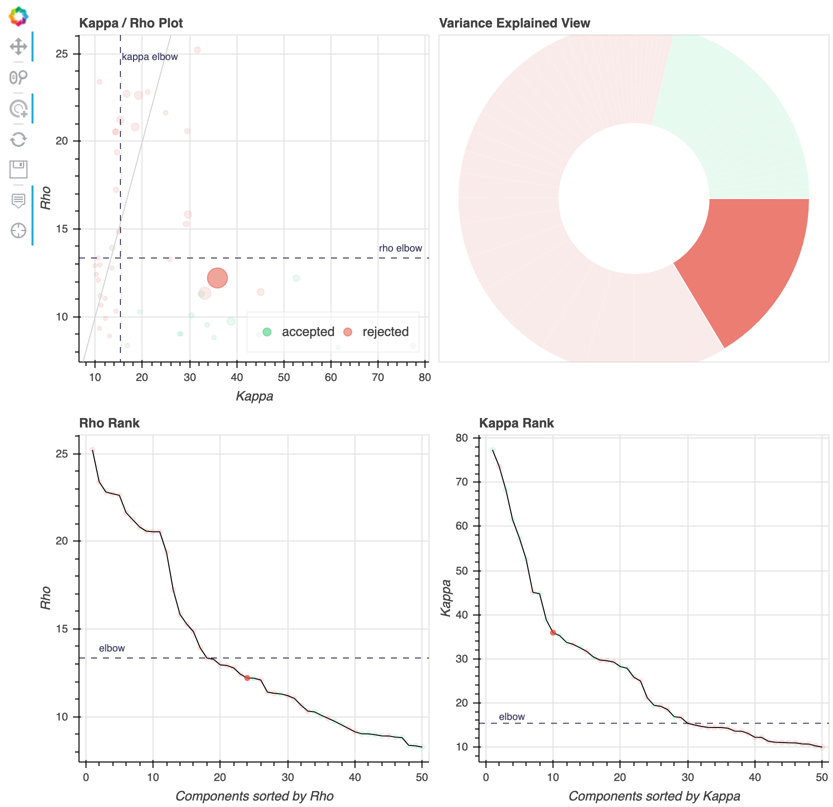
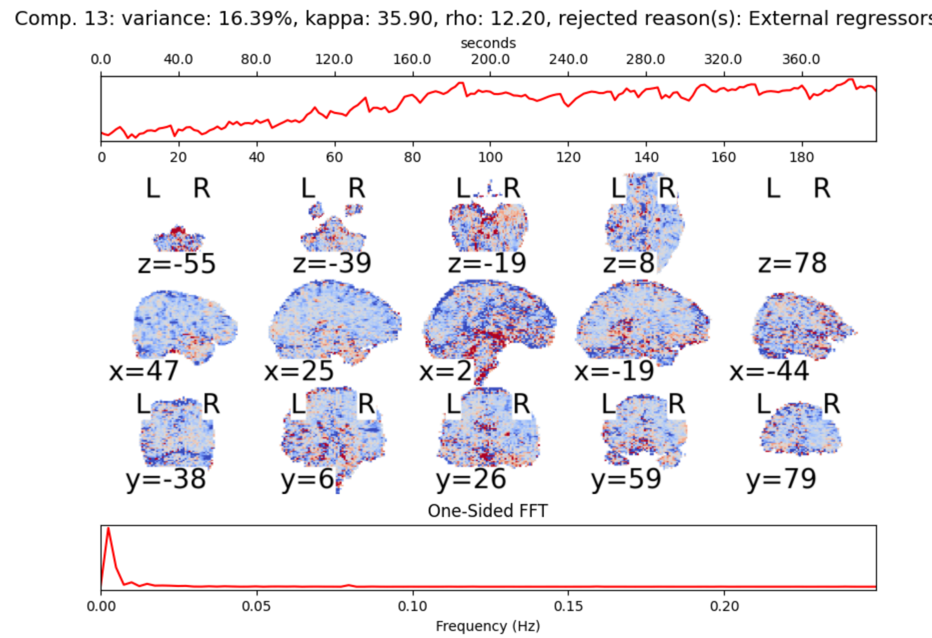
