**NeuroCluster: A Python package to detect electrophysiological activations related to continuous behavioral signals using time-frequency resolved multiple regression and non-parametric cluster-based permutation testing.**

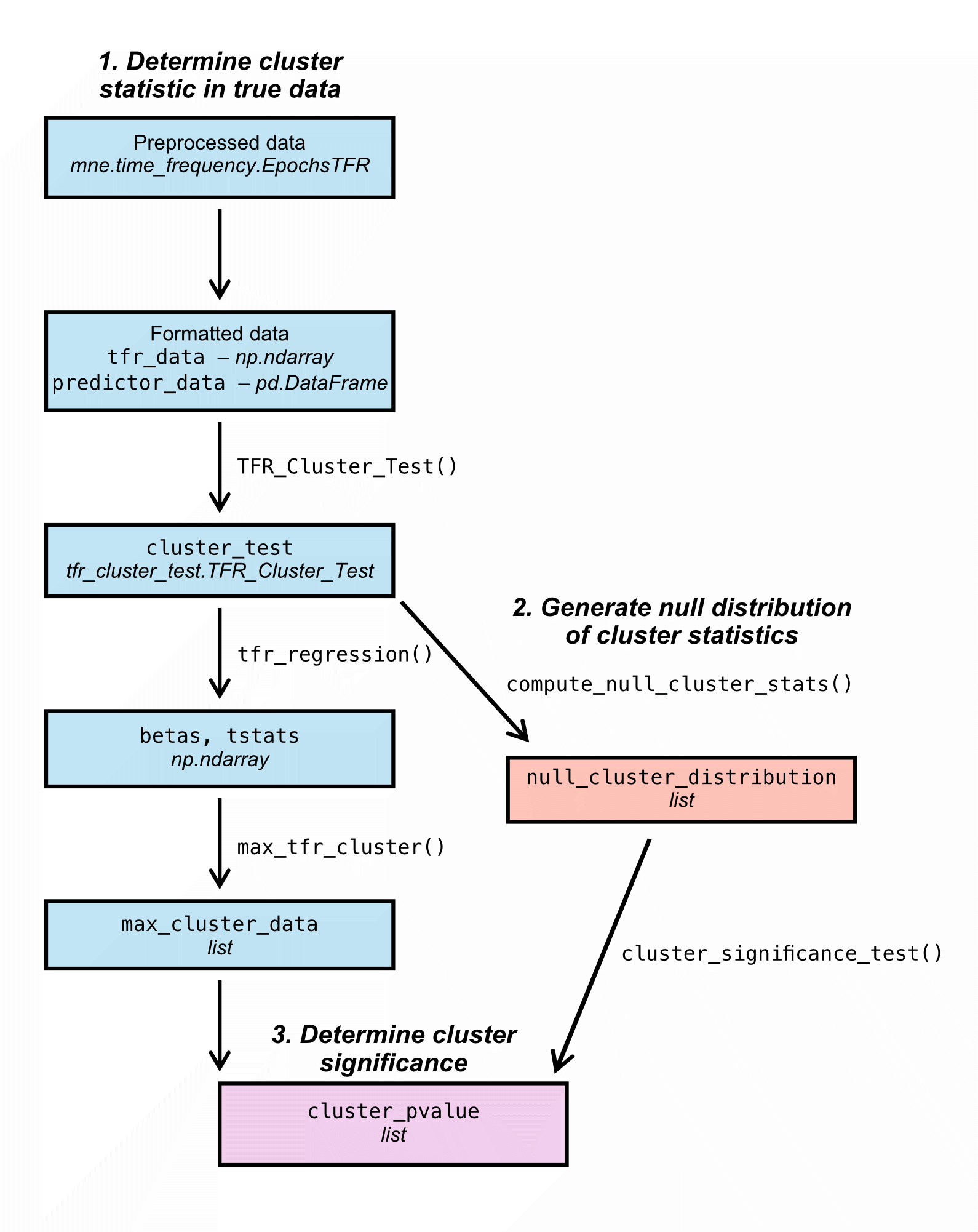
**# Summary**

Interpreting neurophysiological signals related to continuous, time-varying cognitive variables is a fundamental goal of cognitive and systems neuroscience. With the advent of new biotechnologies and neurosurgical practices, large-scale human (and animal) intracranial electrophysiological recordings are becoming widely accessible. However, classic statistical methods for analyzing event-related time series data are ill-equipped to manage the high dimensionality and complex dependencies in datasets from multi-region human intracranial electrophysiology (iEEG) containing simultaneous local field potential recordings from 100s of channels, in conjunction with complex data from human behavioral experiments. Field potential data is inherently multi-dimensional, reflecting the frequency, power, and phase of neuronal oscillations. Behavioral signals, which reflect experimental conditions, participant actions, or underlying cognitive processes, are often continuous and vary over time. These signals can be directly linked to neural activity, offering a dynamic way to study brain-behavior relationships. For instance, computational cognitive models generate trial-by-trial predictors of neural activity, such as reward prediction errors (RPEs; O’Doherty et al. (2007)). These continuous predictors help to model and understand the neural basis of complex behaviors observed in human experiments (Pan et al., 2024). Bridging neurophysiological recordings and complex behavioral analysis is crucial for understanding the neurophysiological mechanisms facilitating complex behaviors. However, the complexity of these recordings presents a significant challenge for neuroscientists using model-based analyses to uncover the neurophysiological signatures of these processes. NeuroCluster is amenable to model-free hypotheses as well, as continuous experimental (i.e., perceptual noise; Bang & Fleming (2018)) or behavioral (i.e., mood ratings; Blain & Rutledge (2020)) may be tested for their ability to predict neural activity using the same statistical approach.

