The Journal of Open Source Software

# pytau: A Python package for streamlined changepoint model analysis in neuroscience

**Abuzar Mahmood** [1,2]

**1** Swartz Foundation Computational Neuroscience Fellow, Volen Center for Complex Systems, Brandeis University, Waltham, MA, USA **2** Department of Psychology, Brandeis University, Waltham, MA, United States of America

**DOI:** 10.21105/joss.08509

**Software**

- Review
- Repository
- Archive

**Editor:** Frederick Boehm

**Reviewers:**

- @HectorMozo3110
- @duodenum96

**Submitted:** 08 April 2025
**Published:** 12 January 2026



## Summary

Neural activity often exhibits sharp transitions between distinct states, captured by changepoint models (Giahi Saravani et al., 2019; Jones et al., 2007; Seidemann et al., 1996). pytau is a Python package for batched Bayesian changepoint inference across parameter grids and datasets. It integrates with PyMC3 to provide uncertainty estimates that are critical for noisy neuroscience data with small sample sizes and low channel counts, and includes tools for preprocessing, model fitting, result visualization, and statistical analysis. The package has been successfully applied in published research examining taste processing (Baas-Thomas et al., 2025; Abuzar Mahmood et al., 2023; A. Mahmood et al., 2025; Maigler et al., 2024) and taste aversion learning (Flores & Lin, 2023), and is currently being utilized in several ongoing neuroscience studies (Calia-Bogan et al., 2025; Mazzio et al., 2025; Raymond et al., 2025).

While pytau is specialized for neural data analysis, the underlying Bayesian changepoint detection methods have broad applicability to any time series data where identifying state transitions is important. The package includes examples demonstrating its use on classic changepoint datasets from other domains, including historical event count data (coal mining disasters) and continuous measurements (temperature data), illustrating how the same framework can be applied to economic time series, environmental monitoring, quality control, and other sequential data analysis problems.

## Statement of need

Understanding how neural populations encode information often involves analyzing activity changes over time across different experimental conditions, parameters, or subjects. Fitting and comparing Bayesian changepoint models across numerous datasets or parameter settings is computationally intensive and logistically challenging. Existing changepoint detection tools lack the specialized functionality needed for neuroscience applications, particularly for handling multi-trial, multi-neuron spike train data.

pytau addresses this gap by providing a modularized pipeline specifically designed for neuroscience data. The package offers several key advantages:

1. **Batch processing**: Automates model fitting across multiple datasets and parameter configurations
2. **Database management**: Organizes and tracks model fits for easy retrieval and comparison
3. **Visualization tools**: Provides specialized plotting functions including raster plots with overlaid changepoints, state-dependent firing rate visualizations, and transition-aligned activity plots
4. **Statistical analysis**: Includes tools for significance testing of state-dependent neural activity, such as ANOVA-based detection of neurons with significant state-dependent

firing and pairwise t-tests for transition-triggered activity

Its adoption in studies of taste processing (Abuzar Mahmood et al., 2023; A. Mahmood et al., 2025; Maigler et al., 2024), taste aversion learning (Flores & Lin, 2023), and ingestive behavior (Baas-Thomas et al., 2025) demonstrates its practical utility for researchers studying state transitions in neural activity.