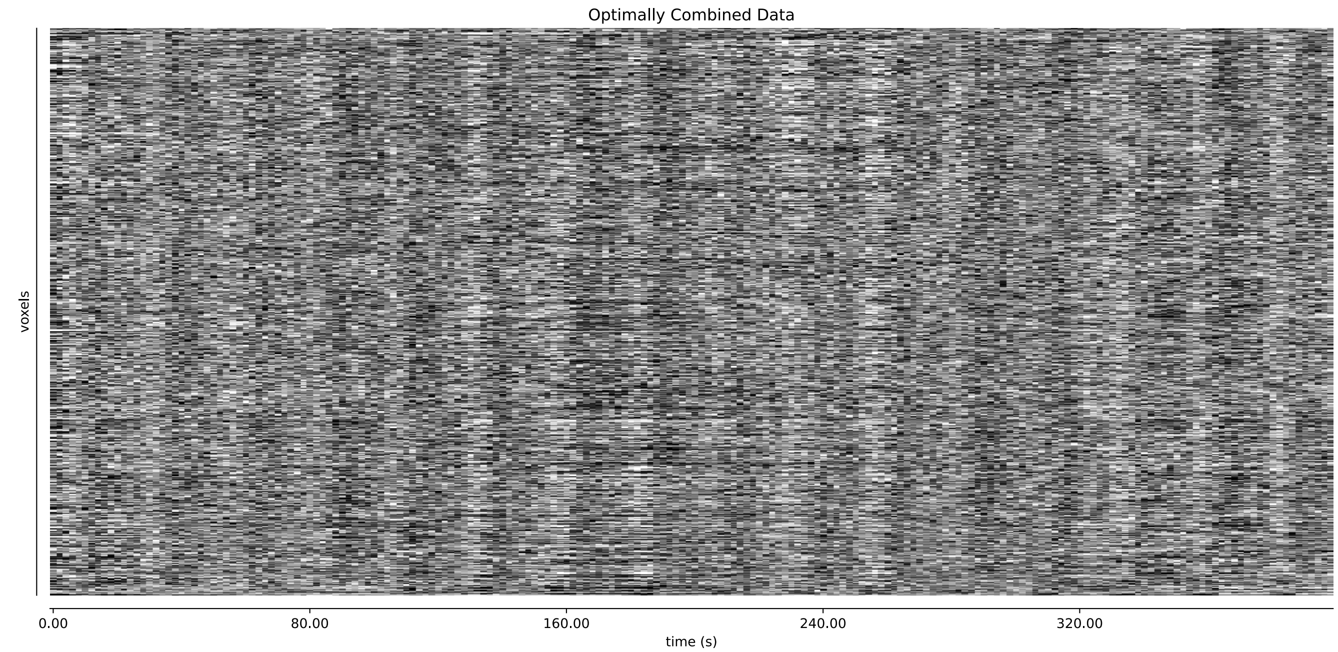
ICA components



