The Journal of Open Source Software

UnfoldSim.jl: Simulating continuous event-based time series data for EEG and beyond

Judith Schepers [1], **Luis Lips**[1], **Maanik Marathe**[1], **and Benedikt V. Ehinger** [1,2]

1 Institute for Visualisation and Interactive Systems, University of Stuttgart, Germany 2 Stuttgart Center for Simulation Science, University of Stuttgart, Germany

DOI: 10.21105/joss.06641

Software

- Review
- Repository
- Archive

Editor: Beatriz Costa Gomes

Reviewers:

- @cbrnr
- @aurorarossi
- @roualdes

Submitted: 19 February 2024
Published: 14 March 2025

Summary

`UnfoldSim.jl` is a Julia package for simulating multivariate time series, with a focus on EEG, especially event-related potentials (ERPs). The user provides four ingredients: 1) an experimental design, with both categorical and continuous variables, 2) event basis functions specified via linear or hierarchical models, 3) an inter-event onset distribution, and 4) a noise specification. `UnfoldSim.jl` then simulates continuous EEG signals with potentially overlapping events. Multi-channel support via EEG-forward models is available as well. `UnfoldSim.jl` is modular, allowing users to implement custom designs, components, onset distributions, or noise types to tailor the package to their needs. This allows support even for other modalities, e.g., single-voxel fMRI or pupil dilation signals.

One can find a detailed example of how to use `UnfoldSim.jl` to simulate continuous EEG data in the documentation.

Statement of Need

In our work (e.g., Ehinger & Dimigen (2019), Dimigen & Ehinger (2021)), we often analyze data containing (temporally) overlapping events (e.g., stimulus onset and button press, or consecutive eye-fixations), non-linear effects, and complex experimental designs. For a multitude of reasons, we often need to simulate such kinds of data: simulated EEG data is useful to test preprocessing and analysis tools, validate statistical methods, illustrate conceptual issues, test toolbox functionalities, and find limitations of traditional analysis workflows. For instance, such simulation tools allow for testing the assumptions of new analysis algorithms and testing their robustness against any violation of these assumptions.

While other EEG simulation toolboxes exist, they each have limitations: they are dominantly based on the proprietary MATLAB software, they do not simulate continuous EEG, and they offer little support for designs more complex than two conditions or with non-linear effects. In contrast, `UnfoldSim.jl` is free and open-source and it allows to simulate continuous EEG signals even for complex designs.

Functionality

The package provides four abstract types: `AbstractDesign`, `AbstractComponent`, `AbstractOnset` and `AbstractNoise`. In the following, we present the concrete types that are currently implemented.

Experimental designs

The design contains the levels of all conditions and predictors. Currently, we support a single and a multi-subject design. The multi-subject design uses the `MixedModelsSim.jl` package (Alday et al., 2024) and allows a flexible specification of the random-effects structure by indicating which predictors are within- or between-subject (or item). Designs can be encapsulated, for instance, using the `RepeatDesign` type which repeats the generated event table multiple times, thus generating new trials.

Event basis functions (Components)

`UnfoldSim.jl` provides a `LinearModelComponent` and a `MixedModelComponent` for single- and multi-subject simulation respectively. These components determine the shape of the response to an event. They consist of a basis function which is weighted by the user-defined regression model. Users can specify a basis function by providing a custom vector or selecting from predefined options, such as simplified EEG components like the N170, modelled as temporally shifted Hanning windows. Further, in the components' model formulae, fixed effects (βs) and random effects (`MultiSubjectDesigns` only) need to be specified.

Each component can be nested in a `MultichannelComponent`, which, using a forward head model, projects the simulated source component to the multi-channel electrode space. Using `Artifacts.jl` we provide on-demand access to the HArtMuT (Harmening et al., 2022) model.

To generate complex activations, it is possible to specify a vector of components.

Inter-onset distributions

The inter-onset distribution defines the distance between events in the continuous (EEG) signal. Currently, `UniformOnset` and `LogNormalOnset` are implemented (see Figure 1). By adjusting the distribution's parameters, one indirectly controls the amount of overlap between the event-related responses.