**# Statement of Need**

NeuroCluster is an open-source Python-based method for time-series analyses of continuous behavioral variables related to neural data which contributes to existing body of tools for analyzing neuroscience data (cite FOOOF, MNE, eBOSC). This approach has been developed specifically for LFP recordings and can be used in conjunction with existing toolboxes, including MNE, which provides functionality for neural data preprocessing. Python tools like mne.stats.permutation\_cluster\_test use non-parametric permutation methods to identify significant clusters in time-frequency power estimates for discrete group comparisons. In contrast, NeuroCluster introduces a novel approach for non-parametric cluster-based identification of electrophysiological activation related to continuous behavioral variables across time and frequency domains in LFP recordings. It also supports multivariate analysis by incorporating multiple predictors to model neural activity. By operationalizing latent cognitive processes, computational cognitive models provide trial-by-trial continuous predictors of neural activity based on participants’ demonstrated behavior and the cognitive processes thought to underpin them (). NeuroCluster addresses a methodological gap in researchers’ ability to relate continuous latent cognitive predictors to underlying electrophysiological activity in the time and frequency domain to determine the neurocomputational processes which give rise to complex behaviors.

NeuroCluster is accompanied by a detailed tutorial [link to Jupyter notebook] which outlines the workflow (Fig 1) for implementing this approach with time-frequency power estimates from multi-region LFP recording.



**Fig 1.** **NeuroCluster workflow**. This approach involves three key steps: (1) determine cluster statistic in true data, (2) generate a null distribution of cluster statistics by permuting dataset, (3) determine significance of true cluster statistic against null distribution.

**# Methods**

Below we outline the statistical approach implemented by NeuroCluster for performing non-parametric permutation-based cluster testing using time-frequency power estimates from neural data estimated using Python-MNE and continuous predictors (i.e., latent cognitive processes, behavior, or experimental conditions). In these example data, we are testing the hypothesis that RPEs are significantly encoded in the electrophysiological signal from a given iEEG channel.

