (Delorme and Makeig, 2004) which also allow creating a fully reproducible pipeline from raw data to group results, with an example to sensor space analysis.

New data formatting conventions and public repositories for electroencephalography (EEG) data have recently been developed and made available to the science community. In particular,

the Brain Imaging Data Structure (Gorgolewski et al., 2016) and its EEG extension (Pernet et al., 2019) allow defining important EEG metadata information, such as additional event information, electrode positions, and experimental conditions. This makes data aggregation from different experiments and analysis automation using standardized pipelines easier. Here we present a fully reproducible workflow (**Figure 1**) from raw data to group results using open data and we document and share the pipeline.

EEGLAB (Delorme and Makeig, 2004) is the most commonly used platform for EEG data analysis (Hanke and Halchenko, 2011; Martínez-Cancino et al., 2020) and all steps proposed can also be reproduced from the user interface. We refer to the extensive EEGLAB online user manual for GUI operations, and simply point that functions called by interface operations in EEGLAB are saved into the EEG.history field, thus allowing to copy/paste the underlying code to build a different pipeline than the one proposed here. EEGLAB is used here in conjunction with newly developed EEG-BIDS¹ tools which allow automatically importing and create a STUDY (see **Figure 1**) for data that follow the EEG-Brain Imaging Data Structure and with the LIMO-EEG toolbox which allows statistical analyses using robust hierarchical linear models (Pernet et al., 2011). The overarching goal of this analysis is to show that it is possible to use a simple pipeline to process “raw” data and perform complex statistical analyses using EEGLAB and LIMO.

METHODS

Open Data

The pipeline was executed using the EEG data from the multimodal face dataset (Wakeman and Henson, 2015). EEG (70 channel EasyCap²) and ECG data were extracted from the binary MEG.fif files that combined MEG, EEG, ECG channels, event markers were time corrected (−34 ms) and electrode positions re-oriented to fit the head coordinate system. Out of the 19 participants, participant 1 was removed because of channels digitization errors leading to 18 participants. Data were then organized using EEG-BIDS and archived for download at <https://openneuro.org/datasets/ds002718>. This dataset is thus a modified and simplified version of the original dataset, with a subset of this original dataset itself made available by authors at <https://openneuro.org/datasets/ds000117>. In the original data, there were multiple runs for each subject, so we have merged them to make it simpler for users. We have also formatted scanned electrode positions so they are available within the BIDS dataset. We have corrected event latencies, renamed some events, and added information on events and stimulus repetition. We also resampled the data to 250 Hz so it is not as large and can be used for tutorials. All modifications to the original data documented in the readme file of BIDS ds002718 and conversion script are also made available.

¹<https://github.com/sccn/bids-matlab-tools>

²<https://www.easy-cap.de/>

The experiment consisted of the presentation of 300 grayscale photographs of familiar and unfamiliar faces, along with their scrambled versions, all repeated twice. Trials started by a fixation cross lasting 400–600 ms, followed by stimuli lasting from 800 to 1000 ms and the repetition of the images occurred either immediately or after 5 or 15 stimuli, leading to a range of time intervals between repeats. Participants had to perform an orthogonal symmetry judgment task ensuring attention to each stimulus.

Software

Analyses are performed using Matlab 2020a (The Mathworks, Inc.³) on Windows or Mac OSX with the Statistical and Machine Learning Toolbox installed, along with EEGLAB⁴ (v2020.0) and its BIDS tool⁵ (v3.5) and LIMO EEG⁶ (v3) plugins—both of them available through the EEGLAB plugin manager.

BIDS-Import

The EEGLAB BIDS plugin allows importing BIDS datasets as EEGLAB studies. The plugin allows overwriting events and channel information contained in the raw EEG data—we are using both of these options here. This allows, for example, to define more precise events and channel information—such as channel locations derived from scanned Polhemus positions. The plugin also allows selecting specific fields in the BIDS event file (in this case we used “trial_type” to be mapped to the EEGLAB “type” field). This is reflected in the call to *pop_importbids.m* function in **Figure 2**.

