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

**# Summary**

Cognitive neurophysiology offers a novel framework for studying cognitive brain-behavior relationships by relating electrophysiological signals to complex behaviors. With the advent of new biotechnologies and neurosurgical practices, large-scale human (and animal) intracranial electrophysiological recordings are becoming widely accessible. As a result, cognitive neurophysiologists can design cognitive experiments that leverage both the spatiotemporal resolution of electrophysiological data and the complexity of continuous behavioral variables (EX CITATIONS). Analyzing these data requires sophisticated statistical methods that can interpret multidimensional neurophysiological data and dynamic, continuous behavioral variables. Classical statistical frameworks for analyzing event-related time series data are ill-equipped to manage the high dimensionality and behavioral complexity of cognitive neurophysiology studies. NeuroCluster is an open-source Python toolbox for analysis of multivariate electrophysiological data related to complex, continuous behavioral variables. NeuroCluster introduces a novel statistical approach, which uses non-parametric cluster-based permutation testing to identify time-frequency clusters of oscillatory power modulations that significantly encode time-varying, continuous behavioral variables. It also supports multivariate analyses by allowing for multiple behavioral predictors to model neural activity. NeuroCluster addresses a methodological gap in statistical approaches to relate continuous, cognitive predictors to underlying electrophysiological activity with time and frequency resolution, to determine the neurocomputational processes giving rise to complex behaviors.

**# Statement of Need**

Determining the neurocomputational processes that give rise to human cognition and generate complex behaviors is a fundamental goal of cognitive and systems neuroscience. Cognitive neurophysiologists study the neural underpinnings of latent cognitive processes by relating complex behavioral signals to electrophysiological time series data. Cognitive behavioral signals, which can reflect experimental conditions, participant actions, or underlying cognitive processes, are often continuous and vary over time, especially in human behavioral experiments. Computational cognitive models are used to operationalize unobservable cognitive processes and provide estimates of latent cognitive variables, based on participants’ behaviors (Pan et al., 2024). For instance, some cognitive models generate continuous, trial-wise value estimates, like reward prediction errors (RPEs, O’Doherty et al. (2007). Directly linking these cognitive variables to neurophysiological activity offers a dynamic way to study brain-behavior relationships. Unfortunately, the innate complexities of both cognitive behaviors and electrophysiological data (CITATION) presents a significant challenge for neuroscientists using model-based analyses to uncover the neurophysiological signatures of these processes.

NeuroCluster addresses a methodological gap in cognitive neurophysiology, by providing a novel statistical pipeline to relate continuous latent cognitive predictors to underlying electrophysiological activity, with both time and frequency resolution. Non-parametric statistical testing is the standard approach to analyze event-related time series data while controlling for multiple comparisons problems and reducing family-wise error rates (CITATIONS). However, current statistical methods are ill-equipped to interpret complex, cognitive behaviors, nor can they manage the high dimensionality of multi-region intracranial electrophysiological recordings (CITATION). Classic analysis methods relate neuronal activity to discrete behavioral categories (generally two conditions), rather than continuous, trial-by-trial behavioral measures (CITATION). Neurophysiological activity is typically aggregated by trial-type to perform a two-sample cluster-based permutation test, which tests whether the neuronal encoding patterns differ between two discrete task variables (CITATION). While two-sample permutation tests provide neurophysiological results in the time and frequency domains, they are insufficient for analyses relating neuronal activity to time-varying, continuous behavioral variables. Unfortunately, standard analysis methods capable of relating neural activity to complex, continuous variables sacrifice spectral resolution in either the time (trial-averaged signals) or frequency domains (broadband frequency-averaged signals) (CITATION). Reducing the spatiotemporal resolution of electrophysiological data hinders our ability to define distinct underlying mechanisms of cognitive processes, by eliminating either the temporal profile of a signal containing within and across-region encoding onset, duration, and latency patterns (CITATION) or eliminating the signal’s frequency-specificity, despite the widely accepted theory that oscillatory activity at different frequencies corresponds to distinct neurophysiological mechanisms (CITATION). Additionally, these approaches require neurophysiologists to define *a priori* hypotheses for relevant within-trial epochs and/or frequencies, reducing the generalizability of these analyses (CITATION). NeuroCluster addresses these shortcomings by implementing a novel statistical approach to identify significant clusters of oscillatory power modulations, with time-frequency resolution, related to trial-varying, continuous behavioral variables.

