Modelling M/EEG data with Bayesian generalised additive multilevel models

Ladislas Nalborczyk[1] and Paul Bürkner[2]
[1]Aix Marseille Univ, CNRS, LPL
[2]TU Dortmund University, Department of Statistics

Abstract

Time-resolved electrophysiological measurements such as those offered by magneto- or electro-encephalography (M/EEG) provide a unique window onto neural activity underlying cognitive process and how they unfold over time. Typically, we are interested in testing whether such measures differ across conditions and/or groups. The conventional approach consists in conducting mass-univariate statistics through followed by some form of multiplicity correction (e.g., FDR, FWER) or cluster-based inference. However, these cluster-based methods have an important downside: they shift the focus of inference from the timepoint to the cluster level, thus preventing any conclusion to be made about the onset and offset of effects (e.g., differences across conditions). Here, we introduce a novel *model-based approch* for analysing one-dimensional M/EEG timeseries such as ERPs or decoding timecourses and their differences across conditions or group. This approach relies on Bayesian generalised additive multilevel models, which output the posterior probabilility of the effect being above 0 (or above chance) at every timestep, while naturally taking into account the temporal dependencies and between-subject variability present in such data.

Keywords: EEG, MEG, generalised additive models, mixed-effects models, multilevel models, Bayesian statistics, brms

Table of contents

Ladislas Nalborczyk https://orcid.org/0000-0002-7419-9855
Paul Bürkner https://orcid.org/0000-0001-5765-8995
The authors have no conflicts of interest to disclose.
Correspondence concerning this article should be addressed to Ladislas Nalborczyk, Aix Marseille Univ, CNRS, LPL, 5 avenue Pasteur, 13100 Aix-en-Provence, France, email: ladislas.nalborczyk@cnrs.fr

Modelling M/EEG data with Bayesian generalised additive multilevel models

Introduction

Here are some useful references to be discussed (Combrisson & Jerbi, 2015; Ehinger & Dimigen, 2019; Frossard & Renaud, 2021, 2022; Gramfort, 2013; Hayasaka, 2003; Luck & Gaspelin, 2017; Maris & Oostenveld, 2007; E. J. Pedersen et al., 2019; C. R. Pernet et al., 2015)... See also (Maris, 2011)... and (Rosenblatt et al., 2018) (history of cluster-based approaches and using a data split?)... Cluster failure (Eklund et al., 2016)...

Previous modelling work

Recent example of GLM for EEG (Fischer & Ullsperger, 2013; Wüllhorst et al., 2025)... "Similar approaches have been successfully applied to EEG time- (Rousselet et al., 2008) and frequency-domain (Cohen and Cavanagh, 2011) data and allow the simultaneous investigation of multiple independent variables while preserving the high temporal resolution of the EEG."... See also (Hauk et al., 2006; Rousselet et al., 2008)... Example of two-stage regression analysis (i.e., individual-level then group-level, Dunagan et al., 2024)...

From Dimigen & Ehinger (2021): "Recently, spline regression has been applied to ERPs (Hendrix, Baayen, & Bolger, 2017; Kryuchkova,Tucker, Wurm, & Baayen,2012; Tremblay & Baayen, 2010; Tremblay & Newman, 2015)"... GAMMs for EEG data (Abugaber et al., 2023; Meulman et al., 2015)...

Disentangling overlapping processes (Skukies et al., 2024; Skukies & Ehinger, 2021)... Weighting single trials (C. Pernet, 2022)... The LIMO toolbox (https://onlinelibrary.wiley.com/doi/10.1155/2011/831409)... Using Bayes factors (Teichmann, 2022)...

Generalised additive models

See for instance these tutorials (Sóskuthy, 2017; Winter & Wieling, 2016) or application to phonetic data (Wieling, 2018) or this introduction (Baayen & Linke, 2020) or these reference books (Hastie & Tibshirani, 2017; Wood, 2017)...

In generalised additive models (GAMs), the functional relationship between predictors and response variable is decomposed into a sum of low-dimensional non-parametric functions. A typical GAM has the following form:

$$y_i \sim \text{EF}\left(\mu_i, \phi\right)$$
$$g\left(\mu_i\right) = A_i + \mathbf{X}_i\gamma + \sum_{j=1}^{J} f_j\left(x_{ij}\right)$$

where $y_i \sim \text{EF}\left(\mu_i, \phi\right)$ denotes that the observations y_i are distributed as some member of the exponential family of distributions (e.g., Gaussian, Gamma, Beta, Poisson) with mean μ_i and scale parameter ϕ; $g(\cdot)$ is the link function, A is an offset, $\mathbf{X}_i$ is the ith row of a parametric model matrix, γ is a vector of parameters for the parametric terms, f_j is a smooth function of covariate x_j. The smooth functions f_j are represented in the model via penalised splines basis expansions of the covariates, that are a weighted sum of basis functions:

$$f_j\left(x_{ij}\right) = \sum_{k=1}^{K} \beta_{jk} b_{jk}\left(x_{ij}\right)$$

where β_{jk} is the weight (coefficient) associated with the kth basis function $b_{jk}()$ evaluated at the covariate value x_{ij} for the jth smooth function f_j. Splines' coefficients are penalised...

Objectives

Focusing on identifying onset and offset of effects (as assessed by ERP differences or decoding performance)... Assessing the performance of a model-based approach (i.e., Bayesian GAMMs) to conventional methods (multiplicity corrections or cluster-based permutation)...

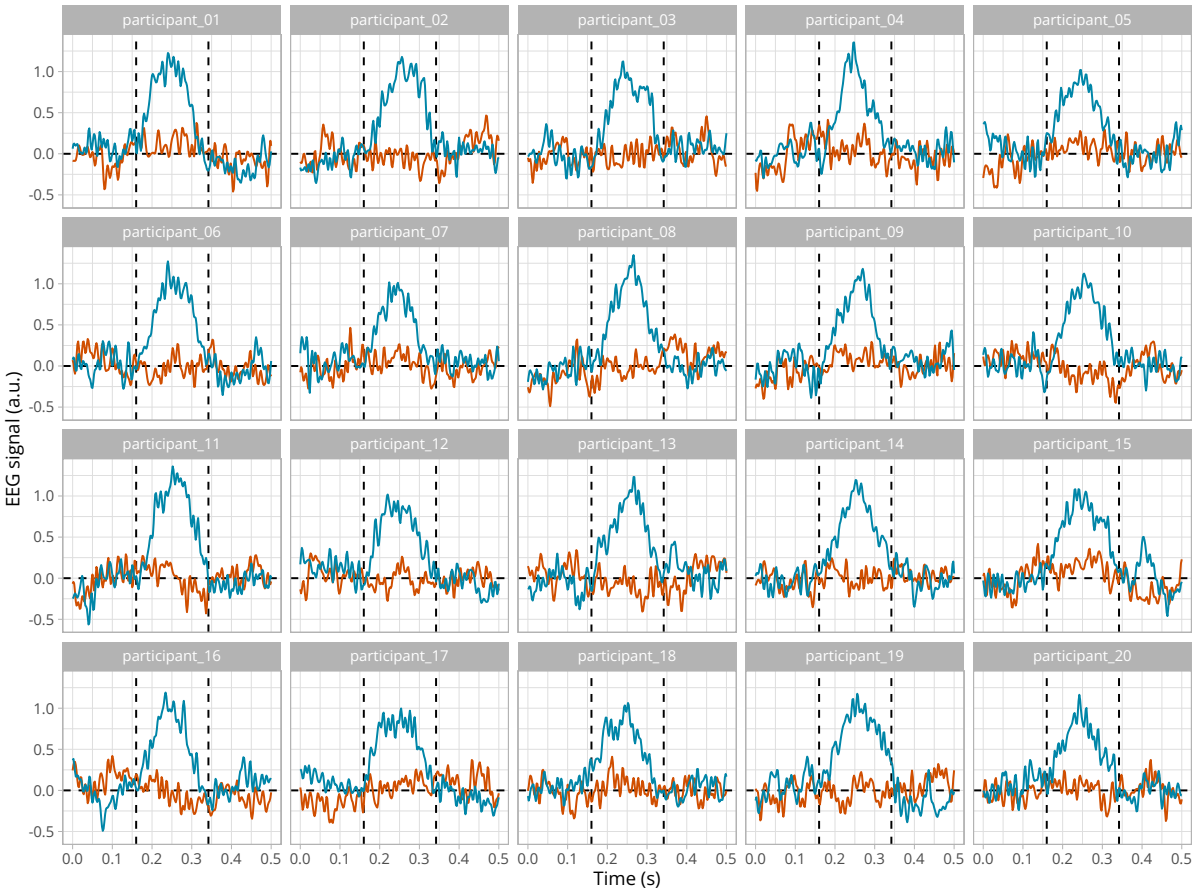
Methods

M/EEG data simulation

Following the approach used by Sassenhagen & Draschkow (2019) and Rousselet (2025), we simulated EEG data stemming from two conditions, one with noise only, and the other with noise + signal. As in previous studies, the noise was generated by superimposing 50 sinusoids at different frequencies, following an EEG-like spectrum (see details and code in Yeung et al., 2004). As in Rousselet (2025), the signal was generated from truncated Gaussian with an objective onset at 160 ms, a peak at 250 ms, and an offset at 342 ms. We simulated this signal for 250 timesteps between 0 and 0.5s, akin to a 500 Hz sampling rate. We simulated such data for a group of 20 participants with 50 trials per participant and condition (Figure 1).