Fully Automated Preprocessing

We present here a workflow with minimal preprocessing (**Figure 2**), which is a set of steps that removes common artifacts without trying to optimize any particular data features, using fully automated methods. Note that, as mentioned earlier, the goal is to present a fully reproducible workflow of ERP analyses with EEGLAB, and not to reproduce or replicate Wakeman and Henson (2015). As such, our preprocessing differs from Wakeman and Henson’s workflow which, in any case, was primarily designed for the associated MEG data, starting with Signal Space Separation for noise removal (Taulu and Simola, 2006), then bad channel removal, and notch filtering.

As explained above, the code underlying the different steps can be obtained from EEG.history when using the GUI to process subjects. First, bad channels are removed and data filtered at 0.5 Hz using *clean_rawdata* plugin of EEGLAB (v2.2) and the *pop_clean_rawdata.m* function (transition band [0.25 0.75]), bad channels defined as a channel exhibiting a flat line of at least 5 s and/or correlation to their robust estimate based on other channels below 0.8). Second, data are re-referenced to the average (*pop_reref.m*) and submitted to an independent component analysis (*pop_runica.m* using the runica algorithm and a reduction in rank to the number of channels −1 to account

³<https://uk.mathworks.com/products/matlab.html>

⁴<https://sccn.ucsd.edu/eeglab>

⁵<https://github.com/sccn/bids-matlab-tools>

⁶https://github.com/LIMO-EEG-Toolbox/limo_tools

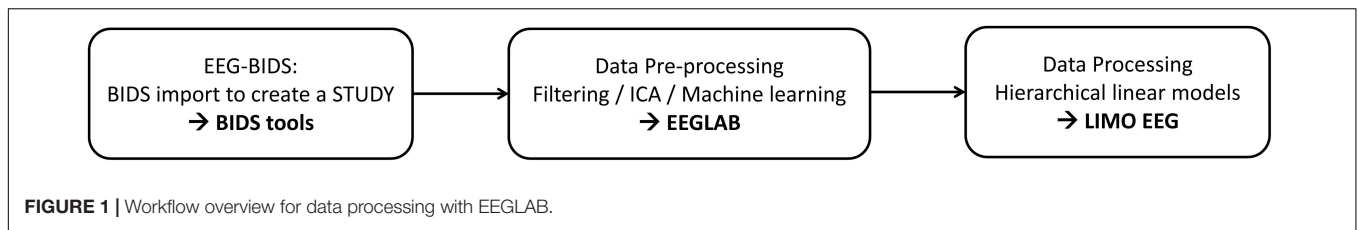


FIGURE 1 | Workflow overview for data processing with EEGLAB.

```

% start EEGLAB
clear; [ALLEEG, EEG, CURRENTSET, ALLCOM] = eeglab;

% call BIDS tool BIDS
studypath = 'XXX\ds002718';
[STUDY, ALLEEG] = pop_importbids(filepath, 'bidsevent', 'on', 'bidschanloc', 'on', ...
    'studyName', 'Face_detection', 'outputdir', fullfile(filepath, 'derivatives'), ...
    'eventtype', 'trial_type');
ALLEEG = pop_select( ALLEEG, 'nochannel', {'EEG061', 'EEG062', 'EEG063', 'EEG064'});
CURRENTSTUDY = 1;
EEG = ALLEEG;
CURRENTSET = 1:length(EEG);

% Remove bad channels
EEG = pop_clean_rawdata( EEG, 'FlatlineCriterion', 5, 'ChannelCriterion', 0.8, ...
    'LineNoiseCriterion', 4, 'Highpass', [0.25 0.75], ...
    'BurstCriterion', 'off', 'WindowCriterion', 'off', 'BurstRejection', 'off', ...
    'Distance', 'Euclidian', 'WindowCriterionTolerances', 'off' );

% Rereference using average reference
EEG = pop_reref( EEG, [], 'interpchan', []);

% Run ICA and flag artifactual components using IClabell
for s=1:size(EEG,2)
    EEG(s) = pop_runica(EEG(s), 'icatype', 'runica', 'concatcond', 'on', 'options', {'pca', EEG(s).nbchan-1});
    EEG(s) = pop_iclabel(EEG(s), 'default');
    EEG(s) = pop_icflag(EEG(s), [NaN NaN; 0.8 1; 0.8 1; NaN NaN; NaN NaN; NaN NaN; NaN NaN]);
    EEG(s) = pop_subcomp(EEG(s), find(EEG(s).reject.gcompreject), 0);
end

% clear data using ASR - just the bad epochs
EEG = pop_clean_rawdata( EEG, 'FlatlineCriterion', 'off', 'ChannelCriterion', 'off', ...
    'LineNoiseCriterion', 'off', 'Highpass', 'off', 'BurstCriterion', 20, ...
    'WindowCriterion', 0.25, 'BurstRejection', 'on', 'Distance', 'Euclidian', ...
    'WindowCriterionTolerances', [-Inf 7] );

% Extract data epochs (no baseline removed)
EEG = pop_epoch( EEG, {'famous_new', 'famous_second_early', 'famous_second_late', ...
    'scrambled_new', 'scrambled_second_early', 'scrambled_second_late', 'unfamiliar_new', ...
    'unfamiliar_second_early', 'unfamiliar_second_late'}, [-0.5 1], 'epochinfo', 'yes');
EEG = eeg_checkset(EEG);
EEG = pop_saveset(EEG, 'savemode', 'resave');
ALLEEG = EEG;

% update study & compute single trials
STUDY = std_checkset(STUDY, ALLEEG);
[STUDY, EEG] = std_precomp(STUDY, EEG, {}, 'savetrials', 'on', 'interp', 'on', 'recompute', 'on', ...
    'erp', 'on', 'erpparams', {'rmbase' [-200 0]}, 'spec', 'off', 'ersp', 'off', 'itc', 'off');
eeglab redraw
  