NeuroCluster is an open-source Python toolbox for identification of electrophysiological time-frequency activity related to continuous behavioral variables, using non-parametric cluster-based permutation testing (CITATION). We demonstrate our approach with human intracranial local field potentials but NeuroCluster provides functionality for all types of spatiotemporal or spectrotemporal neurophysiological measures (EEG, MEG) (CITATION) and may be applicable to phase-amplitude or phase-phase cross-frequency coupling analyses (CITATION). NeuroCluster is designed to supplement existing Python-based electrophysiological analysis toolboxes (CITATION FOOOF, MNE, eBOSC), such as MNE, which currently offers a cluster-based permutation testing approach for discrete group comparisons (mne.stats.permutation\_cluster\_test) (CITATION MNE). Additionally, NeuroCluster performs analyses with multivariate behavioral data by incorporating multiple predictors to model neural activity (CITATION?). NeuroCluster is amenable to analyses using the same statistical approach for model-based latent cognitive predictors (Pan et al., 2024, O’Doherty et al. 2007), model-free cognitive variables (CITATION), as well as continuous experimental (i.e., perceptual noise; Bang & Fleming (2018)) or behavioral (INTEROCEPTIVE?) (i.e., mood ratings; Blain & Rutledge (2020)) predictors. Our novel statistical method is applicable for numerous analysis goals; the major use cases are performing an initial exploratory analysis to generate specific hypotheses, determine data-driven temporal windows and/or frequencies of interest, or to identify regional patterns of significant clusters within and between subjects (CITATION). Future directions for NeuroCluster may implement mixed effects regressions, multifrequency cluster detection, and/or group-level analysis tools (CITATIONS). NeuroCluster addresses a methodological gap in cognitive neurophysiology by implementing a novel statistical framework to relate continuous latent cognitive predictors to underlying time-frequency resolved neurophysiological signals. Directly linking electrophysiological activity to cognitive variables is crucial to understand the neurophysiological mechanisms facilitating complex behaviors and cognition.

**# Documentation \*\*\*** Comment on things in code for speed?

\*\*\* define TFR somewhere!!

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.

A diagram of cluster

Description automatically generated

**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.

Below we outline the statistical approach implemented by NeuroCluster for performing non-parametric permutation-based cluster testing using time-frequency resolved 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 time-frequency representation (TFR).

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**).

***A screenshot of a graph

Description automatically generated***

**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).

**Acknowledgements**

Shawn, other PIs

**References**

CM 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>

**# Statistical Approach**

Additionally, field potential data is inherently multi-dimensional due to the biophysical properties of neuronal oscillations, which are comprised of frequency, power, and phase components for every sample in the time series. The complexity of neurophysiological recordings poses a substantial challenge to neural data scientists trying to decode oscillatory signals related to specific trial-by-trial events.

