

NeuroCAPs: A Python Package for Performing Co-Activation Patterns Analyses on Resting-State and Task-Based fMRI Data

Donisha Smith¹

¹ Department of Psychology, Florida International University

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

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

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Co-Activation Patterns (CAPs) is a dynamic functional connectivity technique that clusters similar spatial distributions of brain activity. To make this analytical technique more accessible to neuroimaging researchers, NeuroCAPs ([D. Smith, 2025a](#)), an open source Python package, was developed to perform end-to-end co-activation patterns (CAPs) analyses on either resting-state or task-based fMRI data. Additionally, it is published under the MIT license.

Background

Numerous functional magnetic resonance imaging (fMRI) studies employ static functional connectivity (sFC) techniques to analyze connectivity within and between brain regions. These approaches operate under the assumption that functional connectivity patterns, defined as correlation in activity between brain regions of interest, remain stationary throughout the entire data acquisition period ([Hutchison et al., 2013](#)). However, substantial evidence suggests that functional connectivity can change within seconds ([Jiang et al., 2022](#)).

Unlike sFC approaches, dynamic functional connectivity (dFC) methods account for temporal variability within and between brain regions. These dFC methods enable analysis of dynamic functional states, characterized by consistent, replicable, and distinct periods of varying connectivity patterns ([Rabany et al., 2019](#)). Among these techniques, co-activation patterns (CAPs) analysis aggregates similar spatial distributions of brain activity using clustering techniques, typically k-means. This process not only clusters these patterns but also creates an average representation of brain states assigned to the same cluster (i.e., CAP), thereby capturing the dynamic nature of brain activity during the scanning session ([Liu et al., 2018](#)). Similar CAPs can be identified across both resting-state data (representing the brain's fundamental architecture that spontaneously fluctuates without external stimuli) and task data (representing functional coupling between brain regions in response to a specific stimuli or task) ([Kupis et al., 2020](#)).

Statement of Need

Currently, performing an end-to-end CAPs analysis presents challenges due to the numerous steps required. Researchers must:

1. clean timeseries data through nuisance regression and censor frames with high framewise displacement (excessive head motion).
2. perform spatial dimensionality reduction.
3. concatenate timeseries data across multiple subjects for k-means clustering.

- 38 4. select an optimal cluster size using heuristics such as the elbow or silhouette methods.
- 39 5. implement various visualization techniques to enhance interpretability of the results.

40 While other excellent CAPs toolboxes exist, they are often implemented in proprietary languages
 41 such as MATLAB (which is the case for TbCAPs (Bolton et al., 2020)), lack comprehensive end-to-
 42 end analytical pipelines for both resting-state and task-based fMRI data with temporal dynamic
 43 metrics and visualization capabilities (such as capcalc (Frederick & Drucker, 2022)), or are
 44 generalized toolboxes for assessing similarity between different dFC methods (such as PydFC
 45 (Torabi et al., 2024)). NeuroCAPs addresses these limitations by providing an accessible Python
 46 package that leverages object-oriented programming design to create a flexible pipeline-like
 47 architecture specifically for performing CAPs analyses. Furthermore, NeuroCAPs offers the
 48 ability to directly save certain outputs, such as the extracted timeseries, and supports continuing
 49 work across sessions through pickle serialization of Python objects, allowing an analysis to be
 50 completed incrementally. The package assumes fMRI data is organized in a BIDS-compliant
 51 directory and is optimized for data preprocessed with fMRIPrep (Esteban et al., 2019), a
 52 preprocessing workflow designed to minimize manual user input and enhance reproducibility.
 53 However, preprocessed fMRI data organized in similar directory formats as fMRIPrep outputs
 54 can still leverage NeuroCAPs' timeseries extraction functionalities. In addition, the necessary
 55 data structure (a dictionary mapping subject IDs to run IDs and its associated timeseries data)
 56 can be manually created to leverage many of NeuroCAPs' analytical capabilities.

57 Modules

58 The package consists of four modules, with core functionality primarily distributed between
 59 two main modules (`neurocaps.extraction` and `neurocaps.analysis`) that handle the entire
 60 workflow from postprocessing to visualization, significantly streamlining the CAPs analysis
 61 process.

62 `neurocaps.exceptions`

63 This module contains custom exceptions, one of which is `BIDSQueryError`. Since NeuroCAPs
 64 utilizes PyBIDS (Yarkoni et al., 2019), a Python package for querying BIDS-compliant
 65 directories, this exception was created to guide users and provide potential fixes when no subject
 66 IDs are detected in the specified BIDS directories. The other exception, `NoElbowDetected`,
 67 was created to provide potential solutions in the event that the elbow method (implemented
 68 by `KneeLocator` from the `Kneed` package (Arvai, 2023)) could not identify the optimal cluster
 69 size for k-means.