```

Import EEG-BIDS data
remove unwanted channels
create a STUDY

Remove bad channels
Filtering
Re-referencing

ICA decomposition
Component labelling
Artefact rejection

Extract epochs
Computes single trials

FIGURE 2 | Code used for importing and preprocessing. The region highlighted in red must be changed by the user to point to the location of the data on his/her hard drive. A version of this code that can be copied and pasted is available in the LIMO MEEG repository: https://github.com/LIMO-EEG-Toolbox/limo_meeg/tree/master/resources/from_bids2stats.m. Note that this part of code was automatically generated from EEGLAB and EEGLAB plugins menus by using the command history function `eegh-` and then edited for clarity.

for average reference). Third, each component is automatically labeled using ICLabel (Pion-Tonachini et al., 2019), rejecting components labeled as eye movements and muscle activity above 80% probability. Finally, continuous data are further cleaned if their power deviated too much from the rest of the data using Artifact Subspace Reconstruction (ASR) algorithm and a 20 burst detection criteria threshold (Kothe and Jung, 2016; Chang et al., 2018) thus taking care of remaining (e.g., line noise, trends) artifacts (*pop_clean_rawdata.m*, burst criterion 20). Note that the data are not corrected by ASR as we only use here the portion of the algorithm detecting bad portions of data and remove them. Also, there is no low-pass filtering since power-line noise is removed via ASR, leaving possible higher frequencies in the otherwise low frequency of visual the ERPs. All parameters are indicated in **Figure 3**.

Statistical Modeling

From the clean continuous data, epochs are created by extracting data snippets time-locked (from -500 ms to 1 s) to the face presentation events (*pop_epoch.m*) and designs created within the EEGLAB STUDY. A STUDY in EEGLAB is a structure

that contains all the information about the data and metadata allowing to create any experimental designs. Here, we made two designs (*std_makedesigns.m*): one that recreates the faces vs. scrambled effect described by Wakeman and Henson (2015) and one that tests if the time between repetitions of the same stimuli influences the event-related potentials (ERPs—see section “Results” for more details on the statistical models). From first level estimates, repeated measures ANOVA were computed at the group level (*limo_random_select.m*). The goal is to illustrate the flexibility of linear models as implemented in LIMO EEG and ease to create such models using EEGLAB STUDY (**Figure 3**).