Figure 1

Some ERPs in two conditions with 50 trials each, for a group of 20 participants.



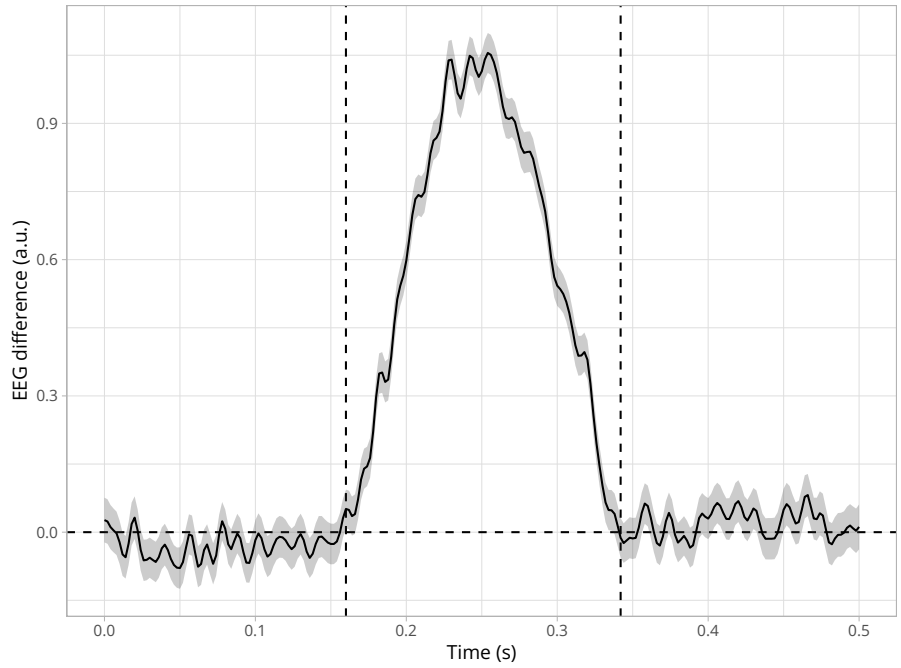
We computed the average of the ERP difference (Figure 2)...

Model fitting

We then fitted a Bayesian GAM using the `brms` package (Bürkner, 2017, 2018; Nalborczyk et al., 2019). We used the default priors in `brms`, that is, weakly informative priors. We ran

Figure 2

Group-level average difference between conditions (mean +/- standard error of the mean). The 'true' onset and offset are indicated by the vertical dashed lines.



eight Markov Chain Monte-Carlo (MCMC) to approximate the posterior distribution, including each 5000 iterations and a warmup of 2000 iterations, yielding a total of $8 \times (5000 - 2000) = 24000$ posterior samples to use for inference. Posterior convergence was assessed examining trace plots as well as the Gelman–Rubin statistic $\hat{R}$.

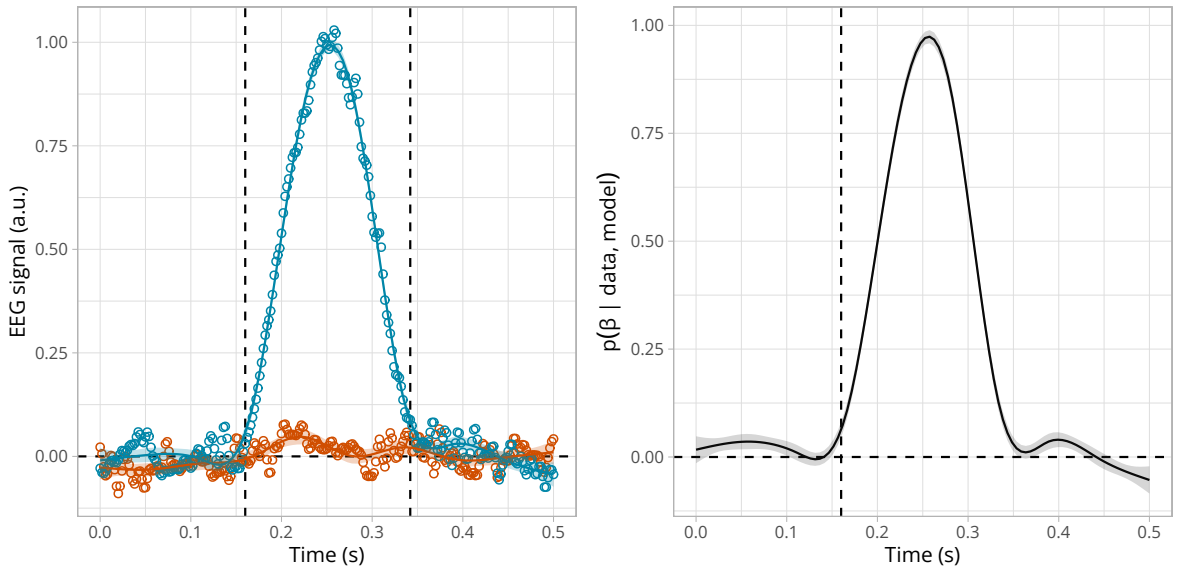
```
# averaging across participants
ppt_df <- raw_df %>%
    group_by(participant, condition, time) %>%
    summarise(eeg = mean(eeg) ) %>%
    ungroup()

# defining a contrast for condition
contrasts(ppt_df$condition) <- c(-0.5, 0.5)

# fitting the GAM
gam <- brm(
    # cubic regression splines with k-1 basis functions
    eeg ~ condition + s(time, bs = "cr", k = 10, by = condition),
    data = ppt_df,
    family = gaussian(),
    warmup = 2000,
    iter = 5000,
    chains = 8,
    cores = 8,
    file = "models/gam.rds"
    )
```

Posterior probability of difference above 0

We then plot the posterior predictions together with the posterior estimate of the slope for `condition` at each timestep (Figure 3).

Figure 3

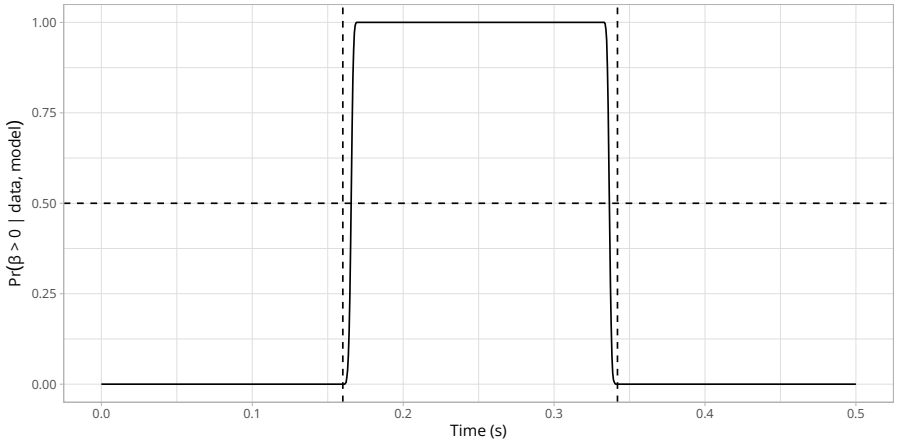
Posterior estimate of the ERP in each condition (left) or directly for the difference of ERPs (right) according to the GAM.



We then compute the posterior probability of the slope for `condition` being above $0 + \epsilon$ (Figure 4), with ϵ defined as 10% of the standard deviation of the raw EEG signal.