70 `neurocaps.extraction`

71 This module contains the `TimeseriesExtractor` class, which:

- 72 ■ extracts both resting-state and task-based functional MRI data using lateralized brain
 73 parcellations (such as the Schaefer (Schaefer et al., 2018), Automated Anatomical
 74 Labeling (Tzourio-Mazoyer et al., 2002), and Human Connectome Project extended
 75 (Huang et al., 2022) parcellations) for spatial dimensionality reduction.
- 76 ■ leverages Nilearn's (contributors, n.d.) `NiftiLabelsMasker` to perform nuisance regres-
 77 sion, censors high-motion volumes using fMRIPrep-derived regressors, and stores the
 78 extracted timeseries information in a dictionary mapping subject IDs to run IDs and their
 79 associated timeseries.
- 80 ■ saves extracted timeseries data in a serialized pickle format.
- 81 ■ reports the number of censored frames, interpolated frames, and mean and standard
 82 deviation of continuous high motion segments for each subject.
- 83 ■ visualizes timeseries data for a specific subject's run.

84 `neurocaps.analysis`

85 This module contains the `CAP` class, which:

- allows group-specific analyses or analyses on all subjects (the default configuration).
- performs k-means clustering (from Scikit-learn (Pedregosa et al., 2011)) for CAP identification, while supporting a single cluster size or a range of clusters in combination with various cluster selection methods to determine the optimal cluster size, including:
 - the elbow method (from Kneed package (Arvai, 2023))
 - the silhouette score, davies-bouldin index, and variance ratio methods (all from Scikit-learn (Pedregosa et al., 2011))
- computes various temporal dynamics metrics (including counts/state initiation, temporal fraction, persistence, transition frequency, and transition probability) at the subject-level and exports data to a CSV file for downstream statistical analyses.
- enables conversion of CAPs to NIFTI statistical maps.
- provides multiple visualization options for CAPs including heatmaps, outer products, correlation matrices, and cosine similarity radar plots (showing network correspondence to both positive and negative CAP activations).

Additionally, the module provides standalone functions for:

- changing the data type and performing additional standardization of timeseries data produced by TimeseriesExtractor.
- merging multiple timeseries data across different dictionaries produced by TimeseriesExtractor by identifying similar subjects and concatenating their data, which facilitates analyses to identify CAPs across sessions or different tasks.
- creating averaged transition probability matrices from subject-level transition probabilities.

neurocaps.typing

This module contains custom type definitions that can be used with static type checkers to build appropriate dictionary structures for specifying parcellations or the timeseries data if these structures must be manually created.

Documentation

Comprehensive documentation for NeuroCAPs, related to installation, usage, and the application programming interface (API) can be found at neurocaps.readthedocs.io (D. Smith, 2025b), with Jupyter Notebook demonstrations available on its Github repository (D. Smith, 2025c).

Example Application

NeuroCAPs was originally developed (and later refined for broader use) to facilitate the analysis in D. D. Smith et al. (2025). In this manuscript that was submitted for publication, NeuroCAPs was used to extract timeseries data, cluster and identify CAPs using the elbow method, and produce visualizations for CAPs (i.e., heatmap, surface plots, correlation matrix, and cosine similarity plots)

Acknowledgements

Funding provided by the Dissertation Year Fellowship (DYF) Program at Florida International University (FIU), assisted in further refinement of the NeuroCAPs package.