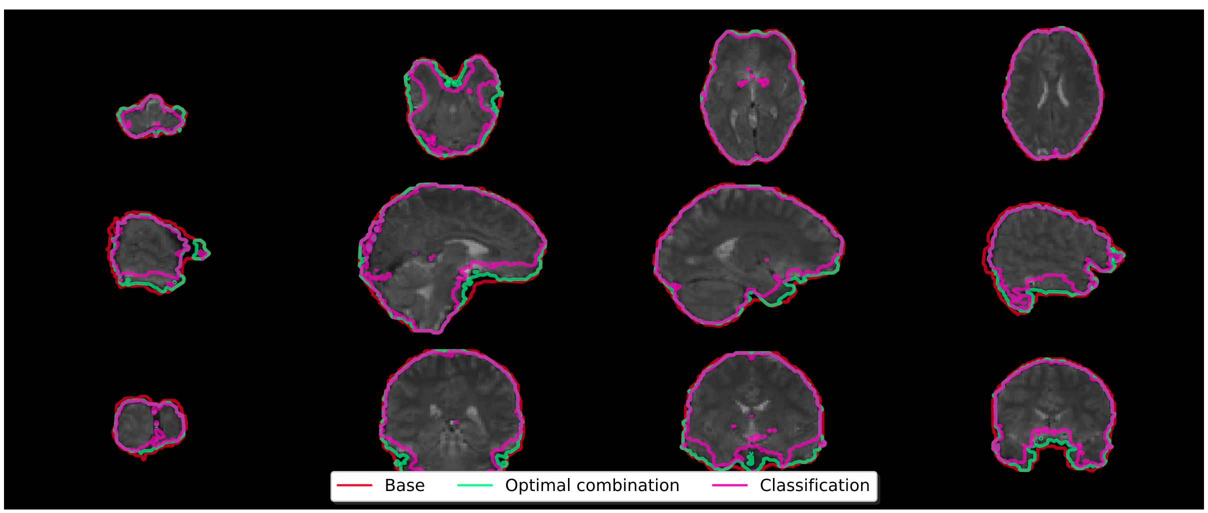


Carpet plots

Optimally combined | Denoised | Accepted | Rejected

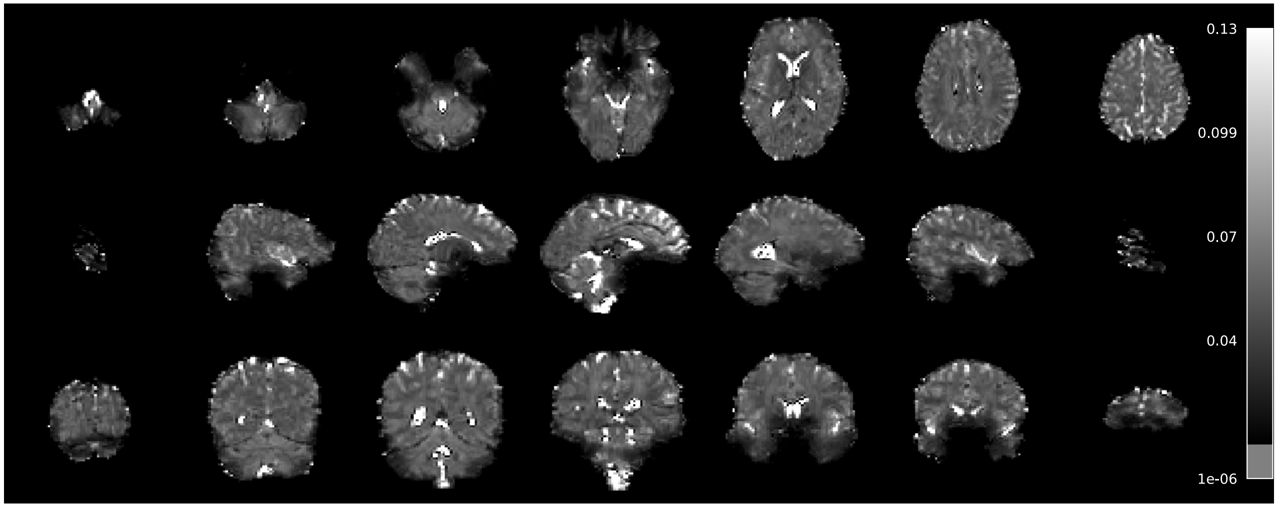