RESULTS

Model 1: Coding Conditions Across Trials

At the first level, data from each participant were modeled with nine experimental conditions: famous faces, famous faces repeated immediately, famous faces repeated late, scrambled faces, scrambled faces repeated immediately, scrambled faces repeated late, unfamiliar faces, unfamiliar faces repeated

```
chanlocs = [STUDY.filepath filesep 'limo_gp_level_chanlocs.mat'];

%% two-way ANOVA faces * repetition

% 1st level analysis
STUDY = std_makedesign(STUDY, ALLEEG, 1, 'name','FaceRepetition','delfiles','off','defaultdesign','off',...
    'variable1','type','values1',{ 'famous_new','famous_second_early','famous_second_late','scrambled_new',...
    'scrambled_second_early','scrambled_second_late','unfamiliar_new','unfamiliar_second_early','unfamiliar_second_late'},...
    'vartype1','categorical','subselect',{ 'sub-002','sub-003','sub-004','sub-005','sub-006','sub-007','sub-008',...
    'sub-009','sub-010','sub-011','sub-012','sub-013','sub-014','sub-015','sub-016','sub-017','sub-018','sub-019'});
[STUDY, EEG] = pop_savestudy( STUDY, EEG, 'savemode','resave');
STUDY = pop_limo(STUDY, ALLEEG, 'method','WLS','measure','daterp','timelim],[-50 650],'erase','on','splitreg','off','interaction','off');

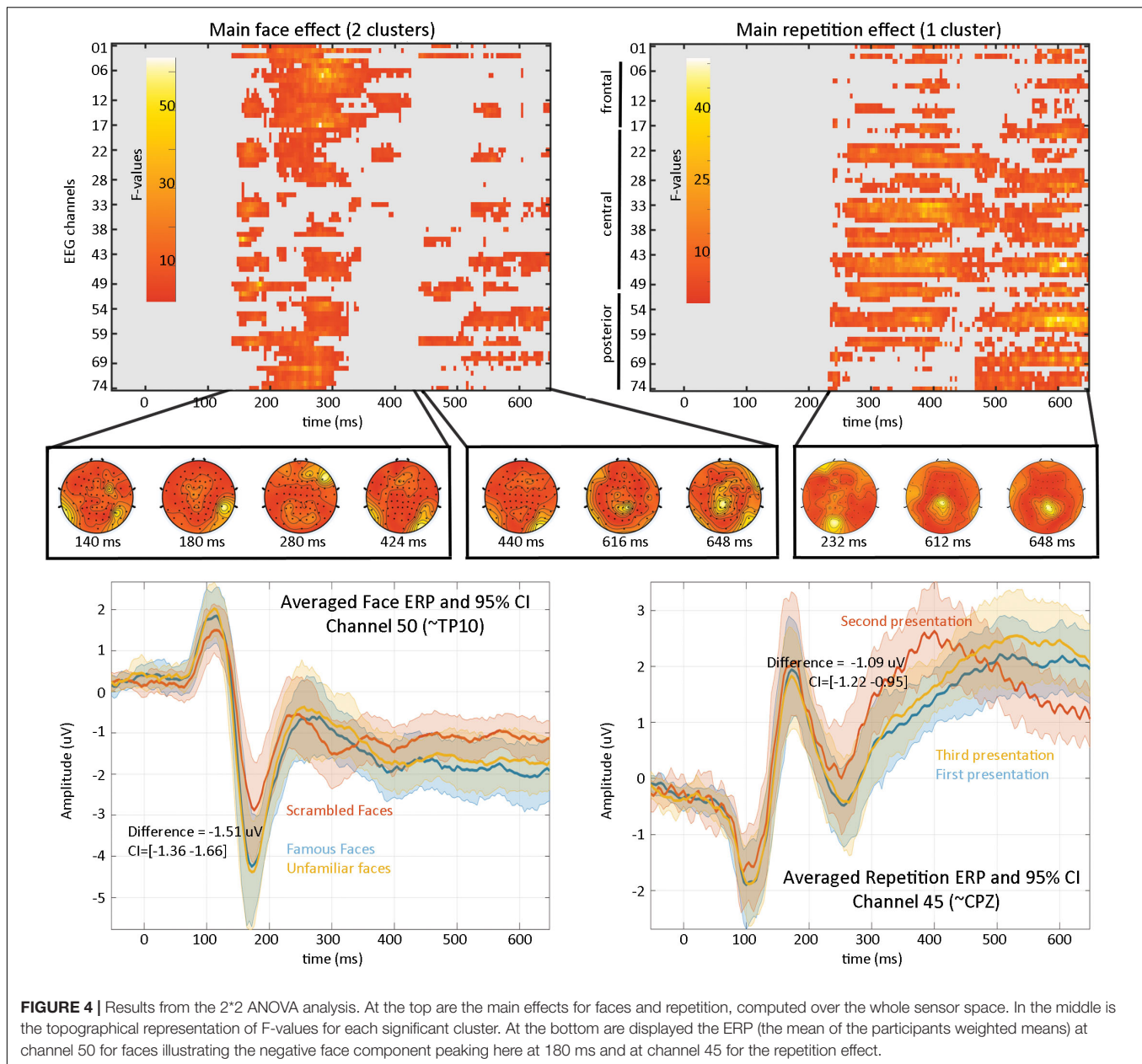
% 2nd level analysis
mkdir([STUDY.filepath filesep 'derivatives' filesep '2-ways-ANOVA'])
cd([STUDY.filepath filesep 'derivatives' filesep '2-ways-ANOVA'])
limo_random_select('Repeated Measures ANOVA',chanlocs,'LIMOfiles',...
    {[STUDY.filepath filesep 'LIMO_Face_detection' filesep 'Beta_files_FaceRepetition_GLM_Channels_Time_WLS.txt']},...
    'analysis_type','Full scalp analysis','parameters',{[1 2 3],[4 5 6],[7 8 9]},...
    'factor names',{'face','repetition'},'type','Channels','nboot',1000,'tfce',0,'skip design check','yes');

%% one-way ANOVA on time distance between trials for each face condition

% 1st level analysis
STUDY = std_makedesign(STUDY, ALLEEG, 2, 'name','Face_time','delfiles','off','defaultdesign','off',...
    'variable1','face_type','values1',{ 'famous','scrambled','unfamiliar'},'vartype1','categorical',...
    'variable2','time_dist','values2',{},'vartype2','continuous',...
    'subselect',{ 'sub-002','sub-003','sub-004','sub-005','sub-006','sub-007','sub-008','sub-009',...
    'sub-010','sub-011','sub-012','sub-013','sub-014','sub-015','sub-016','sub-017','sub-018','sub-019'});
[STUDY, EEG] = pop_savestudy( STUDY, EEG, 'savemode','resave');
STUDY = pop_limo(STUDY, ALLEEG, 'method','WLS','measure','daterp','timelim],[-50 650],'erase','on','splitreg','on','interaction','off');

% 2nd level analysis
mkdir([STUDY.filepath filesep 'derivatives' filesep '1-way-ANOVA'])
cd([STUDY.filepath filesep 'derivatives' filesep '1-way-ANOVA'])
limo_random_select('Repeated Measures ANOVA',chanlocs,'LIMOfiles',...
    {[STUDY.filepath filesep 'LIMO_Face_detection' filesep 'Beta_files_Face_time_GLM_Channels_Time_WLS.txt']},...
    'analysis_type','Full scalp analysis','parameters',{4 5 6},...
    'factor names',{'face','repetition'},'type','Channels','nboot',1000,'tfce',0,'skip design check','yes');
```