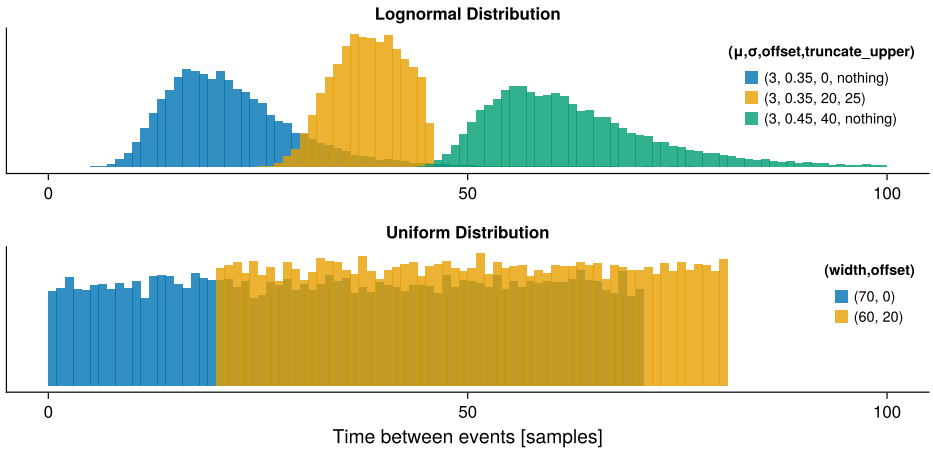

Figure 1: Illustration of the inter-onset distributions. The colours indicate different sets of parameter values. Please note that for the lognormal distribution, the parameters are defined on a logarithmic scale, while the histogram is shown on a linear scale.

Noise types

`UnfoldSim.jl` offers different noise types: `WhiteNoise`, `RedNoise`, `PinkNoise`, and exponentially decaying autoregressive noise (`ExponentialNoise`) (see Figure 2). In the future, we will add simple autoregressive noise and noise based on actual EEG data.

The Journal of Open Source Software

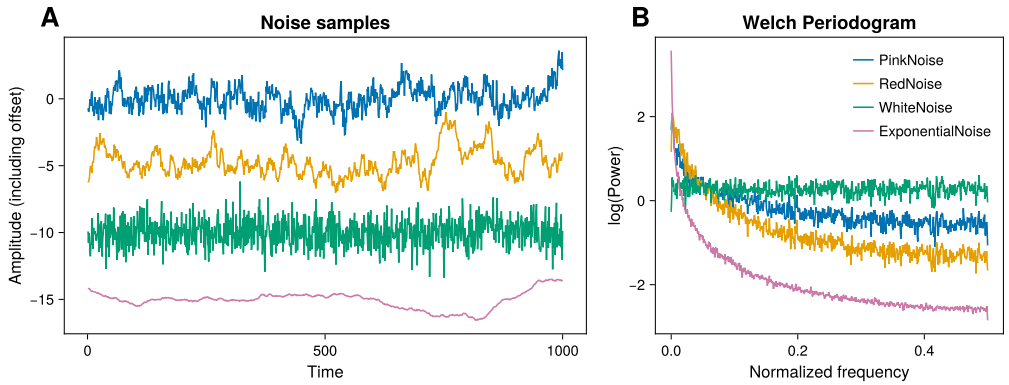

Figure 2: Illustration of the different noise types (indicated by colour). Panel **A** shows the noise over time. Please note that the noise signals are shifted by 5 µV for visualisation purposes. Panel **B** displays its $\log_{10}$(power) at normalized frequencies.

Related tools

Few toolboxes for simulating EEG data exist, most being proprietary MATLAB tools that have often not received any updates in the past years or have very specific applications (e.g., `EEGg` (Vaziri et al., 2023), `SimMEEG` (Herdman, 2021), `SEED-G` (Anzolin et al., 2021), `EEGSourceSim` (Barzegaran et al., 2019), `simBCI` (Lindgren et al., 2018)).

In the following, we highlight two actively developed MATLAB-based tools: `Brainstorm` (Tadel et al., 2011), which especially excels at visualizing the forward model and generating ERPs from phase-aligned oscillations, and `SEREEGA` (Krol et al., 2018), which offers comprehensive simulation capabilities with a focus on ERP-component simulation, tools for benchmarking like signal-to-noise specification and more realistic noise simulation (e.g., via random sources).

In Python, `MNE-Python` (Gramfort et al., 2013) provides some tutorials to simulate EEG data, but the functionality is very basic. `HNN-Core` (Jas et al., 2023) can simulate realistic EEG data by parameterising the neuronal activity in cortical columns.