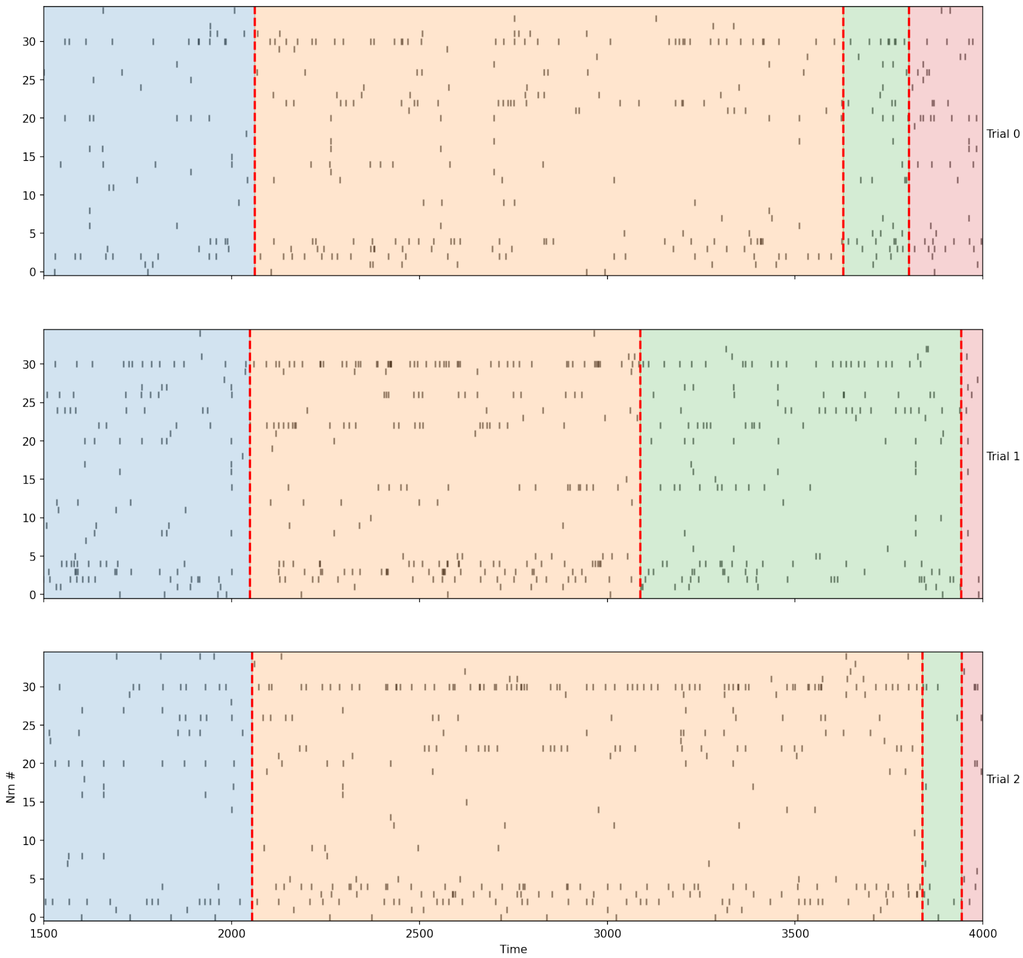


**Figure 1: Spike rasters with changepoint overlays** visualize inferred changepoints across trials and neurons.

## Implementation and architecture

pytau is implemented in Python and built on NumPy, SciPy, PyMC3, and Matplotlib (Harris et al., 2020; Hunter, 2007; Salvatier et al., 2016; Virtanen et al., 2020). The package is organized into several modules:

1. **changepoint_model.py**: Contains model definitions for various changepoint models including Poisson and Gaussian models (see Available Models)
2. **changepoint_io.py**: Handles data loading, preprocessing, and result storage through `FitHandler` and `DatabaseHandler` classes
3. **changepoint_analysis.py**: Provides tools for analyzing fitted models, including significance testing and visualization
4. **changepoint_preprocess.py**: Contains functions for data preprocessing, binning, and transformations (see Data Formats)

The Journal of Open Source Software

The mathematical foundation is Bayesian changepoint detection. For a time series $X = \{x_1, x_2, ..., x_T\}$, the package models $K$ states with transitions at $\tau = \{\tau_1, \tau_2, ..., \tau_{K-1}\}$ and Poisson emissions: $x_t \sim \text{Poisson}(\lambda_k)$ for $\tau_{k-1} < t \leq \tau_k$, where $\lambda_k$ is the firing rate in state $k$.

The package employs Automatic Differentiation Variational Inference (ADVI) (Kucukelbir et al., 2017) for fast posterior approximation and Markov Chain Monte Carlo (MCMC) with the No-U-Turn Sampler (NUTS) (Hoffman & Gelman, 2014) for precise inference (see Inference Methods). A key feature is modeling state transitions with sigmoid functions, which enables continuous parameter exploration and detection of gradual changes in neural dynamics.