FIGURE 3 | Code snippet for the two statistical models proposed. In both cases, only three steps are necessary: (1) create a statistical design (*std_makedesign.m*), (2) compute model parameters for each subject (*pop_limo.m*), and (3) perform the group level analysis (*limo_random_select.m*). Additional contrasts and figures/plots shown in the results section are also available in command lines, see https://github.com/LIMO-EEG-Toolbox/limo_meeeg/tree/master/resources/from_bids2stats.m for details.



immediately, unfamiliar faces repeated late, and a weighted least squares solution was used to obtain parameter estimates of each condition. At the second level, a repeated measure ANOVA (generalized Hotelling T^2) was conducted on beta estimates with “faces” and “repetition” as factors. Statistical significance was assessed using spatial-temporal clustering (Maris and Oostenveld, 2007; C.R. Pernet et al., 2015).

Results (Figure 4) revealed a significant effect of face type with two clusters (cluster 1 starts at 140 ms and ends at 424 ms, maximum F values 64.1281 at 280 ms on channel EEG017, corrected p -value 0.002; cluster 2 starts at 440 ms and ends at 648 ms, maximum F value 17.6071 at 616 ms on channel EEG057, corrected p -value 0.032) and a significant repetition effect with one cluster (cluster starts at 232 ms and ends at 648 ms, maximum