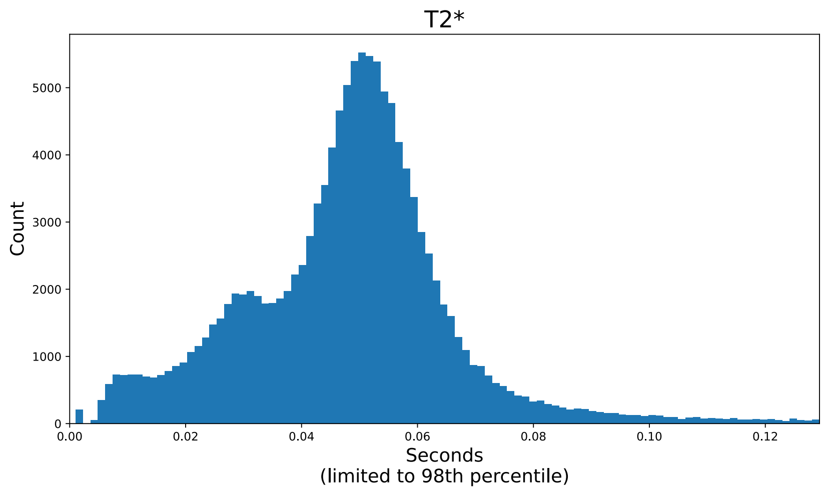
Adaptive mask



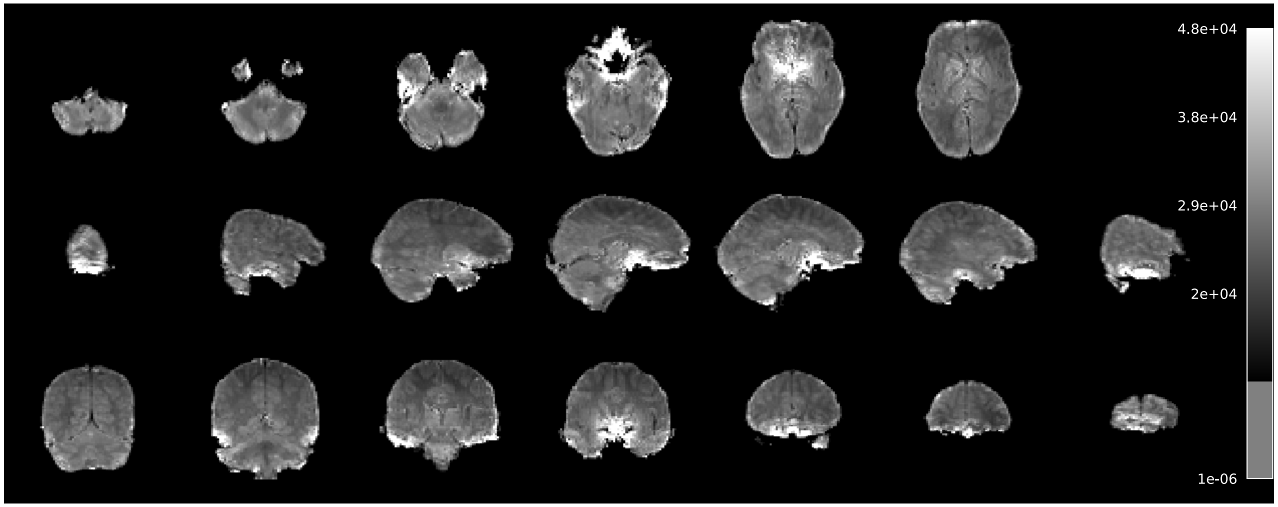
T2* and S0

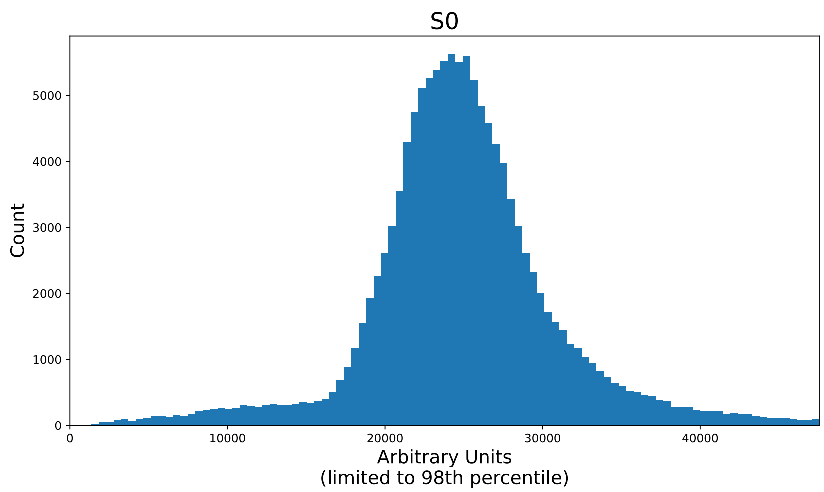
T2*