In contrast to these tools, `UnfoldSim.jl` has a higher-level perspective, uniquely focusing on the regression-ERP aspect. It provides functions to simulate multi-condition experiments, uniquely allows for modeling multi-subject EEG datasets, and offers support to model continuous EEG data with overlapping events. Further, the implementation in Julia offers a platform that is free, actively encourages research software engineering methods, makes it easy to add custom expansions via the `AbstractTypes`, and allows easy access from Python and R.

Acknowledgements

Funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016. The authors further thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Judith Schepers. Moreover, the authors would like to thank Tanja Bien for her valuable feedback on the paper manuscript.

Package references

Please note that we list only the main dependencies here, but the dependencies of the dependencies can be found in the respective `Manifest.toml` files. Furthermore, please note

that we only list rather than cite the packages for which we could not find any citation instructions.

Julia (Bezanson et al., 2017), **DataFrames.jl** (Bouchet-Valat & Kamiński, 2023), **Distributions.jl** (Besançon et al., 2021; Lin et al., 2019), **Documenter.jl** (Hatherly et al., n.d.), **DSP.jl** (Kornblith et al., 2023), **FileIO.jl**, **Glob.jl**, **HDF5.jl**, **HypothesisTests.jl**, **ImageFiltering.jl**, **Literate.jl**, **LiveServer.jl**, **Makie.jl** (Danisch & Krumbiegel, 2021), **MixedModels.jl** (Bates et al., 2023), **MixedModelsSim.jl** (Alday et al., 2024), **Parameters.jl**, **PrettyTables.jl** (Chagas et al., 2023), **ProjectRoot.jl**, **SignalAnalysis.jl**, **Statistics.jl**, **StableRNGs.jl**, **StatsBase.jl**, **StatsModels.jl**, **TimerOutputs.jl**, **ToeplitzMatrices.jl**, **Unfold.jl** (Ehinger & Dimigen, 2019), **UnfoldMakie.jl** (Mikheev et al., 2023)

References

Alday, P., Bates, D., DeBruine, L., Ehinger, B., Calderón, J. B. S., Schepers, J., Choe, J., & Schwetlick, L. (2024). *RePsychLing/MixedModelsSim.jl: v0.2.9* (Version v0.2.9). Zenodo. https://doi.org/10.5281/zenodo.10669002

Anzolin, A., Toppi, J., Petti, M., Cincotti, F., & Astolfi, L. (2021). SEED-g: Simulated EEG data generator for testing connectivity algorithms. *Sensors*, *21*(11), 3632. https://doi.org/10.3390/s21113632

Barzegaran, E., Bosse, S., Kohler, P. J., & Norcia, A. M. (2019). EEGSourceSim: A framework for realistic simulation of EEG scalp data using MRI-based forward models and biologically plausible signals and noise. *Journal of Neuroscience Methods*, *328*, 108377. https://doi.org/10.1016/j.jneumeth.2019.108377

Bates, D., Alday, P., Kleinschmidt, D., Calderón, J. B. S., Zhan, L., Noack, A., Bouchet-Valat, M., Arslan, A., Kelman, T., Baldassari, A., Ehinger, B., Karrasch, D., Saba, E., Quinn, J., Hatherly, M., Piibeleht, M., Mogensen, P. K., Babayan, S., Holy, T., … Nazarathy, Y. (2023). *JuliaStats/MixedModels.jl: v4.22.3* (Version v4.22.3). Zenodo. https://doi.org/10.5281/zenodo.10268806

Besançon, M., Papamarkou, T., Anthoff, D., Arslan, A., Byrne, S., Lin, D., & Pearson, J. (2021). Distributions.jl: Definition and modeling of probability distributions in the JuliaStats ecosystem. *Journal of Statistical Software*, *98*(16), 1–30. https://doi.org/10.18637/jss.v098.i16

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Bouchet-Valat, M., & Kamiński, B. (2023). DataFrames.jl: Flexible and fast tabular data in julia. *Journal of Statistical Software*, *107*(4), 1–32. https://doi.org/10.18637/jss.v107.i04