## Example usage

pytau provides a streamlined workflow through the `FitHandler` class for model fitting and `PklHandler` for analysis:

```python
from pytau.changepoint_io import FitHandler

# Initialize fit handler
fh = FitHandler(data_dir='/path/to/data', taste_num=1,
                region_name='GC', experiment_name='example')

# Set preprocessing and model parameters
fh.set_preprocess_params(time_lims=[0, 2000], bin_width=10)
fh.set_model_params(states=3, fit=5000, samples=1000)

# Run the full pipeline
fh.load_spike_trains()
fh.preprocess_data()
fh.create_model()
fh.run_inference()
fh.save_fit_output()
```

After fitting, results can be analyzed using the `PklHandler` class:

```python
from pytau.changepoint_analysis import PklHandler

# Load fitted model
pkl_handler = PklHandler('/path/to/saved/model.pkl')

# Access model components
tau = pkl_handler.tau  # Changepoint times
firing = pkl_handler.firing  # Firing rate analysis

# Analyze significant neurons
significant_neurons = firing.anova_significant_neurons
```

The package includes visualization tools for examining neural activity with overlaid changepoints, analyzing state-dependent firing rates, and visualizing transition-aligned activity. Comprehensive tutorials and detailed examples are available in the documentation, including Jupyter notebooks and example scripts with test datasets in the repository's `how_to` directory.

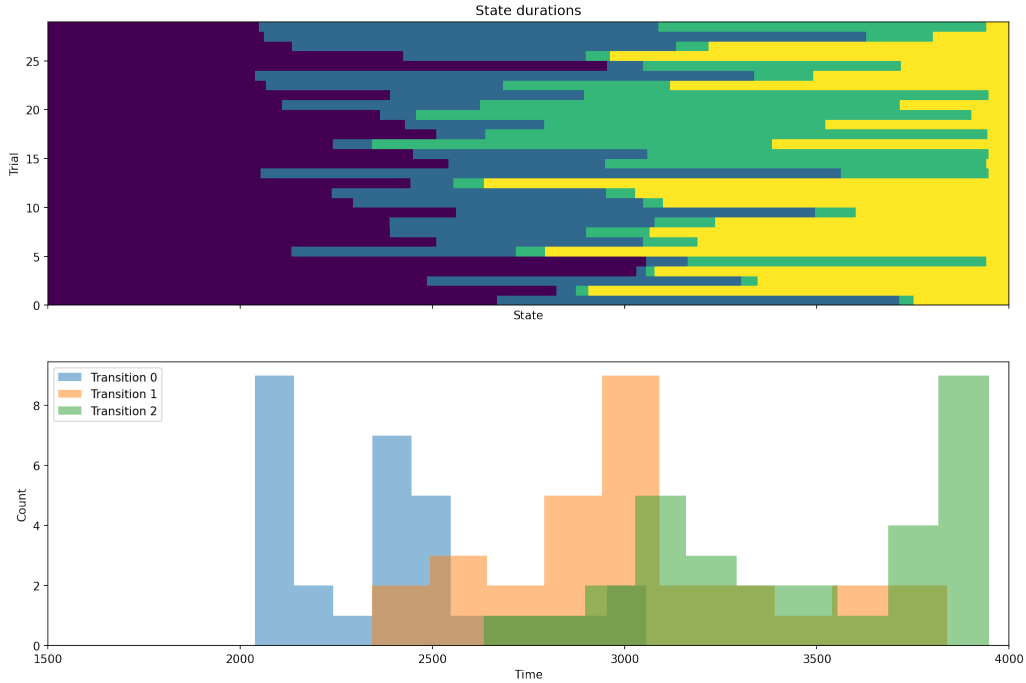


**Figure 2: State timing overview** shows state durations and transition time distributions across trials.

## State of the field

As discussed above, several tools exist for changepoint detection, including `ruptures` (Truong et al., 2018) for offline change point detection, `bayesloop` (Mark et al., 2018) for probabilistic time series analysis, `PyChange` (Gumbsch, 2017) for general time series changepoint detection, and Bayesian online changepoint detection methods (Adams & MacKay, 2007; Fearnhead & Liu, 2007). While these general-purpose tools are valuable, `pytau` differs by focusing specifically on neuroscience applications. It provides specialized functionality for handling multi-trial, multi-neuron spike train data, batch processing across parameter grids, database management for model comparison, and neuroscience-specific visualization and statistical analysis tools. This specialization makes `pytau` particularly suited for researchers analyzing state transitions in neural population activity across different experimental paradigms.

