

mTRFpy: A Python package for temporal response function analysis

Ole Bialas^{1*}, Jin Dou^{1*}, and Edmund C. Lalor^{1,2}

¹ Department of Biomedical Engineering, University of Rochester, USA ² Department of Neuroscience, University of Rochester, USA ¶ Corresponding author * These authors contributed equally.

DOI: [10.21105/joss.05657](https://doi.org/10.21105/joss.05657)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: Britta Westner

Reviewers:

- @Saran-nns
- @sappelhoff

Submitted: 03 May 2023

Published: 11 September 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

Traditionally, studies on the neural processing of speech involved the repetitive display of isolated tokens (e.g., phonemes, words, sentences) where the properties of interest were carefully controlled. Recently, researchers have increasingly focused on investigating brain responses to more naturalistic speech like audiobooks ([Hamilton & Huth, 2020](#)). However, this approach demands statistical tools to account for the different sources of variance that naturally occur in speech. Among the most popular tools to model neural responses to naturalistic speech are multivariate temporal response functions (mTRFs).

One of the most commonly used packages for computing mTRFs with regularized regression is the mTRF-toolbox ([Crosse et al., 2016](#)). However, this toolbox is implemented in the proprietary MATLAB language, restricting accessibility for parts of the scientific community. To overcome this constraint, we present mTRFpy, a Python package which replicates and advances the functionality of the original mTRF-toolbox.

Background

In a nutshell, the mTRF is a regularized linear regression between two continuous signals, computed across multiple time-delays or lags. This accounts for the fact that the relationship between stimulus and neural response is not instantaneous and that the signals are auto-correlated.

mTRFs can be used as forward or encoding models to predict (multiple) univariate brain responses as the weighted sum of various acoustic and linguistic speech features while identifying their relative contributions ([Broderick et al., 2018](#); [Di Liberto et al., 2015](#)). In this case the model's weights have a clear physiological interpretation because they denote the expected change in neural response following a unit change in a given predictor ([Haufe et al., 2014](#)). Thus, they can be understood as a generalization of the event potential, obtained from averaging responses to repetitions of prototypical stimuli for continuous data.

mTRFs can also be used as backward or decoding models which reconstruct stimulus features from multivariate neural recordings, for example to identify the speaker an individual is attending to within a cocktail party scenario ([O'Sullivan et al., 2015](#)). Because a decoder pools information across all neural sensors, it can leverage interactions between individual observations and their underlying generators. Thus, their predictive power is usually higher compared to encoding models. However, because the neural signals are highly interrelated, the decoder will not only amplify relevant, but suppress irrelevant signals, making a physiological interpretation of the weights difficult ([Haufe et al., 2014](#)).

Statement of need

The temporal response function is a powerful and versatile tool to study the neural processing of speech in its natural complexity. It also allows researchers to conduct experiments that are engaging (e.g., listening to a conversation) while monitoring the comprehension of speech independently of classical behavioral tests. This makes mTRFs promising tools for clinical applications in infants or patients with schizophrenia, autism spectrum disorder or disorder of consciousness. We believe that mTRFs can be useful to a large clinical research community and hope that open and accessible software will facilitate their wider adoption.

We implement the same methods as the original MATLAB toolbox and use a sample data set to demonstrate that mTRFpy produces the same results (within the limits of numerical accuracy). However, we use an object oriented design, where training, optimization and visualization are implemented as methods of a generic TRF class. What is more, we added functions for permutation testing and model evaluation which were not included in the original mTRF-toolbox. Finally, we also included a method to conveniently export trained models to MNE-Python which is the most common framework for analyzing MEG and EEG data in Python ([Gramfort et al., 2013](#)).

There is some overlap with other Python packages focused on the neural processing of naturalistic speech such as eelbrain ([Brodbeck et al., 2021](#)) and naplib ([Mischler et al., 2023](#)). However, while these packages provide a whole analysis framework, mTRFpy is more minimalist in its implementation. This makes mTRFpy easy to learn, keeps dependencies to a minimum and makes the package easy to integrate into a broad variety of analysis pipelines.

Overview and Example

mTRFpy provides a sample of EEG recordings during comprehension of naturalistic speech. Here, we use this data set to compute and visualize a forward TRF. Then we estimate the model's accuracy as the Pearson's correlation between the actual and predicted EEG and compare the result against randomly permuted data.

```
import numpy as np
from matplotlib import pyplot as plt
from mtrf.model import TRF, load_sample_data
from mtrf.stats import cross_validate, permutation_distribution
stim, resp, fs = load_sample_data(n_segments=5)
trf = TRF(direction=1, method='ridge')
tmin, tmax = 0, 0.4 # time window in seconds
regularization = 6000 # ridge parameter
trf.train(stim, resp, fs, tmin, tmax, regularization)
r, _ = cross_validate(trf, stim, resp, fs, tmin, tmax, regularization)
r_perm, _ = permutation_distribution(
    trf, stim, resp, fs, tmin, tmax, regularization, n_permute=10000, k=-1)
p = sum(r_perm>r)/len(r_perm)
fig, ax = plt.subplots(1, 2, figsize=(7, 4))
trf.plot(channel='avg', axes=ax[0], show=False, kind='image')
ax[1].hist(r_perm, bins=100)
ax[1].axvline(r, 0, 1, color='black', linestyle='--')
ax[1].set(ylabel='Number of permutations', xlabel='Correlation [r]')
ax[1].text(0.04, 250, f'p={p.round(2)}')
```

