The Journal of Open Source Software

MNE-RSA: Representational Similarity Analysis on EEG, MEG and fMRI data

Marijn van Vliet[1], **Stefan Appelhoff**[2], **Takao Shimizu**[1], **Egor Eremin**[1], **Annika Hultén**[1], **Yuanfang Zhao**[3], and **Richard Höchenberger**[4]

1 Department of Neuroscience and Biomedical Engineering, Aalto University, Finland **2** Max Planck Institute for Human Development, Berlin, Germany **3** Department of Cognitive Science, Johns Hopkins University, Baltimore, USA **4** Inria, CEA, Université Paris-Saclay, Palaiseau, France

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: David Stansby

Reviewers:

- @alexrockhill
- @weiyongxu
- @alinepriscillia

Submitted: 16 June 2025
Published: unpublished

Summary

MNE-RSA is a Python package for performing representational similarity analysis (RSA) on non-invasive measurements of brain activity, namely electroencephalography (EEG), magnetoencephalography (MEG) and functional magnetic resonance imaging (fMRI). It serves as an extension to MNE-Python (Gramfort et al., 2013), which is a comprehensive package for preprocessing EEG/MEG data and performing source estimation and implements the many preprocessing steps required for RSA analysis. After preprocessing is done, MNE-RSA provides a straightforward way to perform the actual RSA on data that is loaded as MNE-Python datastructures.

About RSA

RSA is a technique to compare information flows within complex systems (Kriegeskorte et al., 2008). In the context of this software package, this mostly means comparing different representations of input stimuli to neural representations at different locations and times in the brain. Example representations of a stimulus would be the pixels of an image, or the semantic features of the object depicted in the image (“has a tail”, “barks”, “good boy”), or an embedding vector obtained with a convolutional neural network (CNN) or large language network (LLM) (Diedrichsen & Kriegeskorte, 2017). Example neural representations include the pattern of electric potentials across EEG sensors, or the magnetic field pattern across MEG sensors, or the pattern of source localized activity across the cortex, or the pattern of beta values across fMRI voxels. Whenever one can create multiple representations of the same stimuli, one can compare these representations using RSA to judge their “representational similarity” (Figure 1). The key to this is the creation of a representational dissimilarity matrix (RDM) which is an all-to-all distance matrix between the representations of a set of stimuli, usually obtained by correlating the representation vectors of each pair of stimuli. Once an RDM is obtained for the different representation schemes (typically you have one obtained through some model and one obtained from brain activity) they can be compared (again using correlation) to yield an RSA score. When one does this in a “searchlight” pattern across the brain, the result is a map of RSA scores indicating where and when in the brain the neural representation corresponds to the model.

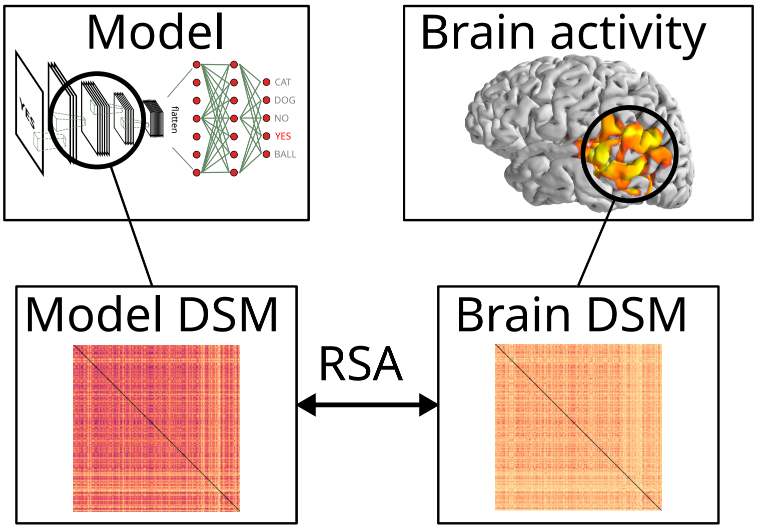

Figure 1: Schematic overview of representational similarity analysis (RSA).

Functionality

The core functionality of MNE-RSA consists of an efficient pipeline that operates on NumPy arrays, starting from "searchlight" (i.e. multi-dimensional sliding window) indexing, to cross-validated computation of RDMs, to the comparison with "model" RDMs to produce RSA values. On top of the general purpose pipeline, MNE-RSA exposes functions that operate on MNE-Python (EEG, MEG) and Nibabel (fMRI) objects and also return the resulting RSA values as such objects. Those functions leverage the available metadata, such as the sensor layout, edges of cortical 3D meshes, and voxel sizes, to present a more intuitive API.