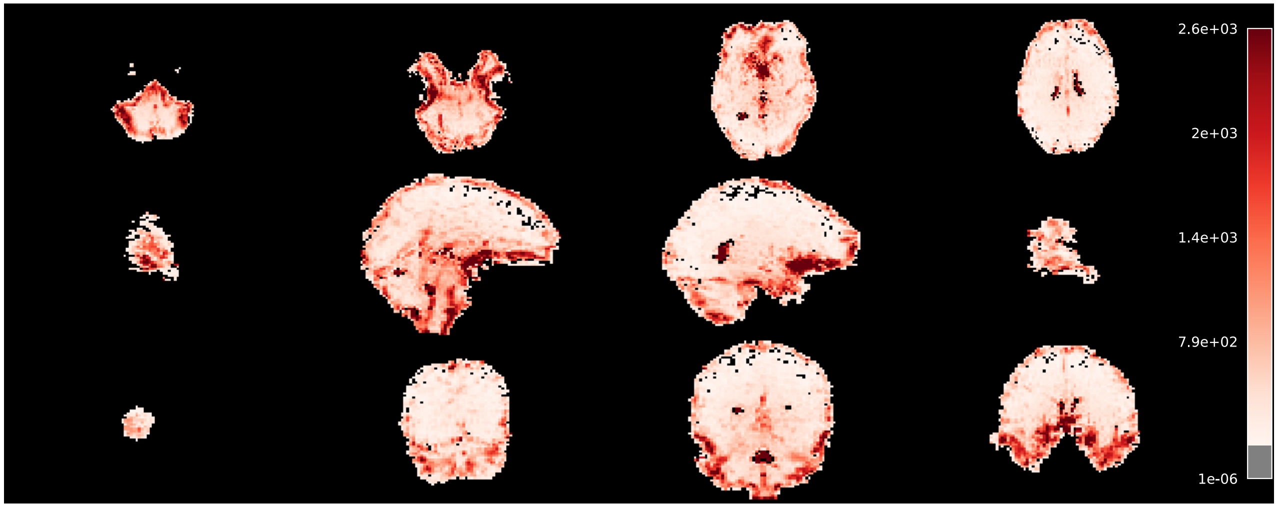


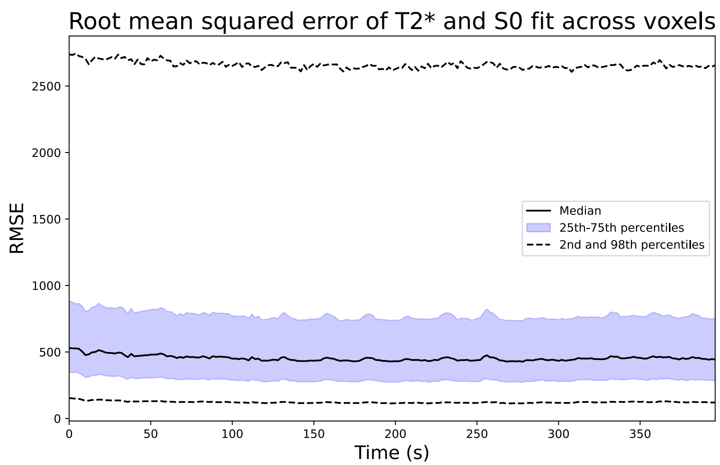
S0





T2* and S0 model fit (RMSE). (Scaled between 2nd and 98th percentiles)