1. ***Determine cluster statistic in true data*** 
   1. **Define clusters:** At each time-frequency index, we perform a linear univariate (or multivariate) regression using behaviorally-derived independent variables (e.g., latent cognitive variables, behavioral measures, task conditions) to predict neuronal activity (i.e., power). The coefficient represents the strength and direction of the relationship between each independent variable and the dependent variable. It is estimated from the regression model and reflects how changes in the independent variable are associated with changes in power at the specific time-frequency pair. For each time-frequency pair, the coefficient for the regressor of interest (the independent variable of primary interest) is extracted from the regression results (**Fig 2A**). A t-statistic is computed for the coefficient to capture how significantly different it is from zero (**Fig 2B**). A significance threshold is applied to the t-statistics of the coefficient for the regressor of interest (**Fig 3**). If the t-statistic for a time-frequency pair exceeds the significance threshold, the pair is deemed significant. Clusters are then defined as adjacent time-frequency pairs where all pairs within the cluster have t-statistics exceeding the threshold, according to the test's desired tails (**Fig 2C**).
   2. **Compute cluster statistics:** For each identified cluster, sum the t-statistics of all time-frequency pairs within the cluster. In a two-tailed test (the default), compute both the maximum and minimum cluster sums (**Fig 2D**).
2. ***Generate null distribution of cluster statistics***
   1. **Permutation procedure:** Labels for the behavioral predictor of interest are shuffled for the desired number of permutations.
   2. **Recalculate cluster statistic:** Steps 1A/1B are repeated to define clusters and compute cluster statistics for each permuted dataset**.**
   3. **Construct null distribution:** The cluster statistics from all permutations are compiled to create a null distribution, representing the distribution of cluster statistics under the null hypothesis (**Fig 2E**).
3. ***Determine cluster significance*** 
   1. **Compare true cluster statistic to null distribution to compute p-values:**The proportion of cluster statistics in the null distribution falling above (or below) the true cluster statistic(s) determines the p-value associated with the cluster(s) identified in the true data (**Fig 2E**).

******

**Fig 2. NeuroCluster methods. A.**  coefficients for continuous predictor of interest (RPE) predicting power in given time-frequency pair (red outline = maximum positive cluster; blue outline = maximum negative cluster). **B.** T-statistics corresponding with RPE coefficients. C. Clusters as determined using t-critical threshold. **D.** Maximum positive and negative clusters determined by summing t-statistics in identified clusters. **E.** Null distribution of cluster statistics generated by permuting dataset for predictor of interest (100 permutations; red dashed line = true cluster statistic).

**References**

[**https://www.pnas.org/doi/full/10.1073/pnas.1800795115**](https://www.pnas.org/doi/full/10.1073/pnas.1800795115) **- Bang & Fleming (2018)**

[**https://elifesciences.org/articles/57977**](https://elifesciences.org/articles/57977) **- Blain & Rutledge (2020)**

[**https://arxiv.org/abs/2406.14742**](https://arxiv.org/abs/2406.14742) **- Pan et al (2024)**

[**https://pubmed.ncbi.nlm.nih.gov/17416921/**](https://pubmed.ncbi.nlm.nih.gov/17416921/) **- O’Doherty et al. (2007)**

Donoghue T, Haller M, Peterson EJ, Varma P, Sebastian P, Gao R, Noto T, Lara AH, Wallis JD, Knight RT, Shestyuk A, Voytek B (2020). Parameterizing neural power spectra into periodic and aperiodic components. Nature Neuroscience, 23, 1655-1665. DOI: 10.1038/s41593-020-00744-x **– FOOOF reference**

Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7(267):1–13, 2013. [doi:10.3389/fnins.2013.00267](https://doi.org/10.3389/fnins.2013.00267). **– MNE-Python**

**eBOSC references:**Kosciessa, J. Q., Grandy, T. H., Garrett, D. D., & Werkle-Bergner, M. (2020). Single-trial characterization of neural rhythms: Potential and challenges. NeuroImage, 206, 116331. <http://doi.org/10.1016/j.neuroimage.2019.116331>

Whitten, T. A., Hughes, A. M., Dickson, C. T., & Caplan, J. B. (2011). A better oscillation detection method robustly extracts EEG rhythms across brain state changes: The human alpha rhythm as a test case. NeuroImage, 54(2), 860–874. <http://doi.org/10.1016/j.neuroimage.2010.08.064>