* Make sure to clarify that these should be standardized regression coefficients (maybe cite salman’s 2std standardization)
* Lack of time-frequency resolved regression approaches for complex behaviors
  + Frequency resolution is important – frequency-specific modulation is widely accepted and understood that oscillations of different frequencies correspond to distinct underlying mechanisms
  + Time resolution is important – can understand the temporal encoding properties of a given region – ie how long does it take information about signal X to get to Region Y. AND timing resolution allows you to construct a pseudo network, where the timing of Regions A,B,C can be related to the involvement of these regions on a mechanistic level – if region A has a cluster that’s 100 ms after region B, that’s a hint that the signal from B may be transmitted to A and gives potential hypothesis-driven directions for connectivity metrics
* Because of the nature of electrophysiological data, one cannot assume that you can determine task-relevant power modulations based on the assumption that any power modulations are going to be task-related
  + Because of the biophysical properties of electrophysiological data, there is always going to be some time-frequency structure in the data – how do you parse what is not just event related, but event related signals specific to your predictor of interest
  + Requires a priori hypothesis for relevant frequencies/times
  + Current approach 1: Collapsing relevant multi-dimensional data into trial-averaged signals from pre-defined frequency bands loses the richness and uniqueness of spectrotemporal data
  + Current approach 2: two-sample cluster based permutation testing – no behavioral complexity, can’t account for possible covariates

Unique approach allows 1) multiple regression with time frequency resolution 2) non-parametric cluster-based permutation testing specific to behavioral variable of interest, controlling for null hypothesis that cluster is due to unrelated structure in data

Unique approach allows 1) multiple regression with time frequency resolution 2) non-parametric cluster-based permutation testing specific to behavioral variable of interest, controlling for null hypothesis that cluster is due to unrelated structure in data

* Lack of time-frequency resolved regression approaches for complex behaviors
  + Frequency resolution is important – frequency-specific modulation is widely accepted and understood that oscillations of different frequencies correspond to distinct underlying mechanisms
  + Time resolution is important – can understand the temporal encoding properties of a given region – ie how long does it take information about signal X to get to Region Y. AND timing resolution allows you to construct a pseudo network, where the timing of Regions A,B,C can be related to the involvement of these regions on a mechanistic level – if region A has a cluster that’s 100 ms after region B, that’s a hint that the signal from B may be transmitted to A and gives potential hypothesis-driven directions for connectivity metrics

Because of the nature of electrophysiological data, one cannot assume that you can determine task-relevant power modulations based on the assumption that any power modulations are going to be task-related

* + Because of the biophysical properties of electrophysiological data, there is always going to be some time-frequency structure in the data – how do you parse what is not just event related, but event related signals specific to your predictor of interest
  + Requires a priori hypothesis for relevant frequencies/times
  + Current approach 1: Collapsing relevant multi-dimensional data into trial-averaged signals from pre-defined frequency bands loses the richness and uniqueness of spectrotemporal data
  + Current approach 2: two-sample cluster based permutation testing – no behavioral complexity, can’t account for possible covariates

## Multiple Comparisons Correction Using Parametric Test Statistic Thresholding

For example, a hypothetical iEEG study has a dataset of N=15 subjects that engage in a cognitive task with n=150 trials. For each task trial, you extract 3.0 seconds from every neural recording. If there are approximately n=750 channels across subjects, each time series will contain 2.25×10^5 data points in the time domain (n=150 trials, epochs = 3.0 s, sampling rate = 500Hz). After spectral decomposition (n=30 frequencies, Morlet wavelets), the spectral domain expands from N=1 dimensions to N=30 of power estimates for every frequency, for every time point, giving . Now each time series has N=6.75x10^7 data points, or 1.0125x10^8 data points across subjects. In addition to the computational load limitations for a dataset of this magnitude, hypothesis testing using standard statistical inference to relate time-frequency power modulations to trial-wise continuous variables is uninterpretable without multiple comparisons correction (MCC) to prevent inflated Type I & II error rates. Standard MCC approaches cannot sufficiently address a problem of such a magnitude. For example, using Bonferroni correction to reduce the significance threshold (alpha=0.05) by the number of independent tests (6.75x10^7), increasing the False Discovery Rate (FDR) beyond acceptable limits [Maris, 2011].