References

- Arvai, K. (2023). *Kneed*. Zenodo. <https://doi.org/10.5281/ZENODO.8127224>

- 126 Bolton, T. A. W., Tuleasca, C., Wotruba, D., Rey, G., Dhanis, H., Gauthier, B., Delavari,
127 F., Morgenroth, E., Gaviria, J., Blondiaux, E., Smigielski, L., & Van De Ville, D. (2020).
128 TbCAPs: A toolbox for co-activation pattern analysis. *NeuroImage*, 211, 116621. <https://doi.org/10.1016/j.neuroimage.2020.116621>
129
- 130 contributors, N. (n.d.). *nilearn*. <https://doi.org/https://doi.org/10.5281/zenodo.8397156>
- 131 Esteban, O., Markiewicz, C. J., Blair, R. W., Moodie, C. A., Isik, A. I., Erramuzpe, A., Kent,
132 J. D., Goncalves, M., DuPre, E., Snyder, M., Oya, H., Ghosh, S. S., Wright, J., Durnez,
133 J., Poldrack, R. A., & Gorgolewski, K. J. (2019). fMRIPrep: A robust preprocessing
134 pipeline for functional MRI. *Nature Methods*, 16(1), 111–116. <https://doi.org/10.1038/s41592-018-0235-4>
135
- 136 Frederick, B. D., & Drucker, D. M. (2022). *Bbfrederick/capcalc: Version 1.2.2.2 - 8/30/22*
137 *deployment bug fix*. Zenodo. <https://doi.org/10.5281/ZENODO.7035806>
- 138 Huang, C.-C., Rolls, E. T., Feng, J., & Lin, C.-P. (2022). An extended human connectome
139 project multimodal parcellation atlas of the human cortex and subcortical areas. *Brain*
140 *Structure and Function*, 227(3), 763–778. <https://doi.org/10.1007/s00429-021-02421-6>
- 141 Hutchison, R. M., Womelsdorf, T., Allen, E. A., Bandettini, P. A., Calhoun, V. D., Corbetta,
142 M., Della Penna, S., Duyn, J. H., Glover, G. H., Gonzalez-Castillo, J., Handwerker, D. A.,
143 Keilholz, S., Kiviniemi, V., Leopold, D. A., De Pasquale, F., Sporns, O., Walter, M., &
144 Chang, C. (2013). Dynamic functional connectivity: Promise, issues, and interpretations.
145 *NeuroImage*, 80, 360–378. <https://doi.org/10.1016/j.neuroimage.2013.05.079>
- 146 Jiang, F., Jin, H., Gao, Y., Xie, X., Cummings, J., Raj, A., & Nagarajan, S. (2022). Time-
147 varying dynamic network model for dynamic resting state functional connectivity in fMRI
148 and MEG imaging. *NeuroImage*, 254, 119131. <https://doi.org/10.1016/j.neuroimage.2022.119131>
149
- 150 Kupis, L., Romero, C., Dirks, B., Hoang, S., Parladé, M. V., Beaumont, A. L., Cardona, S.
151 M., Alessandri, M., Chang, C., Nomi, J. S., & Uddin, L. Q. (2020). Evoked and intrinsic
152 brain network dynamics in children with autism spectrum disorder. *NeuroImage: Clinical*,
153 28, 102396. <https://doi.org/10.1016/j.nicl.2020.102396>
- 154 Liu, X., Zhang, N., Chang, C., & Duyn, J. H. (2018). Co-activation patterns in resting-state
155 fMRI signals. *NeuroImage*, 180, 485–494. <https://doi.org/10.1016/j.neuroimage.2018.01.041>
156
- 157 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M.,
158 Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D.,
159 Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python.
160 *Journal of Machine Learning Research*, 12, 2825–2830.
- 161 Rabany, L., Brocke, S., Calhoun, V. D., Pittman, B., Corbera, S., Wexler, B. E., Bell, M. D.,
162 Pelphrey, K., Pearlson, G. D., & Assaf, M. (2019). Dynamic functional connectivity in
163 schizophrenia and autism spectrum disorder: Convergence, divergence and classification.
164 *NeuroImage: Clinical*, 24, 101966. <https://doi.org/10.1016/j.nicl.2019.101966>
- 165 Schaefer, A., Kong, R., Gordon, E. M., Laumann, T. O., Zuo, X.-N., Holmes, A. J., Eickhoff,
166 S. B., & Yeo, B. T. T. (2018). Local-global parcellation of the human cerebral cortex
167 from intrinsic functional connectivity MRI. *Cerebral Cortex*, 28(9), 3095–3114. <https://doi.org/10.1093/cercor/bhx179>
168
- 169 Smith, D. (2025a). *NeuroCAPs: A python package for performing co-activation patterns*
170 *analyses on resting-state and task-based fMRI data: archive*. Zenodo. <https://doi.org/10.5281/zenodo.15111381>
171
- 172 Smith, D. (2025b). *NeuroCAPs. Documentation*. <https://neurocaps.readthedocs.io/>.
- 173 Smith, D. (2025c). *NeuroCAPs. Github repository*. <https://github.com/donishadsmith/>

174 [neurocaps](#)

175 Smith, D. D., Bartley, J. E., Peraza, J. A., Bottenhorn, K. L., Nomi, J. S., Uddin, L. Q., Riedel,
176 M. C., Salo, T., Laird, R. W., Pruden, S. M., Sutherland, M. T., Brewe, E., & Laird, A. R.
177 (2025). *Dynamic reconfiguration of brain coactivation states associated with active and*
178 *lecture-based learning of university physics*. <https://doi.org/10.1101/2025.02.22.639361>

179 Torabi, M., Mitsis, G. D., & Poline, J.-B. (2024). On the variability of dynamic functional
180 connectivity assessment methods. *GigaScience*, 13, giae009. [https://doi.org/10.1093/](https://doi.org/10.1093/gigascience/giae009)
181 [gigascience/giae009](https://doi.org/10.1093/gigascience/giae009)

182 Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix,
183 N., Mazoyer, B., & Joliot, M. (2002). Automated anatomical labeling of activations in
184 SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain.
185 *NeuroImage*, 15(1), 273–289. <https://doi.org/10.1006/nimg.2001.0978>

186 Yarkoni, T., Markiewicz, C., De La Vega, A., Gorgolewski, K., Salo, T., Halchenko, Y.,
187 McNamara, Q., DeStasio, K., Poline, J.-B., Petrov, D., Hayot-Sasson, V., Nielson, D.,
188 Carlin, J., Kiar, G., Whitaker, K., DuPre, E., Wagner, A., Tirrell, L., Jas, M., ... Blair, R.
189 (2019). PyBIDS: Python tools for BIDS datasets. *Journal of Open Source Software*, 4(40),
190 1294. <https://doi.org/10.21105/joss.01294>

DRAFT