Figure 4

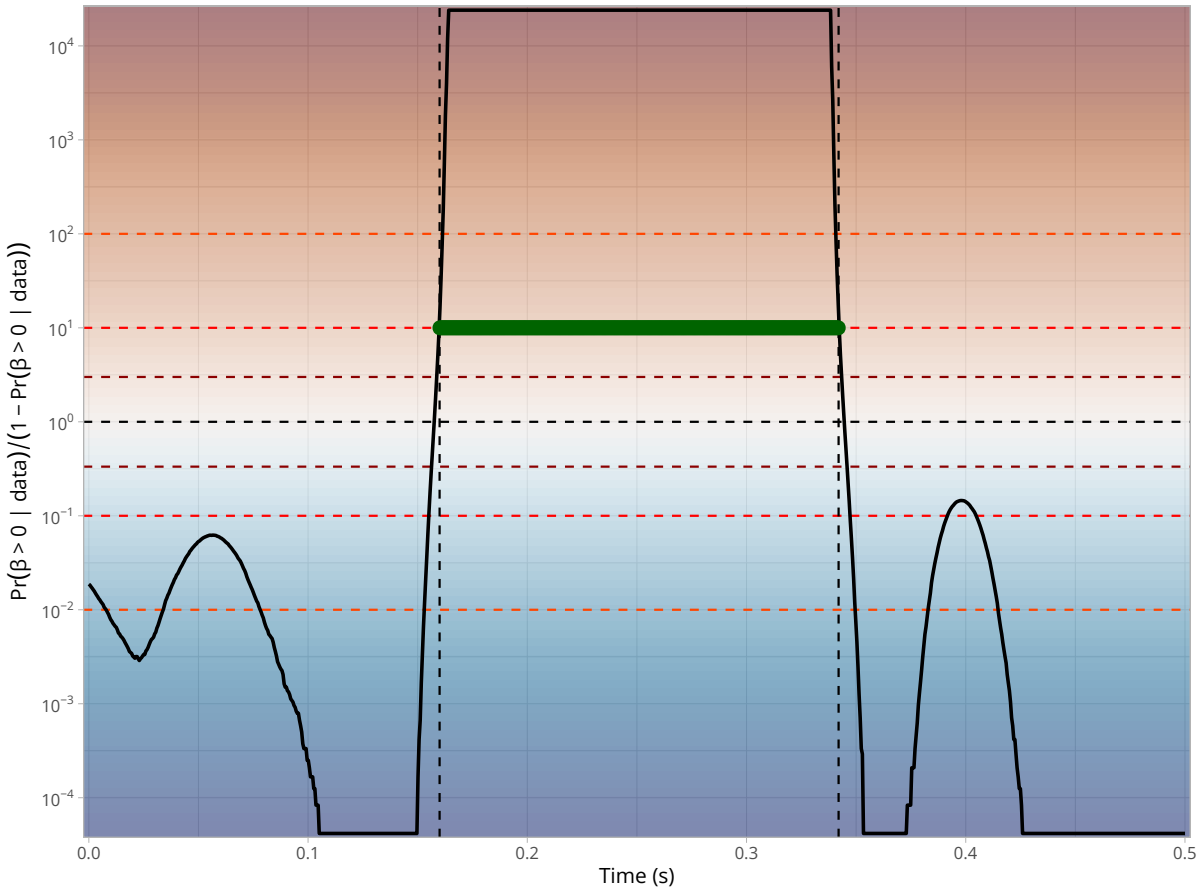
Posterior probability of the ERP difference (slope) being above 0 according to the GAM.



We can also express this as the ratio of posterior probabilities (i.e., $p/(1-p)$) and visualise the timecourse of this ratio superimposed with the conventional thresholds on evidence ratios (Figure 5).

Figure 5

Ratio of posterior probability according to the GAM. Timesteps above threshold (10) are highlighted in green.



Multilevel modelling using ERP summary statistics

Next we fit a hierarchical/multilevel GAM using summary statistics of ERPs (mean and SD) at the participant level (similar to what is done in meta-analysis).

```
# averaging across participants
summary_df <- raw_df %>%
    summarise(
        eeg_mean = mean(eeg),
        eeg_sd = sd(eeg),
        .by = c(participant, condition, time)
        )

# defining a contrast for condition
contrasts(summary_df$condition) <- c(-0.5, 0.5)

# fitting the GAM
meta_gam <- brm(
    # using by-participant SD of ERPs across trials
    eeg_mean | se(eeg_sd) ~
        condition + s(time, bs = "cr", k = 10, by = condition) +
        (1 | participant),
```

```
    data = summary_df,
    family = gaussian(),
    warmup = 2000,
    iter = 5000,
    chains = 8,
    cores = 8,
    file = "models/meta_gam.rds"
    )
```

Error properties of the proposed approach

We then computed the difference between the true and estimated onset/offset of the ERP difference (error := $|\hat{\theta} - \theta|$), according to various `eps` and `threshold` values. Remember that the signal is generated from a truncated Gaussian with an objective onset at 160 ms, a maximum at 250 ms, and an offset at 342 ms. Figure 6 shows that the hierarchical GAM can *exactly* recover the true onset and offset values, given some reasonable choice of `eps` and `threshold` values (e.g., a threshold of 20).

```
   eps threshold estimated_onset estimated_offset error_onset error_offset
1 0.00        13           0.16             0.34           0        0.002
2 0.00        14           0.16             0.34           0        0.002
3 0.00        15           0.16             0.34           0        0.002
4 0.00        16           0.16             0.34           0        0.002
5 0.00        17           0.16             0.34           0        0.002
6 0.01        10           0.16             0.34           0        0.002
```

Figure 6

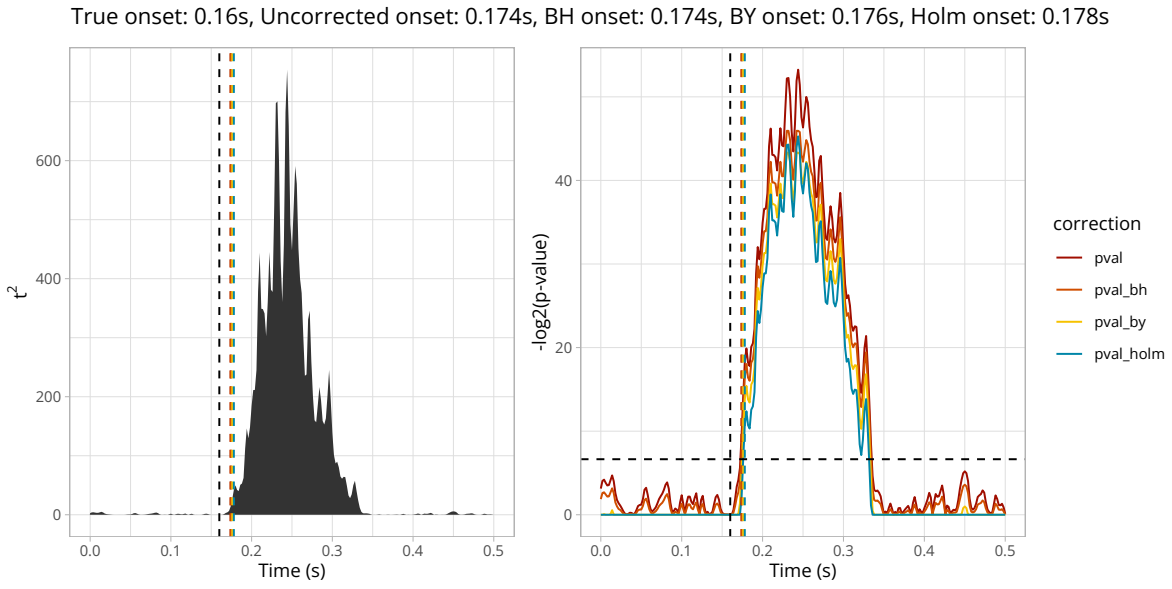
Error function of onset (left) and offset (right) estimation according to various eps and threshold values (according to the hierarchical GAM). Minimum error values are indicated by red crosses.

Comparing the identified onsets/offsets to other approaches

We compared the ability of the GAMM to correctly estimate the onset and offset of the ERP difference to widely used methods. First, we conducted mass-univariate t-tests (thus treating each timestep independently) and identified the onset and offset of the ERP difference as the first and last values crossing an arbitrary significance threshold ($\alpha = 0.01$). We then followed the same approach but after applying different forms of multiplicity correction the the p-values. We compared two methods that control the false discovery rate (FDR) (i.e., BH95, Benjamini & Hochberg, 1995; and BY01, Benjamini & Yekutieli, 2001), one method that controls the familywise error rate (FWER) (i.e., Holm–Bonferroni method, https://www.jstor.org/stable/4615733), and two cluster-based methods (permutation with a single threshold and TFCE, Smith & Nichols, 2009). The BH95, BY01, and Holm corrections were applied to the p-values using the `p.adjust()` function in `R`. The cluster-based inference was implemented using a cluster-sum statistic of squared t-values, as implemented in `MNE-Python` (Gramfort, 2013), called via the `R` package `reticulate` v 1.35.0 (Ushey et al., 2024). We also compared these estimates to the onset and offset as estimated using the binary segmentation algorithm, as implemented in the `R` package `changepoint` v 2.2.4 (Killick et al., 2022a), and applied directly to the squared t-values (as in Rousselet, 2025). For visualisation and interpretability purposes, we converted p-values to s-values, which can be interpreted as bits of surprising information, assuming a null effect (Greenland, 2019) (Figure 7).