F value 51.3596 at 612 ms channel EEG045, corrected p -value 0.001). No significant interaction was observed. As such, the main effect of faces replicates Wakeman and Henson (2015) results who observed “a negative deflection peaking around 170 ms (“N170” component) larger for faces than scrambled faces, which does not differ for familiar and unfamiliar faces. Around 250 ms, a slower potential shift distinguishes familiar and unfamiliar faces until the end of the epoch.” Here we observed higher response at 140 ms (P1 component) for familiar and unfamiliar faces than scrambled faces not previously reported, followed by the same N170 effect—although weaker than reported (from their Figure 1, the face effect is around $\sim 4 \mu\text{V}$ while we observed a difference of $\sim 1.5 \mu\text{V}$, likely due to differences in preprocessing (see section “Fully Automated Preprocessing”) and

processing (i.e., LIMO EEG used a hierarchical linear model with weighted least squares parameter estimate per subjects while Wakeman and Henson averaged trials per subjects). From 250 ms, we also observed a slow potential separating familiar from unfamiliar faces.

Model 2: Regressing the Time Between Stimulus Presentation From Trial to Trial

At the first level, data from each participant were modeled with three experimental conditions (famous faces, scrambled faces, and unfamiliar faces) along with the time between the repetition of each stimulus (Figure 5) and a weighted least squares solution used to obtain parameter estimates of each condition. At the second level, a repeated measure ANOVA (generalized Hotelling T^2) was conducted on beta estimates of the time regressors with “faces” as factors, and a *post hoc* contrast computed, testing if the effect of time for famous faces differed from other stimuli. Statistical significance was again assessed using spatial-temporal clustering.

The *post hoc* contrast revealed a significantly stronger modulation of ERP as time passed between famous faces than for other stimuli (Figure 5—cluster 1 starts at 536 ms and ends at 576 ms, maximum 15.1112 at 556 ms channel EEG039, corrected *p*-value 0.031; cluster 2 starts at 548 ms and ends at 584 ms, maximum 9.58689 at 572 ms channel EEG055, corrected *p*-value 0.033). This suggests that there is an interaction face type by repetition such as late ERPs vary as a function of time between the first/second and third repetition for famous faces but not unfamiliar or scrambled faces.

DISCUSSION

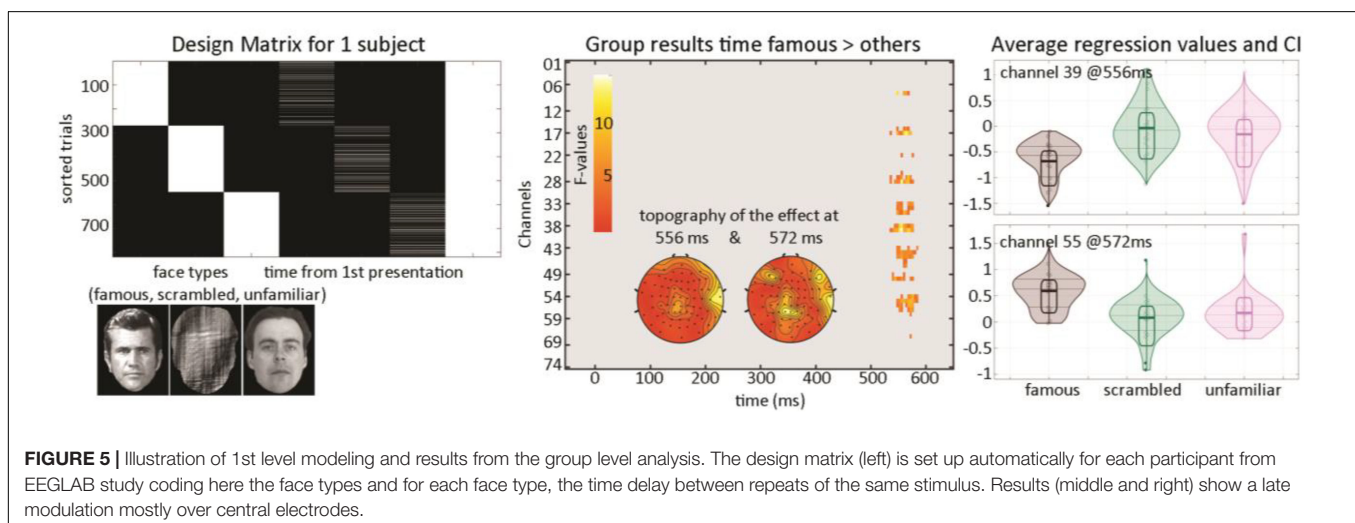
Following community guidelines (Pernet et al., 2020), we have implemented necessary changes and new tools allowing to create fully reproducible workflows with EEGLAB. The pipeline for the presented analysis is available at https://github.com/LIMO-EEG-Toolbox/limo_meeg/tree/master/resources/from_bids2stats.m

and further designs presented on the LIMO MEEG GitHub website (via the user interface and command-line alike https://github.com/LIMO-EEG-Toolbox/limo_tools/wiki).