Info

Tedana command used:

```
        tedana -d /Users/nar423/Documents/Data/tedana/sub-03_MOTOR/inputs/sub-
03_MOTOR_rm10_1_mc_brain.nii.gz /Users/nar423/Documents/Data/tedana/sub-03_MOTOR/inputs/sub-
03_MOTOR_rm10_2_mc_brain.nii.gz /Users/nar423/Documents/Data/tedana/sub-03_MOTOR/inputs/sub-
03_MOTOR_rm10_3_mc_brain.nii.gz /Users/nar423/Documents/Data/tedana/sub-03_MOTOR/inputs/sub-
03_MOTOR_rm10_4_mc_brain.nii.gz /Users/nar423/Documents/Data/tedana/sub-03_MOTOR/inputs/sub-
03_MOTOR_rm10_5_mc_brain.nii.gz -e 10.8 28.03 45.26 62.49 79.72 --tree
/Users/nar423/Documents/Data/tedana/sub-03_MOTOR/inputs/external_regressors_single_model.json --
external /Users/nar423/Documents/Data/tedana/sub-03_MOTOR/inputs/external_regressors.tsv --out-dir
/Users/nar423/Documents/Data/tedana/sub-03_MOTOR/output.extReg_motion --mask
/Users/nar423/Documents/Data/tedana/sub-03_MOTOR/inputs/sub-03_MOTOR_SBREF_1_bet_mask_ero.nii.gz --
```

```
verbose --debug
```

System:	Darwin
Node:	FSMC02CQ5YRML85
Release:	20.5.0
System version:	Darwin Kernel Version 20.5.0: Sat May 8 05:10:33 PDT 2021; root:xnu-7195.121.3~9/RELEASE_X86_64
Machine:	x86_64
Processor:	i386
Python:	3.9.17 (main, Aug 29 2023, 11:17:00) [Clang 13.0.0 (clang-1300.0.29.30)]
Tedana version:	24.0.2.dev99+gd6d3ccc
Other library versions:	{'bokeh': '2.2.3', 'mapca': '0.0.5', 'matplotlib': '3.7.2', 'nibabel': '5.1.0', 'nilearn': '0.10.4', 'numpy': '1.25.2', 'pandas': '2.0.3', 'scikit-learn': '1.3.0', 'scipy': '1.11.2', 'threadpoolctl': '3.2.0'}

About tedana

This is based on the minimal criteria of the original MEICA decision tree (Kundu et al. 2013) without the more aggressive noise removal steps (DuPre et al. 2021). TE-dependence analysis was performed on input data using the tedana workflow (DuPre et al. 2021). A user-defined mask was applied to the data. An adaptive mask was then generated using the dropout method(s), in which each voxel's value reflects the number of echoes with 'good' data. An adaptive mask was then generated using the dropout method(s), in which each voxel's value reflects the number of echoes with 'good' data. A two-stage masking procedure was applied, in which a liberal mask (including voxels with good data in at least the first echo) was used for optimal combination, T2*/S0 estimation, and denoising, while a more conservative mask (restricted to voxels with good data in at least the first three echoes) was used for the component classification procedure. A monoexponential model was fit to the data at each voxel using log-linear regression in order to estimate T2* and S0 maps. For each voxel, the value from the adaptive mask was used to determine which echoes would be used to estimate T2* and S0. Multi-echo data were then optimally combined using the T2* combination method (Posse et al. 1999). Principal component analysis based on the PCA component estimation with a Moving Average(stationary Gaussian) process (Li et al. 2007) was applied to the optimally combined data for dimensionality reduction. The following metrics were calculated: kappa, rho, countnoise, countsigFT2, countsigFS0, dice_FT2, dice_FS0, signal-noise_t, variance explained, normalized variance explained, d_table_score. Kappa (kappa) and Rho (rho) were calculated as measures of TE-dependence and TE-independence, respectively. A t-test was performed between the distributions of T2*-model F-statistics associated with clusters (i.e., signal) and non-cluster voxels (i.e., noise) to generate a t-statistic (metric signal-noise_z) and p-value (metric signal-noise_p) measuring relative association of the component to signal over noise. The number of significant voxels not from clusters was calculated for each component. Independent component analysis was then used to decompose the dimensionally reduced dataset. The following metrics were calculated: R2stat nuisance model, countsigFS0, countsigFT2, dice_FS0, dice_FT2, kappa, normalized variance explained, pval nuisance model, rho, signal-noise_t, variance explained. Kappa (kappa) and Rho (rho) were calculated as measures of TE-dependence and TE-independence, respectively. A t-test was performed between the distributions of T2*-model F-statistics associated with clusters (i.e., signal) and non-cluster voxels (i.e., noise) to generate a t-statistic (metric signal-noise_z) and p-value (metric signal-noise_p) measuring relative association of the component to signal over noise. {'External nuisance regressors that fit to components using a linear model were rejected.'}

Next, component selection was performed to identify BOLD (TE-dependent) and non-BOLD (TE-independent) components using a decision tree.