Figure 7

Timecourse of squared t-values and s-values, with true onset (black dashed line) and onsets identified using the raw (uncorrected) p-values or the corrected p-values (BH, BY, Holm).



Simulation study

Onset/offset estimation methods were assessed using these summary statistics of the Monte Carlo sampling distributions of onsets: median absolute error (MAE) and variance, with 10.000 iterations (i.e., 10.000 simulated datasets)...

Application to actual MEG data

Assessing the reliability of the proposed approach using some sort of split-half reliability (e.g., Rosenblatt et al., 2018)? Using the MEG decoding results of Nalborczyk et al. (in preparation) in 33 participants:

- Create many (e.g., 10 or 20) half train/test splits of the data
- For each fold, estimate the onset/offset on both splits using all methods
- Then summarise the distribution of between-split difference across folds

This will allow checking that the proposed approach produces more reliable onset/offset estimates.

Results

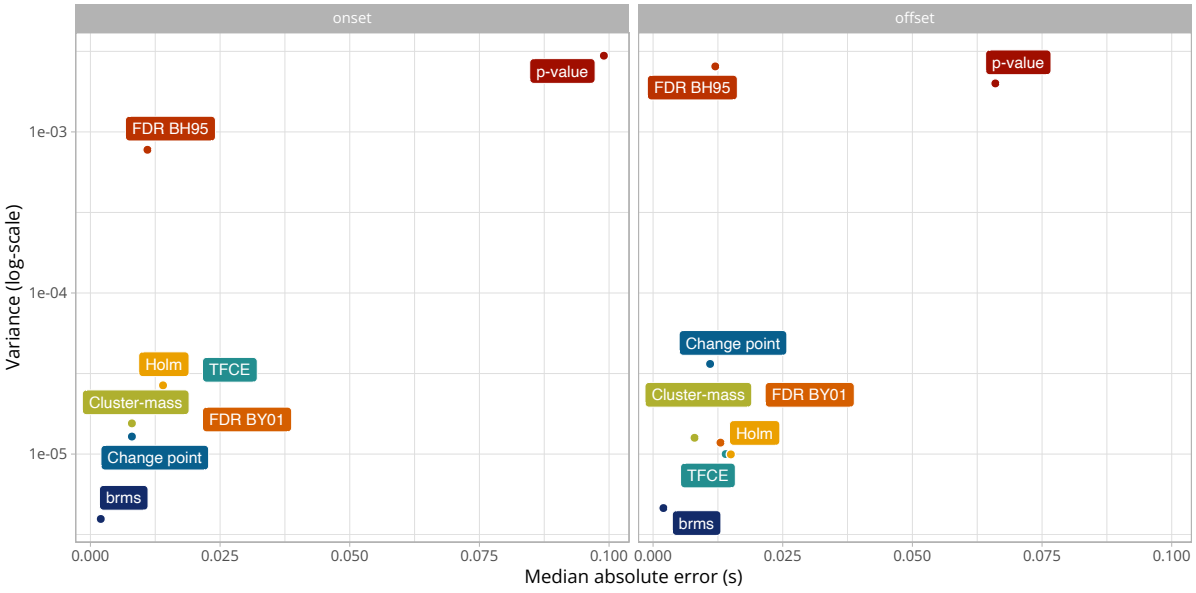
...

Simulation results (bias and variance)

Figure 8 shows a summary of the simulation results, revealing that the proposed approach (`brms`) has the lowest MAE and variance for both the onset and offset estimates...

Figure 8

Median absolute error and variance of onset and offset estimates for each method.

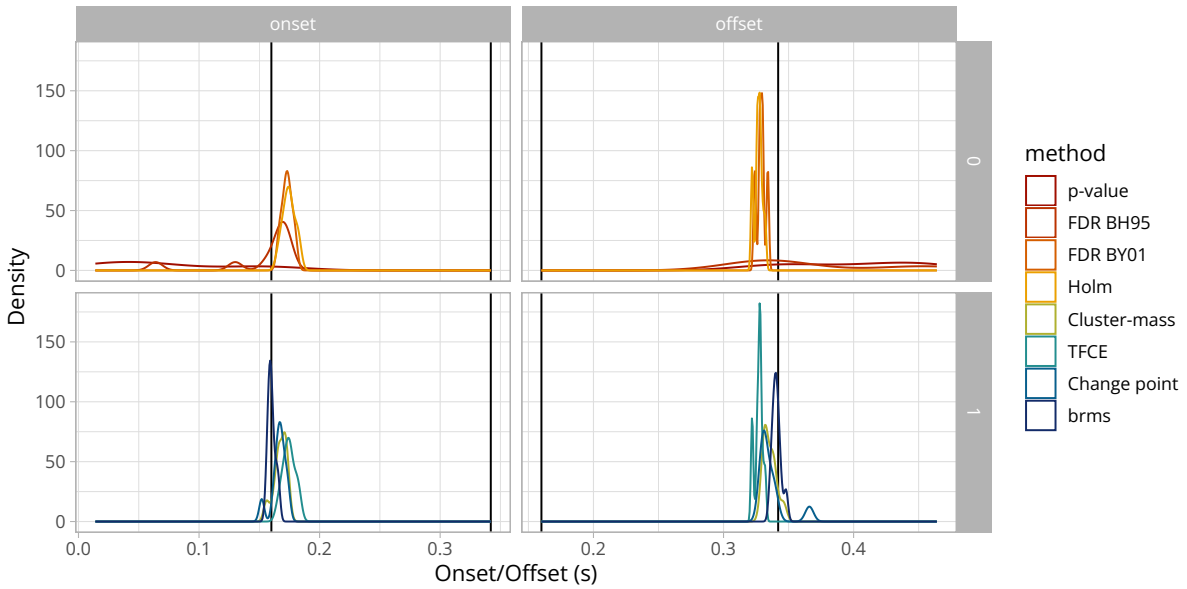


Application to actual EEG data (reliability)

...

Figure 9

Distributions of onset and offset estimates for each method.



Discussion

...

Summary of the proposed approach

...

Increasing potential usage

Prepare a wrapper `R` package and show how to call it in `Python` and integrate it with `MNE-Python` (Gramfort, 2013) pipelines...

Limitations and future directions

Can be applied to any 1D timeseries (e.g., pupillometry, electromyography)... Extending the approach to spatiotemporal data (i.e., time + sensors)...

The error properties depend on the threshold parameter, a value of 10 or 20 seems to be a reasonable default, but the optimal threshold parameter can be adjusted using split-half reliability assessment...

Conclusions

...

Data and code availability

The simulation results as well as the `R` code to reproduce the simulations are available on GitHub: https://github.com/lnalborczyk/brms_meeg.

Packages

We used R version 4.2.3 (R Core Team, 2023) and the following R packages: brms v. 2.22.0 (Bürkner, 2017, 2018, 2021), changepoint v. 2.2.4 (Killick et al., 2022b; Killick & Eckley, 2014), grateful v. 0.2.10 (Rodriguez-Sanchez & Jackson, 2023), knitr v. 1.45 (Xie, 2014, 2015, 2023), MetBrewer v. 0.2.0 (Mills, 2022), pakret v. 0.2.2 (Gallou, 2024), patchwork v. 1.2.0 (T. L. Pedersen, 2024), rmarkdown v. 2.29 (Allaire et al., 2024; Xie et al., 2018, 2020), scales v. 1.3.0 (Wickham et al., 2023), scico v. 1.5.0 (T. L. Pedersen & Crameri, 2023), tidybayes v. 3.0.6 (Kay, 2023), tidyverse v. 2.0.0 (Wickham et al., 2019).