A key development for EEG reproducibility is the recent development of EEG-BIDS (Pernet et al., 2019) which not only structures how data are organized and shared but also populates many of the metadata necessary for data analysis. The newly developed BIDS tools⁷ used here allow to import such data and create automatically an EEGLAB STUDY with the different experimental conditions. Note that from raw data imported into EEGLAB, those tools allow just as easy to export in the BIDS compliant format. While raw EEG data files often define channel labels, EEG-BIDS defines channel properties and associated labels (channels.tsv) corresponding to the electrodes for which locations are defined (electrodes.tsv) given a reference coordinate system (coordsystem.json). BIDS-Matlab-tools will always check for consistency between the data and BIDS meta-data and users have the choice on which information to use (in Figure 2, *pop_import_bids* parameter “bidschanlocs”). Similarly, raw EEG data files typically contain behavioral and experimental events; these are also defined in BIDS with separate text files (events.tsv). Sometimes the BIDS event files contain different information than the raw EEG data file and users have options to choose which one of the two types of event information to import (in Figure 2, *pop_import_bids* parameter “bidsevent”). Here, when preparing the BIDS dataset with EEG data only, events.tsv file was prepared as to include the nine experimental conditions but also the repetition order, distance, and time between repetitions which allowed to create the proposed design automatically.

Data preprocessing is performed here by first cleaning the raw data using the *clean_rawdata* EEGLAB plugin (v2.2), then by running the Infomax Independent Component Analysis algorithm (*runica* function of EEGLAB 2020.0) and performing automated ICA component labeling (*ICLabel v1.2.6*). EEGLAB includes a variety of algorithms and other approaches can be implemented to preprocess data automatically. One key

⁷<https://github.com/scn/bids-matlab-tools>



development for reproducible EEG artifact reduction is the use of the ICLABEL EEGLAB plugin which labels independent brain and non-brain components automatically thus allowing to remove artifactual components in a consistent manner (Pion-Tonachini et al., 2019). After preprocessing, various designs can be created from a STUDY (depending upon the type and quantity of events) and LIMO tools are called to run first level analyses automatically from which group-level analyses can be performed.

The current analysis focused on ERPs for sensor space, but EEGLAB-LIMO tools can be used similarly for spectral and time-frequency analyses. Analyses may also be applied to source resolved EEG data. LIMO tools already allow using ICA components as input, since ICA components have been shown to represent activity within localized patches of cortex (Delorme et al., 2012) and future versions will also allow automatic source space analyses. Source resolved activity calculated using eLoreta, and summarized using principal component analysis within regions of interest (ROI) is also possible with the ROIconnect plugin⁸ which allows exporting activity in ROI defined in standard atlases to EEGLAB native EEG linear decomposition matrix activity