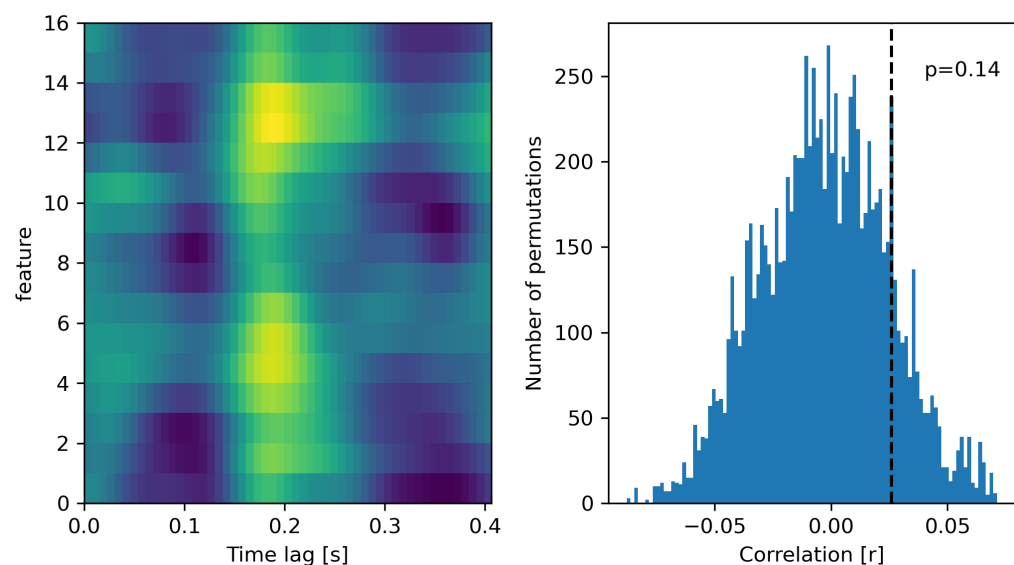


Figure 1: Left panel shows the TRFs weights, averaged across channels, for each spectral band where bright yellow indicates high and dark blue indicates low weights. The histogram on the right shows the distribution of correlation coefficients obtained by random permutation. The dashed line marks the actually observed value.

References

- Brodbeck, C., Das, P., Gillis, M., Kulasingham, J. P., Bhattasali, S., Gaston, P., Resnik, P., & Simon, J. Z. (2021). Eelbrain: A Python toolkit for time-continuous analysis with temporal response functions. *BioRxiv*, 2021–2008. <https://doi.org/10.1101/2021.08.01.454687>
- Broderick, M. P., Anderson, A. J., Di Liberto, G. M., Crosse, M. J., & Lalor, E. C. (2018). Electrophysiological correlates of semantic dissimilarity reflect the comprehension of natural, narrative speech. *Current Biology*, 28(5), 803–809. <https://doi.org/10.1016/j.cub.2018.01.080>
- Crosse, M. J., Di Liberto, G. M., Bednar, A., & Lalor, E. C. (2016). The multivariate temporal response function (mTRF) toolbox: A MATLAB toolbox for relating neural signals to continuous stimuli. *Frontiers in Human Neuroscience*, 10, 604. <https://doi.org/10.3389/fnhum.2016.00604>
- Di Liberto, G. M., O'Sullivan, J. A., & Lalor, E. C. (2015). Low-frequency cortical entrainment to speech reflects phoneme-level processing. *Current Biology*, 25(19), 2457–2465. <https://doi.org/10.1016/j.cub.2015.08.030>
- Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., & Hämäläinen, M. (2013). MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 267. <https://doi.org/10.3389/fnins.2013.00267>
- Hamilton, L. S., & Huth, A. G. (2020). The revolution will not be controlled: Natural stimuli in speech neuroscience. *Language, Cognition and Neuroscience*, 35(5), 573–582. <https://doi.org/10.1080/23273798.2018.1499946>
- Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., & Bießmann, F. (2014). On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage*, 87, 96–110. <https://doi.org/10.1016/j.neuroimage.2013.10.067>

- Mischler, G., Raghavan, V., Keshishian, M., & Mesgarani, N. (2023). naplib-python: Neural acoustic data processing and analysis tools in Python. *arXiv Preprint arXiv:2304.01799*. <https://doi.org/10.48550/arXiv.2304.01799>
- O'Sullivan, J. A., Power, A. J., Mesgarani, N., Rajaram, S., Foxe, J. J., Shinn-Cunningham, B. G., Slaney, M., Shamma, S. A., & Lalor, E. C. (2015). Attentional selection in a cocktail party environment can be decoded from single-trial EEG. *Cerebral Cortex*, 25(7), 1697–1706. <https://doi.org/10.1093/cercor/bht355>