References

Abugaber, D., Finestrat, I., Luque, A., & Morgan-Short, K. (2023). Generalized additive mixed modeling of EEG supports dual-route accounts of morphosyntax in suggesting no word frequency effects on processing of regular grammatical forms. *Journal of Neurolinguistics*, *67*, 101137. https://doi.org/10.1016/j.jneuroling.2023.101137

Allaire, J., Xie, Y., Dervieux, C., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., & Iannone, R. (2024). *rmarkdown: Dynamic documents for r*. https://github.com/rstudio/rmarkdown

Baayen, R. H., & Linke, M. (2020). *Generalized Additive Mixed Models* (pp. 563–591). Springer International Publishing. https://doi.org/10.1007/978-3-030-46216-1_23

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *57*(1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, *29*(4). https://doi.org/10.1214/aos/1013699998

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. https://doi.org/10.18637/jss.v080.i01

Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, *10*(1), 395–411. https://doi.org/10.32614/RJ-2018-017

Bürkner, P.-C. (2021). Bayesian item response modeling in R with brms and Stan. *Journal of Statistical Software*, *100*(5), 1–54. https://doi.org/10.18637/jss.v100.i05

Combrisson, E., & Jerbi, K. (2015). Exceeding chance level by chance: The caveat of theoretical chance levels in brain signal classification and statistical assessment of decoding accuracy. *Journal of Neuroscience Methods*, *250*, 126–136. https://doi.org/10.1016/j.jneumeth.2015.01.010

Dimigen, O., & Ehinger, B. V. (2021). Regression-based analysis of combined EEG and eye-tracking data: Theory and applications. *Journal of Vision*, *21*(1), 3. https://doi.org/10.1167/jov.21.1.3

Dunagan, D., Jordan, T., Hale, J. T., Pylkkänen, L., & Chacón, D. A. (2024). *Evaluating the timecourses of morpho-orthographic, lexical, and grammatical processing following rapid parallel visual presentation: An EEG investigation in english*. http://dx.doi.org/10.1101/2024.04.10.588861

Ehinger, B. V., & Dimigen, O. (2019). Unfold: An integrated toolbox for overlap correction, non-linear modeling, and regression-based EEG analysis. *PeerJ*, *7*, e7838. https://doi.org/10.7717/peerj.7838

Eklund, A., Nichols, T. E., & Knutsson, H. (2016). Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates. *Proceedings of the National Academy of Sciences*, *113*(28), 7900–7905. https://doi.org/10.1073/pnas.1602413113

Fischer, Adrian G., & Ullsperger, M. (2013). Real and Fictive Outcomes Are Processed Differently but Converge on a Common Adaptive Mechanism. *Neuron*, *79*(6), 1243–1255. https://doi.org/10.1016/j.neuron.2013.07.006

Frossard, J., & Renaud, O. (2021). Permutation Tests for Regression, ANOVA, and Comparison of Signals: The **permuco** Package. *Journal of Statistical Software*, *99*(15). https://doi.org/10.18637/jss.v099.i15

Frossard, J., & Renaud, O. (2022). The cluster depth tests: Toward point-wise strong control of the family-wise error rate in massively univariate tests with application to M/EEG. *NeuroImage*, *247*, 118824. https://doi.org/10.1016/j.neuroimage.2021.118824

Gallou, A. (2024). *pakret: Cite "R" packages on the fly in "R Markdown" and "Quarto"*. https://CRAN.R-project.org/package=pakret

Gramfort, A. (2013). MEG and EEG data analysis with MNE-python. *Frontiers in Neuroscience*, *7*. https://doi.org/10.3389/fnins.2013.00267

Greenland, S. (2019). Valid P -Values Behave Exactly as They Should: Some Misleading Criticisms of P -Values and Their Resolution With S -Values. *The American Statistician*, *73*(sup1), 106–114. https://doi.org/10.1080/00031305.2018.1529625

Hastie, T. J., & Tibshirani, R. J. (2017). *Generalized Additive Models*. Routledge. https://doi.org/10.1201/9780203753781

Hauk, O., Davis, M. H., Ford, M., Pulvermüller, F., & Marslen-Wilson, W. D. (2006). The time course of visual word recognition as revealed by linear regression analysis of ERP data. *NeuroImage*, *30*(4), 1383–1400. https://doi.org/10.1016/j.neuroimage.2005.11.048

Hayasaka, S. (2003). Validating cluster size inference: Random field and permutation methods. *NeuroImage*, *20*(4), 2343–2356. https://doi.org/10.1016/j.neuroimage.2003.08.003

Kay, M. (2023). *tidybayes: Tidy data and geoms for Bayesian models*. https://doi.org/10.5281/zenodo.1308151

Killick, R., & Eckley, I. A. (2014). changepoint: An R package for changepoint analysis. *Journal of Statistical Software*, *58*(3), 1–19. https://www.jstatsoft.org/article/view/v058i03

Killick, R., Haynes, K., & Eckley, I. A. (2022a). *changepoint: An R package for changepoint analysis*. https://CRAN.R-project.org/package=changepoint

Killick, R., Haynes, K., & Eckley, I. A. (2022b). *changepoint: An R package for changepoint analysis*. https://CRAN.R-project.org/package=changepoint

King, J.-R., & Dehaene, S. (2014). Characterizing the dynamics of mental representations: the temporal generalization method. *Trends in Cognitive Sciences*, *18*(4), 203–210. https://doi.org/10.1016/j.tics.2014.01.002

Luck, S. J., & Gaspelin, N. (2017). How to get statistically significant effects in any ERP experiment (and why you shouldn't). *Psychophysiology*, *54*(1), 146–157. https://doi.org/10.1111/psyp.12639

Maris, E. (2011). Statistical testing in electrophysiological studies. *Psychophysiology*, *49*(4), 549–565. https://doi.org/10.1111/j.1469-8986.2011.01320.x

Maris, E., & Oostenveld, R. (2007). Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods*, *164*(1), 177–190. https://doi.org/10.1016/j.jneumeth.2007.03.024

Meulman, N., Wieling, M., Sprenger, S. A., Stowe, L. A., & Schmid, M. S. (2015). Age Effects in L2 Grammar Processing as Revealed by ERPs and How (Not) to Study Them. *PLOS ONE*, *10*(12), e0143328. https://doi.org/10.1371/journal.pone.0143328

Mills, B. R. (2022). *MetBrewer: Color palettes inspired by works at the metropolitan museum of art*. https://CRAN.R-project.org/package=MetBrewer

Nalborczyk, L., Batailler, C., Lœvenbruck, H., Vilain, A., & Bürkner, P.-C. (2019). An Introduction to Bayesian Multilevel Models Using brms: A Case Study of Gender Effects on Vowel Variability in Standard Indonesian. *Journal of Speech, Language, and Hearing Research*, *62*(5), 1225–1242. https://doi.org/10.1044/2018_jslhr-s-18-0006

Pedersen, E. J., Miller, D. L., Simpson, G. L., & Ross, N. (2019). Hierarchical generalized additive models in ecology: An introduction with mgcv. *PeerJ*, *7*, e6876. https://doi.org/10.7717/peerj.6876

Pedersen, T. L. (2024). *patchwork: The composer of plots*. https://CRAN.R-project.org/package=patchwork

Pedersen, T. L., & Crameri, F. (2023). *scico: Colour palettes based on the scientific colour-maps*. https://CRAN.R-project.org/package=scico

Pernet, C. (2022). Electroencephalography robust statistical linear modelling using a single weight per trial. *Aperture Neuro*, *2*, 1–22. https://doi.org/10.52294/apertureneuro.2022.2.seoo9435

Pernet, C. R., Latinus, M., Nichols, T. E., & Rousselet, G. A. (2015). Cluster-based com-

putational methods for mass univariate analyses of event-related brain potentials/fields: A simulation study. *Journal of Neuroscience Methods*, *250*, 85–93. https://doi.org/10.1016/j.jneumeth.2014.08.003

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Rodriguez-Sanchez, F., & Jackson, C. P. (2023). *grateful: Facilitate citation of r packages*. https://pakillo.github.io/grateful/

Rosenblatt, J. D., Finos, L., Weeda, W. D., Solari, A., & Goeman, J. J. (2018). All-Resolutions Inference for brain imaging. *NeuroImage*, *181*, 786–796. https://doi.org/10.1016/j.neuroimage.2018.07.060

Rousselet, G. A. (2025). Using cluster-based permutation tests to estimate MEG/EEG onsets: How bad is it? *European Journal of Neuroscience*, *61*(1), e16618. https://doi.org/10.1111/ejn.16618

Rousselet, G. A., Pernet, C. R., Bennett, P. J., & Sekuler, A. B. (2008). Parametric study of EEG sensitivity to phase noise during face processing. *BMC Neuroscience*, *9*(1). https://doi.org/10.1186/1471-2202-9-98