Here we propose a novel statistical framework to perform time-frequency resolved multiple regression to predict trial-wise power modulations from multivariate predictors. For every pixel in time-frequency space, the following regression is performed:

Yf,t [matrix notation for n epochs] = A black and white text

Description automatically generated

\*\*\*make formula in latex + make sure to cite statsmodels OLS [summarize: β i = partial slope coefficient (also called partial regression coefficient, metric coefficient). It represents the change in E(Y) associated with a one-unit increase in Xi when all other IVs are held constant. α = the intercept. Geometrically, it represents the value of E(Y) where the regression surface (or plane) crosses the Y axis. Substantively, it is the expected value of Y when all the IVs equal 0. ε = the deviation of the value Yj from the mean value of the distribution given X. This error term may be conceived as representing (1) the effects on Y of variables not explicitly included in the equation, and (2) a residual random element in the dependent variable. From <https://www3.nd.edu/~rwilliam/stats2/l02.pdf>]

to estimate the partial regression coefficient of the predictor of interest for every TFR pixel (t,f).

The significance of partial slope coefficients is computed via parametric significance testing under given null hypothesis using an observed test statistic, specifically the *t*-statistic. For every pixel, we calculate the observed *t*-value of the partial slope for the regressor of interest from a Student *t*-distribution with df=N-K-1. Though each pixel(t,f) regression is unique, the degrees of freedom for every regression model are equal because the predictor data is identical across pixels, making the *t*-distributions for every regression identical (T~tdf). Since each pixel-wise *T*-distribution is identical, we are able to compute a global *t*-statistic, *t-critical*, that can be used to simultaneously threshold the observed *t*-values in time-frequency space.

\*\*insert t stat equation + t values heatmap\*\*

\*use this fig to explain possible null hypotheses, defaults, and rationale\*

A diagram of a normal distribution

Description automatically generated

* Pixel-level correction of FWER –
  + T-statistic from regression is to test the hypothesis that the regression coefficients are not zero – NOT the power values

### Multiple Comparisons Correction using Cluster-Based Permutation Tests

* Non parametric statistics is the gold standard for statistical testing of event related signals to control for FWER, type I errors – can be applied to univariate + multivariate data
* Emphasize importance of channel-specific null distribution & highlight that baseline oscillatory dynamics vary across regions/individuals so you must create a null distribution for each electrode

Unique approach allows 1) multiple regression with time frequency resolution 2) non-parametric cluster-based permutation testing specific to behavioral variable of interest, controlling for null hypothesis that cluster is due to unrelated structure in data

Methods description/functionality:

* Cluster-based permutation testing approach: must test clusters against non-parametric null hypothesis, rather than frequentist null hypothesis.
  + Cluster detection identifies latent structures in multi-dimensional data that are not just spurious signals (one pixel signal can’t be a significant ‘encoding’ profile)
  + Cannot assume time-freq power is normally distributed around zero – must build a null distribution from permuted regressions to capture spectrotemporal clusters specifically related to parameter of interest, not all event related signals
  + Non parametric statistics: doesn’t rely on assumptions about data distributions + makes multiple comparisons correction easy \* give example of Bonferroni correction p value\* (Cohen, 2014) [chapter 13] errors – can be applied to univariate + multivariate data
  + Emphasize importance of channel-specific null distribution & highlight that baseline oscillatory dynamics vary across regions/individuals so you must create a null distribution for each electrode
    - For each channel, generate a unique null distribution of time-frequency clusters related to permuted predictor data
    - A unique null distribution is generated for each channel to control for channel-specific spectrotemporal profiles
  + Highlight that this approach keeps time-frequency structure in-tact and allows us to test whether the cluster is specifically related to predictor of interest or whether it’s a false positive due to underlying structure in channel data

Emphasize importance of channel-specific null distribution & highlight that baseline oscillatory dynamics vary across regions/individuals so you must create a null distribution for each electrode

* Emphasize importance of channel-specific null distribution & highlight that baseline oscillatory dynamics vary across regions/individuals so you must create a null distribution for each electrode
* Emphasize importance of channel-specific null distribution & highlight that baseline oscillatory dynamics vary across regions/individuals so you must create a null distribution for each electrode