MNE-RSA supports all the distance metrics in `scipy.spatial.distance` for computing RDMs and the following metrics for comparing RDMs:

- Spearman correlation (the default)
- Pearson correlation
- Kendall's Tau-A
- Linear regression (when comparing multiple RDMs at once)
- Partial correlation (when comparing multiple RDMs at once)

Here is an example showcasing how to use MNE-RSA to perform an RSA analysis between the Wakeman & Nelson "faces" dataset(Wakeman & Henson, 2015) and FaceNet(Schroff et al., 2015) embedding vectors. The RSA is performed using a searchlight across the cortical surface and a sliding window across time. The result is shown in Figure 2.

```python
import os

import mne_rsa
import numpy as np
import pooch
from mne import read_epochs
from mne.minimum_norm import apply_inverse_epochs, read_inverse_operator

# Download the Wakeman & Nelson (2015) "faces" dataset.
pooch.retrieve(
    "https://tinyurl.com/3rr3vb6u",
    known_hash="7c2c1f220b7519798534769e10cabd43690a46dca280729b0fea0599b78d142b",
    processor=pooch.Unzip(extract_dir=os.getcwd()),
)

# Use FaceNet embeddings to compute the model RDM.
facenet = np.load("./data/stimuli/facenet_embeddings.npz")
model_rdm = mne_rsa.compute_rdm(facenet["embeddings"])

# Load MEG epochs and perform source estimation.
epochs = read_epochs("./data/sub-02/sub-02-epo.fif")
inv = read_inverse_operator("./data/sub-02/sub-02-inv.fif")
stcs = apply_inverse_epochs(epochs, inv, lambda2=1 / 9, pick_ori="normal")

# Use MNE-RSA to perform RSA analysis using a seachlight.
stc_rsa = mne_rsa.rsa_stcs(
    stcs,  # data to perform RSA on
    model_rdm,  # model RDM
    labels_stcs=epochs.metadata.file,  # align epochs and embedding vectors
    labels_rdm_model=facenet["filenames"],
    tmin=0,  # restrict the analysis to a time window
    tmax=0.5,
    spatial_radius=0.02,  # use a searchlight across the cortex
    src=inv["src"],  # to compute geodesic distance
    temporal_radius=0.05,  # use a sliding window across time
    verbose=True,  # display a progress bar
    n_jobs=-1,  # use all available CPU cores
)

# Plot the result using MNE-Python.
stc_rsa.plot(
    "sub-02",
    subjects_dir="./data/freesurfer",
    initial_time=0.193,
    hemi="both",
    views="caudal",
    background="white",
)
```

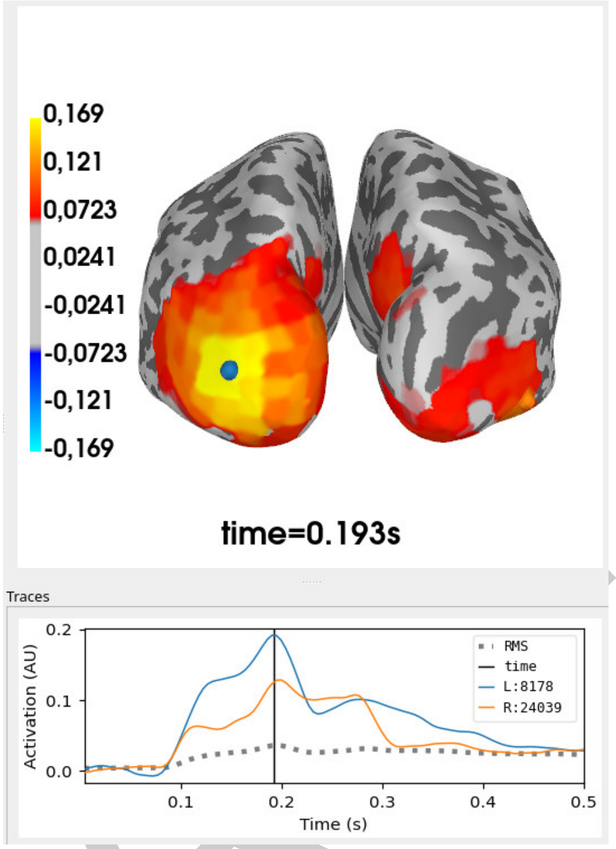

Figure 2: Result of a RSA performed with a sliding window across time.

Performance

Performing RSA in a searchlight pattern will produce tens of thousands of RDMs that can take up multiple gigabytes of space. For memory efficiency, RDMs are never kept in memory longer than they need to be, hence the useage of python generators. It is almost always easier to re-compute RDMs than it is to write them to disk and later read them back in. The computation of RDMs is parallelized across CPU cores.

Statement of need

While the core computations behind RSA are simple, getting the details right is hard. Creating a "searchlight" patches across the cortex means using geodesic rather than Euclidean distance (Figure 3), combining MEG gradiometers and magnetometers requires signal whitening, creating proper evoked responses requires averaging across stimulus repetitions, and creating reliable brain RDMs requires cross-validated distance metrics (Guggenmos et al., 2018). MNE-RSA provides turn-key solutions for all of these details by interfacing with the metadata available in MNE-Python objects.