Sassenhagen, J., & Draschkow, D. (2019). Cluster-based permutation tests of MEG/EEG data do not establish significance of effect latency or location. *Psychophysiology*, *56*(6). https://doi.org/10.1111/psyp.13335

Skukies, R., & Ehinger, B. (2021). Modelling event duration and overlap during EEG analysis. *Journal of Vision*, *21*(9), 2037. https://doi.org/10.1167/jov.21.9.2037

Skukies, R., Schepers, J., & Ehinger, B. (2024, December 9). *Brain responses vary in duration - modeling strategies and challenges*. https://doi.org/10.1101/2024.12.05.626938

Smith, S., & Nichols, T. (2009). Threshold-free cluster enhancement: Addressing problems of smoothing, threshold dependence and localisation in cluster inference. *NeuroImage*, *44*(1), 83–98. https://doi.org/10.1016/j.neuroimage.2008.03.061

Sóskuthy, M. (2017). *Generalised additive mixed models for dynamic analysis in linguistics: A practical introduction*. https://doi.org/10.48550/ARXIV.1703.05339

Teichmann, L. (2022). An empirically driven guide on using bayes factors for m/EEG decoding. *Aperture Neuro*, *2*, 1–10. https://doi.org/10.52294/apertureneuro.2022.2.maoc6465

Ushey, K., Allaire, J., & Tang, Y. (2024). *Reticulate: Interface to 'python'*. https://CRAN.R-project.org/package=reticulate

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., ... Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., Pedersen, T. L., & Seidel, D. (2023). *scales: Scale functions for visualization*. https://CRAN.R-project.org/package=scales

Wieling, M. (2018). Analyzing dynamic phonetic data using generalized additive mixed modeling: A tutorial focusing on articulatory differences between L1 and L2 speakers of English. *Journal of Phonetics*, *70*, 86–116. https://doi.org/10.1016/j.wocn.2018.03.002

Winter, B., & Wieling, M. (2016). How to analyze linguistic change using mixed models, Growth Curve Analysis and Generalized Additive Modeling. *Journal of Language Evolution*, *1*(1), 7–18. https://doi.org/10.1093/jole/lzv003

Wood, S. N. (2003). Thin Plate Regression Splines. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *65*(1), 95–114. https://doi.org/10.1111/1467-9868.00374

Wood, S. N. (2017). *Generalized Additive Models*. Chapman; Hall/CRC. https://doi.org/10.1201/9781315370279

Wüllhorst, V., Wüllhorst, R., Overmeyer, R., & Endrass, T. (2025). Comprehensive Analysis of Event-Related Potentials of Response Inhibition: The Role of Negative Urgency and

Compulsivity. *Psychophysiology*, *62*(2). https://doi.org/10.1111/psyp.70000

Xie, Y. (2014). knitr: A comprehensive tool for reproducible research in R. In V. Stodden, F. Leisch, & R. D. Peng (Eds.), *Implementing reproducible computational research*. Chapman; Hall/CRC.

Xie, Y. (2015). *Dynamic documents with R and knitr* (2nd ed.). Chapman; Hall/CRC. https://yihui.org/knitr/

Xie, Y. (2023). *knitr: A general-purpose package for dynamic report generation in r*. https://yihui.org/knitr/

Xie, Y., Allaire, J. J., & Grolemund, G. (2018). *R markdown: The definitive guide*. Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown

Xie, Y., Dervieux, C., & Riederer, E. (2020). *R markdown cookbook*. Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown-cookbook

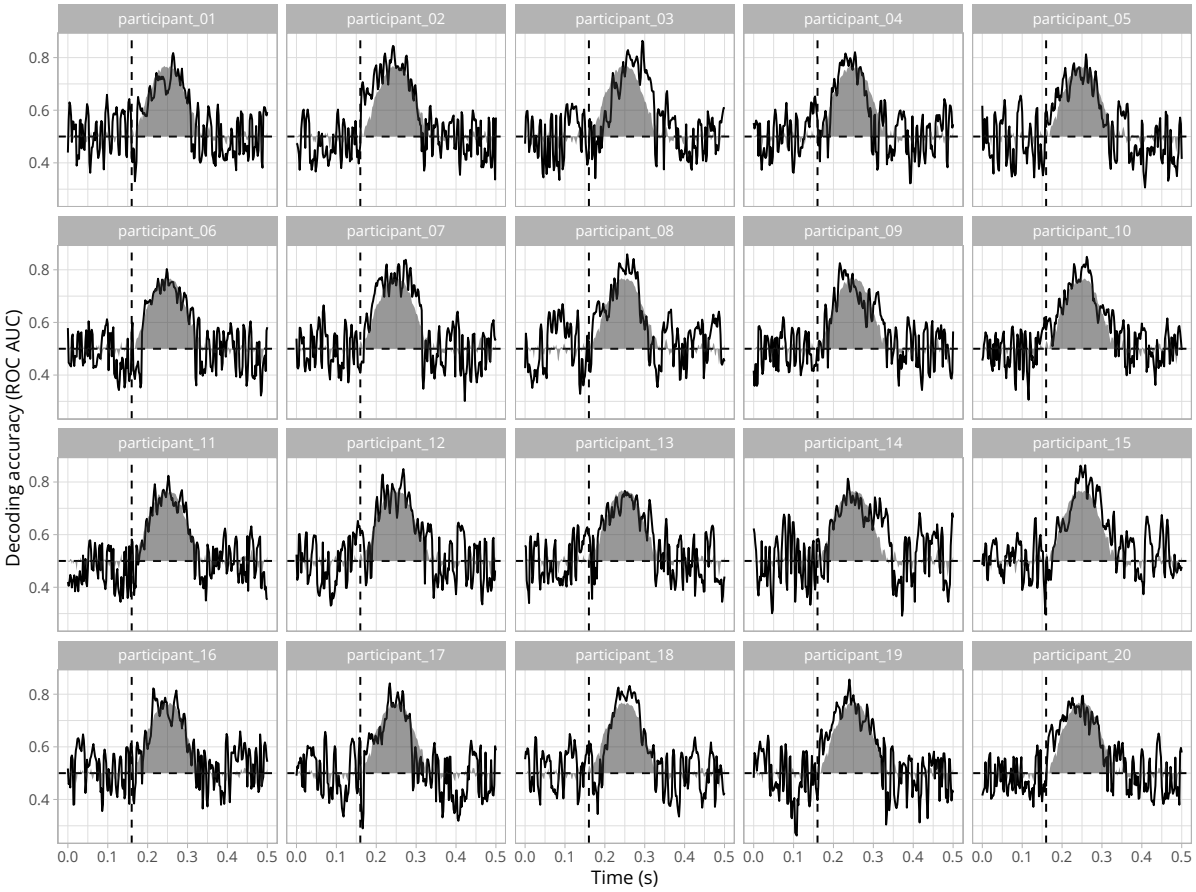
Yeung, N., Bogacz, R., Holroyd, C. B., & Cohen, J. D. (2004). Detection of synchronized oscillations in the electroencephalogram: An evaluation of methods. *Psychophysiology*, *41*(6), 822–832. https://doi.org/10.1111/j.1469-8986.2004.00239.x

Appendix A
Application to time-resolved decoding results (accuracy over time)

We conducted time-resolved multivariate pattern analysis (MVPA), also known as decoding. As a result, we have a timecourse of decoding accuracies (e.g., ROC AUC), bounded between 0 and 1, per participant (Figure A1)...

Figure A1

Exemplary average timecourse of binary decoding accuracy (ROC AUC) for each participant. Group-level average decoding accuracy is depicted as a grey background density in each panel.