Next, component selection was performed to identify BOLD (TE-dependent) and non-BOLD (TE-independent) components using a decision tree.

This workflow used numpy (Van Der Walt et al. 2011), scipy (Virtanen et al. 2020), pandas (McKinney et al. 2010, pandas development team et al. 2020), scikit-learn (Pedregosa et al. 2011), nilearn, bokeh (Team et al. 2018), matplotlib (Hunter et al. 2007), and nibabel (Brett et al. 2019). This workflow also used the Dice similarity index (Dice et al. 1945, Sorensen et al. 1948).

References

- Brett, M., Markiewicz, C. J., Hanke, M., Côté, M.-A., Cipollini, B., McCarthy, P., … freec84 (2019 , May). *nipy/nibabel: 2.4.1*.
- Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, *26*(3), 297–302. URL: https://doi.org/10.2307/1932409, doi:10.2307/1932409
- DuPre, E., Salo, T., Ahmed, Z., Bandettini, P. A., Bottenhorn, K. L., Caballero-Gaudes, C., … others. (2021). Te-dependent analysis of multi-echo fmri with* tedana. *Journal of Open Source Software*, *6*(66), 3669. URL: https://doi.org/10.21105/joss.03669, doi:10.21105/joss.03669
- Hunter, J. D. (2007). Matplotlib: a 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi:10.1109/MCSE.2007.55
- Kundu, P., Brenowitz, N. D., Voon, V., Worbe, Y., Vértes, P. E., Inati, S. J., … Bullmore, E. T. (2013). Integrated strategy for improving functional connectivity mapping using multiecho fmri. *Proceedings of the National Academy of Sciences*, *110*(40), 16187–16192. URL: https://doi.org/10.1073/pnas.1301725110, doi:10.1073/pnas.1301725110

- Li, Y.-O., Adalı, T., & Calhoun, V. D. (2007). Estimating the number of independent components for functional magnetic resonance imaging data. *Human brain mapping*, *28*(11), 1251–1266. URL: https://doi.org/10.1002/hbm.20359, doi:10.1002/hbm.20359
- McKinney, W., & others. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference* (pp. 51–56). URL: https://doi.org/10.25080/Majora-92bf1922-00a, doi:10.25080/Majora-92bf1922-00a
- pandas development team, T. (2020 , February). *pandas-dev/pandas: Pandas*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: machine learning in python. *the Journal of machine Learning research*, *12*, 2825–2830. URL: http://jmlr.org/papers/v12/pedregosa11a.html
- Posse, S., Wiese, S., Gembris, D., Mathiak, K., Kessler, C., Grosse-Ruyken, M.-L., … Kiselev, V. G. (1999). Enhancement of bold-contrast sensitivity by single-shot multi-echo functional mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, *42*(1), 87–97. URL: https://doi.org/10.1002/(SICI)1522-2594(199907)42:1<87::AID-MRM13>3.0.CO;2-O, doi:10.1002/(SICI)1522-2594(199907)42:1<87::AID-MRM13>3.0.CO;2-O
- Sorensen, T. A. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, *5*, 1–34.
- Team, B. D. (2018). *Bokeh: Python library for interactive visualization*. URL: https://bokeh.pydata.org/en/latest/
- Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in science & engineering*, *13*(2), 22–30. URL: https://doi.org/10.1109/MCSE.2011.37, doi:10.1109/MCSE.2011.37
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … others. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, *17*(3), 261–272. URL: https://doi.org/10.1038/s41592-019-0686-2, doi:10.1038/s41592-019-0686-2