Report from Brainhack Global 2020 Montreal

Load fMRIprep confounds in Python

Project URL: https://github.com/SIMEXP/load_confounds

Pierre Bellec[1], Hao-Ting Wang[1,2??] and Steven Lee Meisler[3,4*]

*Correspondence:
smeisler@g.harvard.edu
[3]Harvard University, Cambridge, 86 Brattle Street, 02139, MA, USA
Full list of author information is available at the end of the article

1 Introduction

fMRIprep [1] is a popular minimal preprocessing software for functional MRI data. 'Minimal preprocessing' refers to motion correction, field unwarping, normalization, bias field correction, and brain extraction. Confound regression and smoothing are excluded from the workflow. Instead, fMRIprep provides users with a large set of potential confound regressors that covers many denoising strategies. The users have to select the confound regressors for denoising in subsequent analyses. Loading a sensible subset of confounds is difficult and error-prone for many strategies, such as ICA-AROMA [2] and CompCor [3]. `load_confounds` can extract confound variables and implement preset strategies for confound selections. The loaded format is compatible with `nilearn` analysis functions such as `NiftiMasker` and the GLM modules. Our aim was to provide a easy and foolproof API for users to perform subsequent denoising of `fMRIprep` outputs.

2 Progress

At Brianhack Global Montreal 2020, our goal was to prepare the package for a potential Beta release. To prepare for the release, related issues involved implementing several preset strategies and improving the user experience with better examples and error messages. Several issues had been identified before Brainhack, and the full discussion can be found under the `load_confounds` GitHub issue page. Participants at Brainhack were encouraged to pick up existing issues from the list. The following issues have been discussed and/or resolved:

2.1 Strategies

We worked on three strategies:

- Added ICA-AROMA [2] (contributed by Hao-Ting Wang)
- Added Scrubbing [4](contributed by Steven Meisler)
- Improved the anatomical mask selection for anatomical CompCor [3] (contributed by Steven Meisler)

2.1.1 ICA-AROMA

ICA-AROMA is only applicable to `fMRIprep` outputs generated with `usearoma`. `fMRIprep` produces files with suffix `descsmoothAROMAnonaggr_bold` and outputs ICA components in the confounds file. Using the `descsmoothAROMAnonaggr_bold` output is the recommanded way of applying ICA-AROMA and implemented in `load_confounds` as a preset strategy. When passing regular fMRIprep outputs suffixed with `descprepro_bold`, `load_confounds` retreives the noise independent compoenents for aggressive denoising.

2.1.2 Scrubbing

`load_confounds` provides both a basic and full approach for scrubbing. The basic scrubbing approach flags and censors time frames with excessive motion, using thresholds on framewise displacement and standardized DVARS. For full scrubbing, described in Power et al [4], after censoring volumes as in the basic approach, the full approach further remove continuous segments containing fewer than 5 volumes.

2.1.3 Anatomical CompCor

`fMRIprep` uses three kinds of anatomical masks to compute CompCor components, WM, CSF and the combination of the two. Metadata in fMRIPrep's confounds .json file specify the anatomical compartment associated with each component. The previous iteration of `load_confounds` did not consider the metadata, hence the difference of masks were not taken into account. In the revised aCompCor approach, `load_confounds` provides selections of anaotomical maps (WM, CSF, and combined) to avoid regressing relevant signal twice, which may introduce noise.

2.2 Demo

An executable demo using the nilearn developmental fMRI dataset (OpenNeuro ds000228) was added (contributed by Michael W. Weiss). The demo was adapted from an exisitng nilearn example on denoising 'Extracting signals from a brain parcellation'. This example notebook shows how to extract signals from a brain parcellation and compute a correlation matrix, using different denoising strategies from the package.

2.3 Error message

A new class `NotFoundConfoundException` was added for collecting all the missing parameters needed for the given noise component(s). A final error would be raised with the list of all parameters missing, rather than just the first encountered missing parameter. (Contributed by Michael W. Weiss and François Paugam)

2.4 Identify test dataset

OpenNeuro ds003 was adopted as the new test data, including all ICA-AROMA related components. However, non-steady-state volume and motion estmation mertic RMSD [5] are still missing. We plan on preprocessing the nilearn demo dataset (OpenNeuro ds000228) with the latest stable fMRIprep LTS release for collecting all possible confounds in the future. (Discussions amongs Pierre Bellec, Hao-Ting Wang, Elizabeth DuPre, and Chris Markiewicz)

2.5 All contributor bot

`allcontributors` bot is added to track community contributions (Contributed by Pierre Bellec).

2.6 Add `load_confounds` to `nixtract`

In addition to the main package, there was a collaborative project with the developers of `nixtract`. `nixtract` is a tool that extract and process timeseries data from neuroimaging files. Annabelle Harvey and Dan Gale added `load_confounds` as a dependency of `nixtract` for reading fMRIPrep confound variables.

3 Results

`load_confounds` has now covered most of the strategies tested in Ciric et al. [6]. The following noise components and a set of paramaters for dedicated approaches are supported.