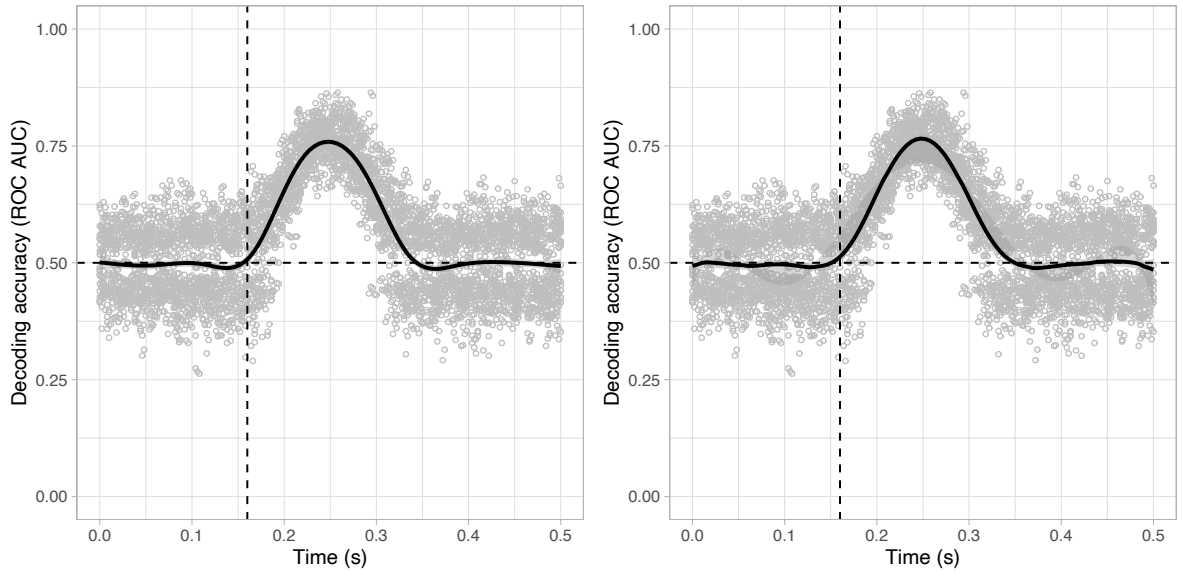
Now, we want to *test* whether the group-level average decoding accuracy is above chance (i.e., 0.5) at each timestep. We use a similar GAM/GP as previously, but we replace the Normal likelihood function by a Beta one to account for the bounded nature of AUC values (between 0 and 1).

```
# fitting the GAM
decoding_gam <- brm(
    auc ~ s(time, bs = "cr", k = 10),
    data = decoding_data,
    family = Beta(),
    iter = 5000,
    chains = 4,
    cores = 4,
    file = "models/decoding_gam.rds"
    )
```

```
# fitting the GP
decoding_gp <- brm(
    auc ~ gp(time, k = 20),
    data = decoding_data,
    family = Beta(),
    control = list(adapt_delta = 0.99),
    iter = 5000,
    chains = 4,
    cores = 4,
    file = "models/decoding_gp.rds"
    )
```

Figure A2

Posterior predictions of the GAM (left) and GP (right) fitted on decoding accuracy over time.



Next, we plot the posterior probability of decoding accuracy being above chance level (plus some epsilon) (Figure A3).

Figure A3

Posterior probability of decoding accuracy being above chance level according to the GAM (left) or the GP (right).

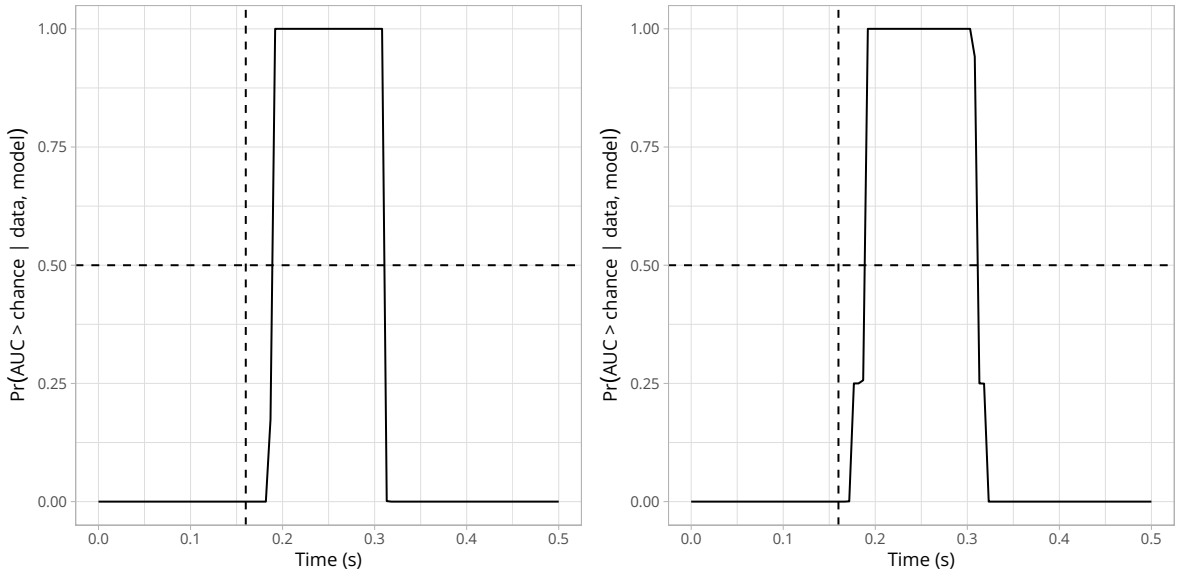
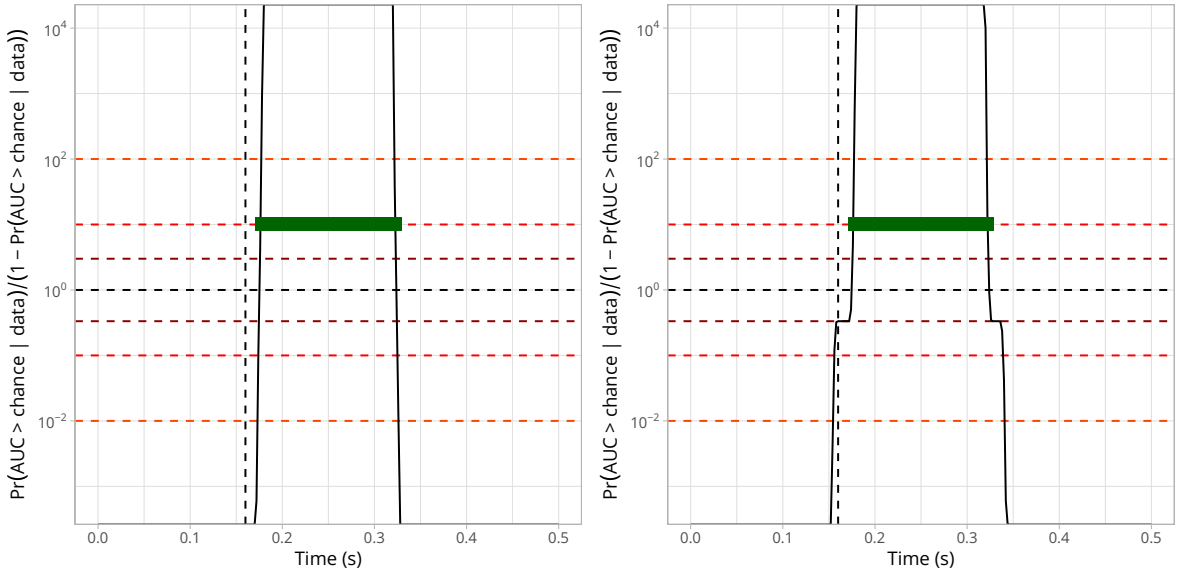


Figure A4

Ratio of posterior probabilities of decoding accuracy being above chance level according to the GAM (left) or the GP (right).



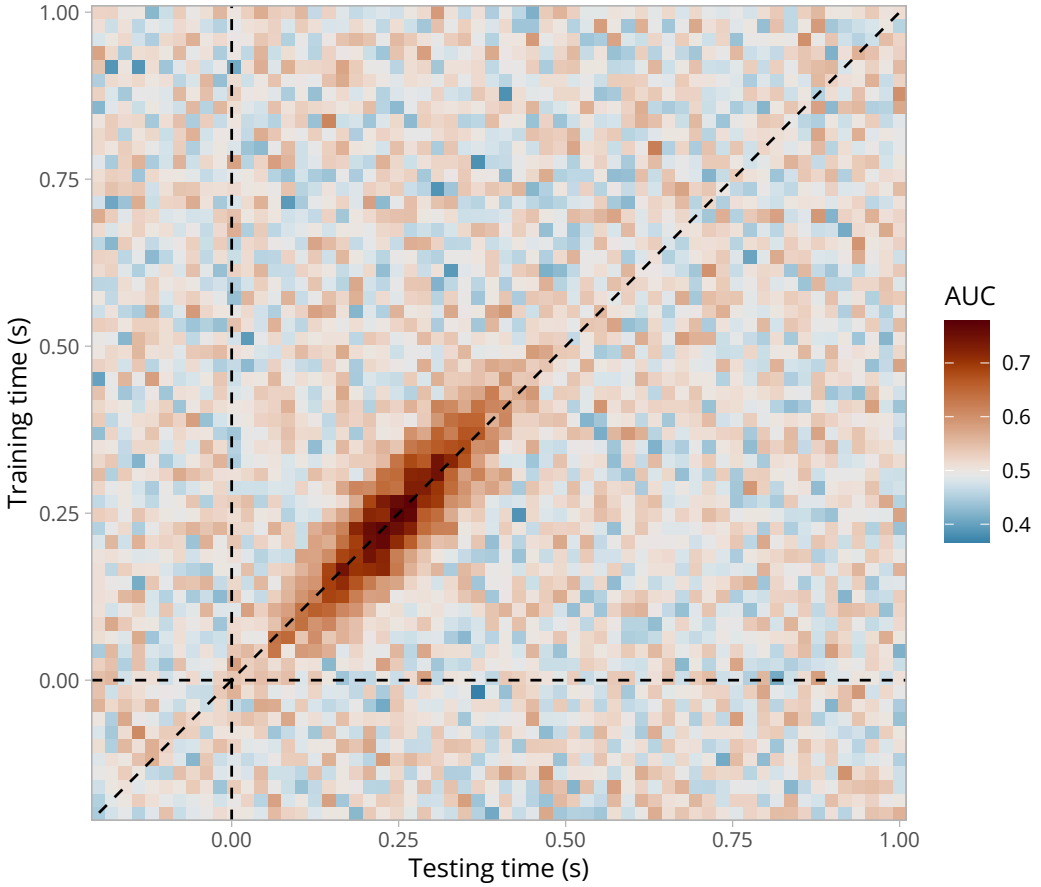
Appendix B
Application to 2D time-resolved decoding results (cross-temporal generalisation)