Chagas, R. A. J., Baumgold, B., Hertz, G., Ranocha, H., Wells, M., Ritchie, N., Christensen, Z. P., gustaphe, pdeffebach, Pasquier, B., Kawczynski, C., Sambrani, D., Vanguelov, D., Hanson, E., Butterworth, I., Peters, J., Kraak, J., Storopoli, J., TagBot, J., … lorenzoh. (2023). *Ronisbr/PrettyTables.jl: v2.3.1* (Version v2.3.1). Zenodo. https://doi.org/10.5281/zenodo.10214175

Danisch, S., & Krumbiegel, J. (2021). Makie.jl: Flexible high-performance data visualization for Julia. *Journal of Open Source Software*, *6*(65), 3349. https://doi.org/10.21105/joss.03349

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

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

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., & Hämäläinen, M. S. (2013). MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, *7*(267), 1–13. https://doi.org/10.3389/fnins.2013.00267

Harmening, N., Klug, M., Gramann, K., & Miklody, D. (2022). HArtMuT—modeling eye and muscle contributors in neuroelectric imaging. *Journal of Neural Engineering*, *19*(6), 066041. https://doi.org/10.1088/1741-2552/aca8ce

Hatherly, M., Piibeleht, M., Ekre, F., & Contributors. (n.d.). *Documenter: A documentation generator for Julia*. https://github.com/JuliaDocs/Documenter.jl

Herdman, A. T. (2021). SimMEEG software for simulating event-related MEG and EEG data with underlying functional connectivity. *Journal of Neuroscience Methods*, *350*, 109017. https://doi.org/10.1016/j.jneumeth.2020.109017

Jas, M., Thorpe, R., Tolley, N., Bailey, C., Brandt, S., Caldwell, B., Cheng, H., Daniels, D., Pujol, C. F., Khalil, M., Kanekar, S., Kohl, C., Kolozsvári, O., Lankinen, K., Loi, K., Neymotin, S., Partani, R., Pelah, M., Rockhill, A., … Jones, S. (2023). HNN-core: A python software for cellular and circuit-level interpretation of human MEG/EEG. *Journal of Open Source Software*, *8*(92), 5848. https://doi.org/10.21105/joss.05848

Kornblith, S., Lynch, G., Holters, M., Santos, J. F., Russell, S., Kickliter, J., Bezanson, J., Adalsteinsson, G., Arslan, A., Yamamoto, R., jordancluts, Pastell, M., Kelman, T., Arthur, B., Krauss, T., HDictus, El-Saawy, H., Kofron, J., Hanson, E., … Smith, J. (2023). *JuliaDSP/DSP.jl: v0.7.9* (Version v0.7.9). Zenodo. https://doi.org/10.5281/zenodo.8344531

Krol, L. R., Pawlitzki, J., Lotte, F., Gramann, K., & Zander, T. O. (2018). SEREEGA: Simulating event-related EEG activity. *Journal of Neuroscience Methods*, *309*, 13–24. https://doi.org/10.1016/j.jneumeth.2020.109017

Lin, D., White, J. M., Byrne, S., Bates, D., Noack, A., Pearson, J., Arslan, A., Squire, K., Anthoff, D., Papamarkou, T., Besançon, M., Drugowitsch, J., Schauer, M., & contributors, other. (2019). *JuliaStats/Distributions.jl: a Julia package for probability distributions and associated functions*. https://doi.org/10.5281/zenodo.2647458

Lindgren, J. T., Merlini, A., Lecuyer, A., & Andriulli, F. P. (2018). simBCI—a framework for studying BCI methods by simulated EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *26*(11), 2096–2105. https://doi.org/10.1109/TNSRE.2018.2873061

Mikheev, V., Döring, S., Gärtner, N., Baumgartner, D., & Ehinger, B. (2023). *UnfoldMakie* (Version v0.4.0). Zenodo. https://doi.org/10.5281/zenodo.10235220

Tadel, F., Baillet, S., Mosher, J. C., Pantazis, D., & Leahy, R. M. (2011). Brainstorm: A user-friendly application for MEG/EEG analysis. *Computational Intelligence and Neuroscience*, *2011*, 1–13. https://doi.org/10.1155/2011/879716

Vaziri, A. Y., Makkiabadi, B., & Samadzadehaghdam, N. (2023). EEGg generating synthetic EEG signals in matlab environment. *Frontiers in Biomedical Technologies*, *10*(3), 370–381. https://doi.org/10.18502/fbt.v10i3.13165