- `motion` : motion parameters, including 6 translation/rotation (`basic`), and optionally derivatives, squares, and squared derivatives (`full`).
- `high_pass` : basis of discrete cosines covering slow time drift frequency band.
- `wm_csf` the average signal of white matter and cerebrospinal fluid masks (`basic`), and optionally derivatives, squares, and squared derivatives (`full`).
- `global` : the global signal (`basic`), and optionally derivatives, squares, and squared derivatives (`full`).
- `compcor` [3]: the results of a PCA applied on a mask based on either anatomy (`anat`) or temporal variance (`temp`). For aCompCor, one can choose either separate or combined WM and CSF compartments, as well as choosing to extract components contributing 50
- `ica_aroma` [2]: the results of an idependent component analysis (ICA) followed by identification of noise components. This can be implementing by incorporating ICA regressors (`basic`) or directly loading a denoised file generated by fMRIprep (`full`).
- `scrub` [4]: regressors coding for time frames with excessive motion, using thresholds on framewise displacement and standardized DVARS (`basic`) and suppressing short time windows using the (Power et al., 2014) appreach (`full`).

`load_confounds` can produce the following strategies from Ciric et al. [6]. The following table highlights the relevant options:

Strategy	high_pass	motion	wm_csf	global	compcor	ica_aroma	scrub
Params2	x		basic				
Params6	x	basic					
Params9	x	basic	basic	basic			full
Params9scrub	x	basic	basic				
Params24	x	full					
Params36	x	full	full	full			full
Params36scrub	x	full	full				
AnatCompCor	x	full			anat		
TempCompCor	x				temp		
ICAAROMA	x		basic			full	
AROMAGSR	x		basic	basic		full	
AggrICAAROMA	x		basic	basic		basic	

The future direction is to integrate `load_confounds` in `nilearn` for better reach to a wider range of users who may benifit from the package. To facilitate the nilearn inegration, we will add `sample_mask` to support volume-sensoring based scrubbing and improve the `nilearn` `NiftiMasker` related feature.

Availability of Supporting Data
Supplemental material has not been provided. More information about this project can be found at: `https://github.com/SIMEXP/load_confounds`. This report is generated at: `https://github.com/htwangtw/2021_08_Wang_load-confounds`.

Competing interests
None

Author's contributions
HTW, SLM wrote the software, HTW, SLM wrote the report

Acknowledgements
The authors would like to thank the organizers and attendees of Brainhack Global Montreal 2020. Especially, we would like to thank Elizabeth DuPre and Chris Markiewicz for discussions on the choice of test data.

Reviewers
No reviewers has been added yet.

Author details
[1]CRIUGM, Montreal, 4565 chemin Queen Mary, H3W 1W4, Quebec, Canada. [2]Brighton & Sussex Medical School, Brighton, Trafford Centre, BN1 9RX, East Sussex, UK. [3]Harvard University, Cambridge, 86 Brattle Street, 02139, MA, USA. [4]Massachusetts Institute of Technology, Cambridge, 46-4033 Vassar Street, 02139, MA, USA.

References
1. Esteban, O., Markiewicz, C.J., Blair, R.W., Moodie, C.A., Isik, A.I., Erramuzpe, A., Kent, J.D., Goncalves, M., DuPre, E., Snyder, M., *et al.*: fmriprep: a robust preprocessing pipeline for functional mri. Nature methods **16**(1), 111–116 (2019)
2. Pruim, R.H.R., Mennes, M., van Rooij, D., Llera, A., Buitelaar, J.K., Beckmann, C.F.: Ica-aroma: A robust ica-based strategy for removing motion artifacts from fmri data. NeuroImage **112**, 267–277 (2015). doi:10.1016/j.neuroimage.2015.02.064
3. Behzadi, Y., Restom, K., Liau, J., Liu, T.T.: A component based noise correction method (compcor) for bold and perfusion based fmri. NeuroImage **37**(1), 90–101 (2007). doi:10.1016/j.neuroimage.2007.04.042
4. Power, J.D., Barnes, K.A., Snyder, A.Z., Schlaggar, B.L., Petersen, S.E.: Spurious but systematic correlations in functional connectivity mri networks arise from subject motion. NeuroImage **59**(3), 2142–2154 (2012). doi:10.1016/j.neuroimage.2011.10.018
5. Jenkinson, M., Bannister, P., Brady, M., Smith, S.: Improved optimization for the robust and accurate linear registration and motion correction of brain images. NeuroImage **17**(2), 825–841 (2002). doi:10.1006/nimg.2002.1132
6. Ciric, R., Wolf, D.H., Power, J.D., Roalf, D.R., Baum, G.L., Ruparel, K., Shinohara, R.T., Elliott, M.A., Eickhoff, S.B., Davatzikos, C., Gur, R.C., Gur, R.E., Bassett, D.S., Satterthwaite, T.D.: Benchmarking of participant-level confound regression strategies for the control of motion artifact in studies of functional connectivity. Neuroimage **154**, 174–187 (2017)