Assume we have M/EEG data and we have conducted cross-temporal generalisation analyses (King & Dehaene, 2014). As a result, we have a 2D matrix where each element contains the decoding accuracy (e.g., ROC AUC) of a classifier trained at timestep training$_i$ and tested at timestep testing$_j$ (Figure B1).

Now, we want to test whether and when decoding performance is above chance level (0.5 for a binary decoding task). These two models are computationally heavier to fit (more

Figure B1

Exemplary (simulated) group-level average cross-temporal generalisation matrix of decoding performance (ROC AUC).



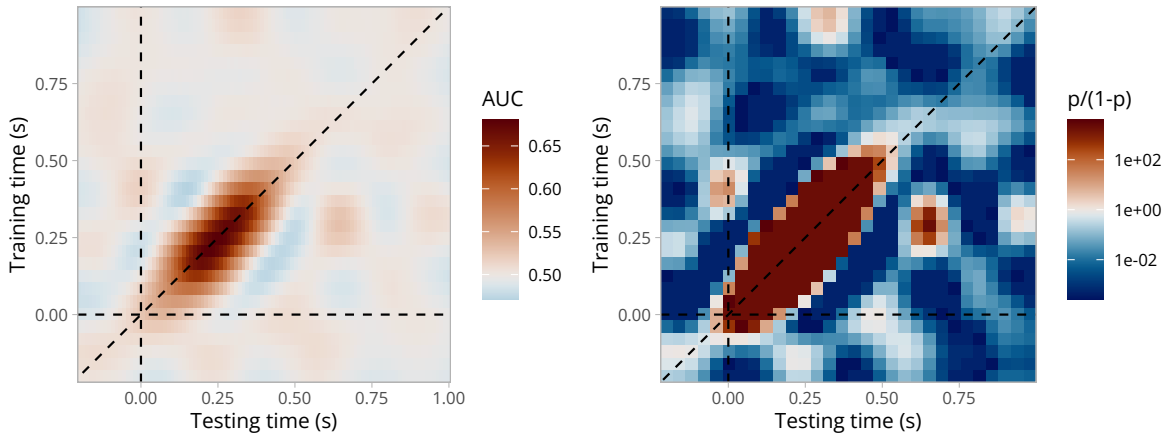
observations and 2D smooth functions)...

```
# fitting a GAM with two temporal dimensions
timegen_gam <- brm(
    # 2D thin-plate spline (tp)
    # auc ~ s(train_time, test_time, bs = "tp", k = 10),
    auc ~ t2(train_time, test_time, bs = "tp", k = 10),
    data = timegen_data,
    family = Beta(),
    iter = 5000,
    chains = 4,
    cores = 4,
    file = "models/timegen_gam_t2.rds"
    )

# fitting a GP with two temporal dimensions
# timegen_gp <- brm(
#     auc ~ gp(train_time, test_time, k = 20),
#     data = timegen_data,
#     family = Beta(),
#     control = list(adapt_delta = 0.95),
```

```
#     iter = 2000,
#     chains = 4,
#     cores = 4,
#     file = "models/timegen_gp.rds"
#     )
```

Figure B2

Posterior probability of decoding accuracy being above chance level (2D GAM).



Appendix C
Mathematical formulation of the bivariate GAM

To model cross-temporal generalisation matrices of decoding performance (ROC AUC), we extended the initial (decoding) GAM to take into account the bivariate temporal distribution of AUC values, thus producing naturally smoothed estimates (timecourses) of AUC values and posterior probabilities. This model can be written as follows:

$$\begin{aligned} \mathrm{AUC}_i &\sim \mathrm{Beta}(\mu_i, \phi) \\ g(\mu_i) &= f\left(\mathrm{train}_i, \mathrm{test}_i\right) \end{aligned}$$

where we assume that AUC values come from a Beta distribution with two parameters μ and ϕ. We can think of $f\left(\mathrm{train}_i, \mathrm{test}_i\right)$ as a surface (a smooth function of two variables) that we can model using a 2-dimensional splines. Let $\mathbf{s}_i = \left(\mathrm{train}_i, \mathrm{test}_i\right)$ be some pair of training and testing samples, and let $\mathbf{k}_m = \left(\mathrm{train}_m, \mathrm{test}_m\right)$ denote the m^{th} knot in the domain of train_i and test_i. We can then express the smooth function as:

$$f\left(\mathrm{train}_i, \mathrm{test}_i\right) = \alpha + \sum_{m=1}^{M} \beta_m b_m \left(\tilde{s}_i, \tilde{k}_m\right)$$

Note that $b_m(,)$ is a basis function that maps $R \times R \to R$. A popular bivariate basis function uses *thin-plate splines* (Wood, 2003), which extend to $\mathbf{s}_i \in \mathbb{R}^d$ and ∂l_g penalties. These splines are designed to interpolate and approximate smooth surfaces over two dimensions (hence the "bivariate" term). For $d = 2$ dimensions and $l = 2$ (smoothness penalty involving second order derivative):

$$f\left(\tilde{s}_i\right) = \alpha + \beta_1 x_i + \beta_2 z_i + \sum_{m=1}^{M} \beta_{2+m} b_m \left(\tilde{s}_i, \tilde{k}_m\right)$$

using the the radial basis function given by:

$$b_m \left(\tilde{s}_i, \tilde{k}_m\right) = \left\|\tilde{s}_i - \tilde{k}_m\right\|^2 \log \left\|\tilde{s}_i - \tilde{k}_m\right\|$$

where $\|\mathbf{s}_i - \mathbf{k}_m\|$ is the Euclidean distance between the covariate $\mathbf{s}_i$ and the knot location $\mathbf{k}_m$.

Appendix D
Threshold-free cluster enhancement

Cluster-based permutation approaches require defining a cluster-forming threshold (e.g., a t- or f-value) as the initial step of the algorithm. As different cluster-forming thresholds lead to clusters with different spatial or temporal extent, this threshold modulates the sensitivity of the subsequent permutation test. The threshold-free cluster enhancement method (TFCE) was introduced by Smith & Nichols (2009) to overcome this arbitrary threshold.

In brief, the TFCE method works as follows. Instead of picking an arbitrary cluster-forming threshold (e.g., $t = 2$), we try all (or many) possible thresholds in a given range and check whether a given timestep/voxel belongs to a significant cluster under any of the set of thresholds... Then, instead of using cluster mass, we use a weighted average between the cluster extend (e, how broad is the cluster, that is, how many connected samples it contains) and the cluster height (h, how high is the cluster, that is, how large is the test statistic) according to the formula:

$$\text{TFCE} = \int_h e(h)^E h^H \mathrm{d}h$$

Where... the parameters E and H are set a priori and control the influence of the extend and height on the TFCE. Then, p-value for timestep/voxel i is computed by comparing it TFCE with the null distribution of TFCE values. For each permuted signal, we keep the maximal value over the whole signal for the null distribution of the TFCE.... But see Sassenhagen & Draschkow (2019)...

Appendix E
Using the `R` package and integration with `MNE-Python`

Explain how to use the `R` package and to integrate it with `MNE` epochs...

```
# to-do adding some code here...
```