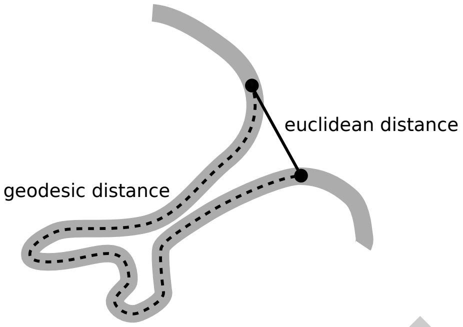

Figure 3: Depiction of geodesic versus Euclidean distance between points along the cortex.

At the time of writing, MNE-RSA has been used in five studies, two of which involve the authors (Ghazaryan et al., 2023; Hultén et al., 2021; Klimovich-Gray et al., 2021; Messi & Pylkkanen, 2025; Xu et al., 2024).

Software ecosystem

The scope of MNE-RSA is to to add RSA capabilities to MNE-Python and as such is geared towards users who are analyzing EEG/MEG data in Python. It provides a streamlined user experience for the most common use cases in this domain.

For users of MATLAB toolboxes such as FieldTrip (https://www.fieldtriptoolbox.org), Brainstorm (https://neuroimage.usc.edu/brainstorm) or EEGLab (https://sccn.ucsd.edu/eeglab), the original RSA-toolbox (https://github.com/rsagroup/rsatoolbox_matlab) may be a good choice. The original RSA-toolbox was implemented in MATLAB, although the third iteration now implemented in python (Bosch et al., 2025). While its focus is mostly on fMRI analysis, the RSA-toolbox aims for a broad implementation of everything related to RSA and its documentation includes an MEG demo.

A python package worth mentioning is PyMVPA (Hanke et al., 2009), which implements a wide array of machine learning methods, including an RSA variant where RDMs are created using decoding performance as distance metric. While it is possible to use it for EEG and MEG analysis, it mostly focuses on fMRI.

Acknowledgements

Development of MNE-RSA was funded by the Research Council of Finland (grants #310988 and #343385 to M.v.V).

References

Bosch, J. J. F. van den, Golan, T., Peters, B., Taylor, J., Shahbazi, M., Lin, B., Charest, I., Diedrichsen, J., Kriegeskorte, N., Mur, M., & Schütt, H. H. (2025). *A python toolbox for representational similarity analysis* (p. 2025.05.22.655542). bioRxiv. https://doi.org/10.1101/2025.05.22.655542

Diedrichsen, J., & Kriegeskorte, N. (2017). Representational models: A common framework for understanding encoding, pattern-component, and representational-similarity analysis. *PLOS Computational Biology*, *13*(4), e1005508. https://doi.org/10.1371/journal.pcbi.1005508

Ghazaryan, G., van Vliet, M., Lammi, L., Lindh-Knuutila, T., Kivisaari, S., Hultén, A., & Salmelin, R. (2023). Cortical time-course of evidence accumulation during semantic processing. *Communications Biology*, *6*(1), 1–12. https://doi.org/10.1038/s42003-023-05611-6

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*(December), 1–13. https://doi.org/10.3389/fnins.2013.00267

Guggenmos, M., Sterzer, P., & Cichy, R. M. (2018). Multivariate pattern analysis for MEG: A comparison of dissimilarity measures. *NeuroImage*, *173*, 434–447. https://doi.org/10.1016/j.neuroimage.2018.02.044

Hanke, M., Halchenko, Y. O., Sederberg, P. B., Hanson, S. J., Haxby, J. V., & Pollmann, S. (2009). PyMVPA: A python toolbox for multivariate pattern analysis of fMRI data. *Neuroinformatics*, *7*(1), 37–53. https://doi.org/10.1007/s12021-008-9041-y

Hultén, A., van Vliet, M., Kivisaari, S., Lammi, L., Lindh-Knuutila, T., Faisal, A., & Salmelin, R. (2021). The neural representation of abstract words may arise through grounding word meaning in language itself. *Human Brain Mapping*, *42*(15), 4973–4984. https://doi.org/10.1002/hbm.25593

Klimovich-Gray, A., Barrena, A., Agirre, E., & Molinaro, N. (2021). One way or another: Cortical language areas flexibly adapt processing strategies to perceptual and contextual properties of speech. *Cerebral Cortex*, *31*(9), 4092–4103. https://doi.org/10.1093/cercor/bhab071

Kriegeskorte, N., Mur, M., & Bandettini, P. A. (2008). Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, *2*(November), 4. https://doi.org/10.3389/neuro.06.004.2008

Messi, A.-P., & Pylkkanen, L. (2025). Tracking neural correlates of contextualized meanings with representational similarity analysis. *Journal of Neuroscience*, *45*(19). https://doi.org/10.1523/jneurosci.0409-24.2025

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823. https://doi.org/10.1109/CVPR.2015.7298682

Wakeman, D. G., & Henson, R. N. (2015). A multi-subject, multi-modal human neuroimaging dataset. *Scientific Data*, *2*(1), 150001. https://doi.org/10.1038/sdata.2015.1

Xu, W., Li, X., Parviainen, T., & Nokia, M. (2024). Neural correlates of retrospective memory confidence during face–name associative learning. *Cerebral Cortex*, *34*(5), bhae194. https://doi.org/10.1093/cercor/bhae194