⁸<https://github.com/arnodelorme/roiconnect>

REFERENCES

- Andersen, L. M. (2018). Group analysis in FieldTrip of time-frequency responses: a pipeline for reproducibility at every step of processing, going from individual sensor space representations to an across-group source space representation. *Front. Neurosci.* 12:261. doi: 10.3389/fnins.2018.00261
- Bigdely-Shamlo, N., Mullen, T., Kothe, C., Su, K. M., and Robbins, K. A. (2015). The PREP pipeline: standardized preprocessing for large-scale EEG analysis. *Front. Neuroinform.* 9:16. doi: 10.3389/fninf.2015.00016
- Chang, C.-Y., Hsu, S.-H., Pion-Tonachini, L., and Jung, T.-P. (2018). "Evaluation of artifact subspace reconstruction for automatic EEG artifact removal," in *Proceedings of the 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, (Honolulu, HI: IEEE), 1242–1245.
- Delorme, A., and Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J. Neurosci. Methods* 134, 9–21. doi: 10.1016/j.jneumeth.2003.10.009
- Delorme, A., Palmer, J., Onton, J., Oostenveld, R., and Makeig, S. (2012). Independent MEG sources are dipolar. *PLoS One* 7:e30135. doi: 10.1371/journal.pone.0030135
- Gorgolewski, K. J., Auer, T., Calhoun, V. D., Craddock, R. C., Das, S., Duff, E. P., et al. (2016). The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Sci. Data* 3, 160044.
- Hanke, M., and Halchenko, Y. (2011). Neuroscience Runs on GNU/Linux. *Front. Neuroinform.* 5:8. doi: 10.3389/fninf.2011.00008
- Jas, M., Larson, E., Engemann, D., Leppäkangas, J., Taulu, S., Hämäläinen, M., et al. (2018). A reproducible MEG/EEG group study with the MNE software: recommendations, quality assessments, and good practices. *Front. Neurosci.* 12:530. doi: 10.3389/fnins.2018.00530
- Kothe, C. A. E., and Jung, T.-P. (2016). *Artifact Removal Techniques with Signal Reconstruction*. US Patent App. 14/895,440 – Google Patents.
- Maris, E., and Oostenveld, R. (2007). Nonparametric statistical testing of EEG- and MEG-data. *J. Neurosci. Methods* 164, 177–190. doi: 10.1016/j.jneumeth.2007.03.024
- Martinez-Cancino, R., Delorme, A., Truong, D., Artoni, F., Kreutz-Delgado, K., Sivagnanam, S., et al. (2020). The open EEGLAB portal interface: high-performance computing with EEGLAB. *NeuroImage* 224, 116778. doi: 10.1016/j.neuroimage.2020.116778
- Meunier, D., Pascarella, A., Altukhov, D., Jas, M., Combrisson, E., Lajnef, T., et al. (2020). NeuroPycon: an open-source python toolbox for fast multi-modal and reproducible brain connectivity pipelines. *NeuroImage* 219, 117020. doi: 10.1016/j.neuroimage.2020.117020
- Niso, G., Tadel, F., Bock, E., Cousineau, M., Santos, A., and Baillet, S. (2019). Brainstorm pipeline analysis of resting-state data from the open MEG archive. *Front. Neurosci.* 13:284. doi: 10.3389/fnins.2019.00284
- Pernet, C. R., Appelhoff, S., Gorgolewski, K. J., Flandin, G., Phillips, C., Delorme, A., et al. (2019). EEG-BIDS, an extension to the brain imaging data structure for electroencephalography. *Sci. Data* 6, 103. doi: 10.1038/s41597-019-0104-8
- Pernet, C. R., Chauveau, N., Gaspar, C., and Rousset, G. A. (2011). LIMO EEG: a toolbox for hierarchical linear modeling of electroencephalographic data. *Comput. Intell Neurosci.* 2011:831409. doi: 10.1155/2011/831409
- Pernet, C. R., Garrido, M., Gramfort, A., Maurits, N., Michel, C., Pang, E., et al. (2020). Issues and recommendations from the OHBM COBIDAS MEEG committee for reproducible EEG and MEG research. *Nat. Neurosci.* 23, 1473–1483. doi: 10.1038/s41593-020-00709-0
- Pernet, C. R., Latinus, M., Nichols, T. E., and Rousset, G. A. (2015). Cluster-based computational methods for mass univariate analyses of event-related brain potentials/fields: a simulation study. *J. Neurosci. Methods* 250, 85–93. doi: 10.1016/j.jneumeth.2014.08.003
- Pion-Tonachini, L., Kreutz-Delgado, K., and Makeig, S. (2019). The ICLABEL dataset of electroencephalographic (EEG) independent component (IC) features. *Data Brief* 25, 104101. doi: 10.1016/j.dib.2019.104101
- Taulu, S., and Simola, J. (2006). Spatiotemporal signal space separation method for rejecting nearby interference in MEG measurements. *Phys. Med. Biol.* 51, 1759–1768. doi: 10.1088/0031-9155/51/7/008
- Wakeman, D. G., and Henson, R. N. (2015). A multi-subject, multi-modal human neuroimaging dataset. *Sci. Data* 2, 150001. doi: 10.1038/sdata.2015.1

(usually used for ICA activity), therefore, enabling their use in LIMO EEG.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: <https://openneuro.org/datasets/ds002718>.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by the Cambridge University Psychological Ethics Committee. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

RM-C, DT, and AD prepared the data. CP and AD wrote the code. SM overviewed the data preparation and pipeline. All authors contributed to the manuscript.