## Acknowledgements

We acknowledge support from the Katz Lab and the PyMC development team.

## References

Adams, R. P., & MacKay, D. J. (2007). Bayesian online changepoint detection. *arXiv Preprint arXiv:0710.3742*. https://doi.org/10.48550/arXiv.0710.3742

Baas-Thomas, N., Mahmood, A., & Katz, D. B. (2025). The ingestive response reflects neural dynamics in gustatory cortex. *bioRxiv*. https://doi.org/10.1101/2025.10.01.679845

Calia-Bogan, V., Mahmood, A., Steindler, J. R., & Katz, D. B. (2025). *Taste-evoked intra-state dynamics in the gustatory cortex (in preparation)*.

Fearnhead, P., & Liu, Z. (2007). On-line inference for multiple changepoint problems. *Journal* *of the Royal Statistical Society: Series B (Statistical Methodology)*, *69*(4), 589–605. https://doi.org/10.1111/j.1467-9868.2007.00601.x

Flores, V. L., & Lin, J.-Y. (2023). Benign taste experience enhances cortical response reliability during aversion conditioning towards a novel taste. *bioRxiv*. https://doi.org/10.1101/2023.12.19.572413

Giahi Saravani, A., Forseth, K. J., Tandon, N., & Pitkow, X. (2019). Dynamic brain interactions during picture naming. *Eneuro*, *6*(4). https://doi.org/10.1523/eneuro.0472-18.2019

Gumbsch, T. (2017). *PyChange: Multiple change detection with Python*. https://github.com/BorgwardtLab/PyChange

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*, 1593–1623. https://doi.org/10.48550/arXiv.1111.4246

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jones, L. M., Fontanini, A., Sadacca, B. F., Miller, P., & Katz, D. B. (2007). Natural stimuli evoke dynamic sequences of states in sensory cortical ensembles. *Proceedings of the National Academy of Sciences*, *104*(47), 18772–18777. https://doi.org/10.1073/pnas.0705546104

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., & Blei, D. M. (2017). Automatic differentiation variational inference. *Journal of Machine Learning Research*, *18*(14), 1–45. https://doi.org/10.48550/arXiv.1603.00788

Mahmood, A., Steindler, J. R., & Katz, D. B. (2025). Sensory and palatability coding of taste stimuli in cortex involves dynamic and asymmetric cortico-amygdalar interactions. *bioRxiv*. https://doi.org/10.1101/2025.07.01.662567

Mahmood, Abuzar, Steindler, J., Germaine, H., Miller, P., & Katz, D. B. (2023). Coupled dynamics of stimulus-evoked gustatory cortical and basolateral amygdalar activity. *Journal of Neuroscience*, *43*(3), 386–404. https://doi.org/10.1523/JNEUROSCI.1412-22.2022

Maigler, K. C., Mahmood, A., Osman, B. T., & Katz, D. B. (2024). Individual taste preferences predict cortical taste dynamics but are modified by experience. *bioRxiv*. https://doi.org/10.1101/2024.04.02.587752

Mark, C., Metzner, C., Lautscham, L., Strissel, P. L., Strick, R., & Fabry, B. (2018). Bayesian model selection for complex dynamic systems. *Nature Communications*, *9*(1), 1803. https://doi.org/10.1038/s41467-018-04241-5

Mazzio, C., Germaine, H., Lin, J.-Y., & Katz, D. B. (2025). *Cortical dynamics underlying learned and non-learned aversive behavior (in preparation)*.

Raymond, M., Lin, J.-Y., & Katz, D. B. (2025). *A comparison of cortical taste representations arising from free licking and intraoral delivery (in preparation)*.

Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, *2*, e55. https://doi.org/10.7717/peerj-cs.55

Seidemann, E., Meilijson, I., Abeles, M., Bergman, H., & Vaadia, E. (1996). Simultaneously recorded single units in the frontal cortex go through sequences of discrete and stable states in monkeys performing a delayed localization task. *The Journal of Neuroscience*, *16*(2), 752–768. https://doi.org/10.1523/jneurosci.16-02-00752.1996

Truong, C., Oudre, L., & Vayatis, N. (2018). Ruptures: Change point detection in Python. *arXiv Preprint arXiv:1801.00826*. https://doi.org/10.48550/arXiv.1801.00826

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2