

MEG/EEG/iEEG/NIRS data processing in Python

<http://mne.tools>

Alexandre Gramfort

<http://alexandre.gramfort.net>

GitHub : @agramfort 

Twitter : @agramfort 

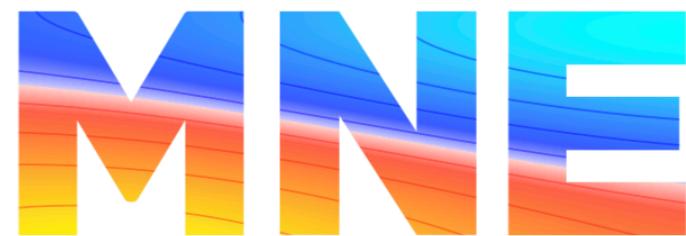
<https://github.com/maansidesai/WIRED-2024-Paris/>

MNE software for processing MEG and EEG data, A. Gramfort, M. Luessi, E. Larson, D. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, M. Hämäläinen, Neuroimage, 2014



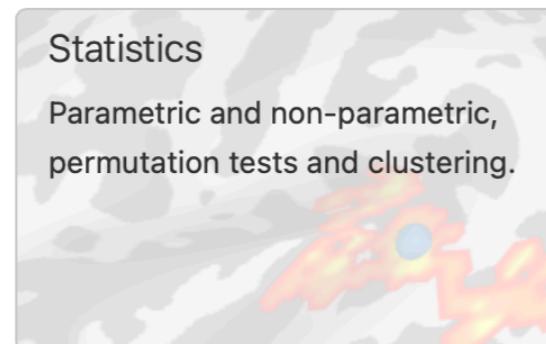
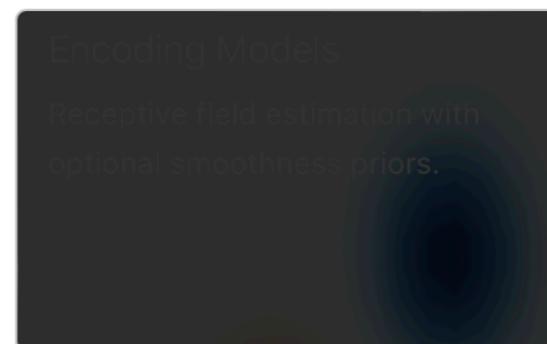
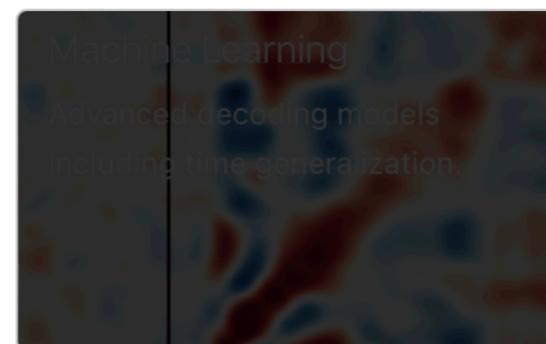
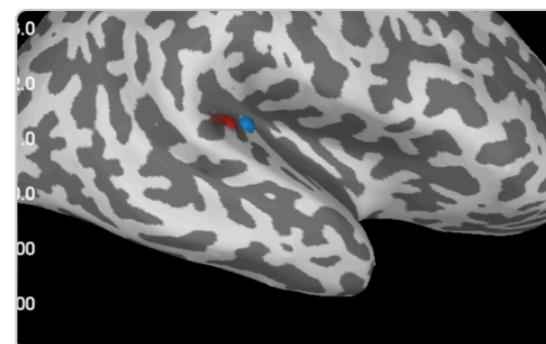
Version 1.6.1

- Tutorials
- Changelog
- Get help
- Cite
- Contribute



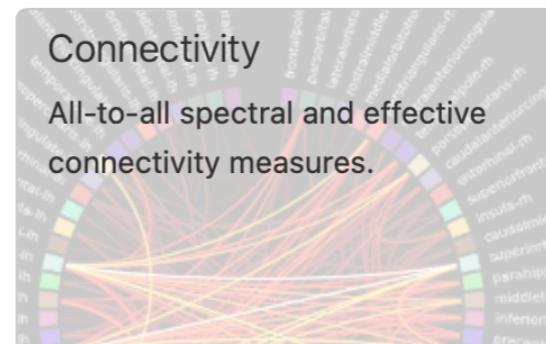
MEG + EEG ANALYSIS & VISUALIZATION

Open-source Python package for exploring, visualizing, and analyzing human neurophysiological data: MEG, EEG, sEEG, ECoG, NIRS, and more.



Statistics

Parametric and non-parametric, permutation tests and clustering.



Connectivity

All-to-all spectral and effective connectivity measures.



Data Visualization

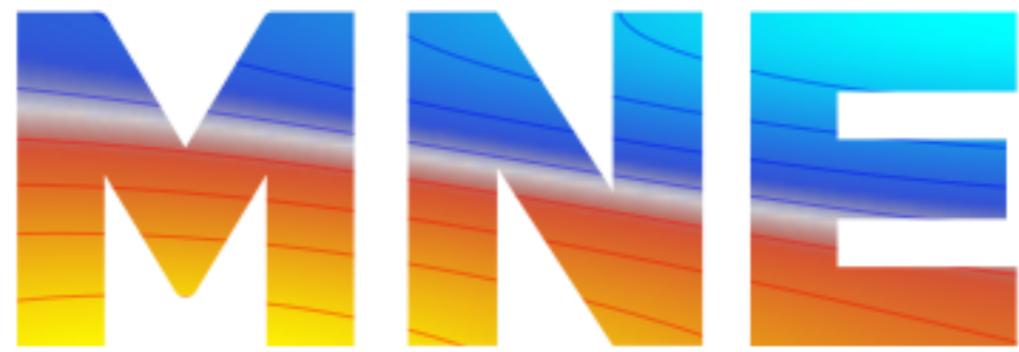
Explore your data from multiple perspectives.

Funders



Supporting institutions





built on top of the scientific Python ecosystem



<https://github.com/mne-tools/mne-python>

mne-tools / mne-python

Code Issues 456 Pull requests 66 Actions Projects 3 Wiki Security Insights

mne-python Public Edit Pins Unwatch 84 Fork 1.3k Starred 2.5k

main Go to file + <> Code About

mscheltienne Fix cleaning of channel ... 9006789 · 2 hours ago 17,877 Commits

.circleci Unwatch 84 Fork 1.3k Starred 2.5k

.github doc examples logo mne tools tutorials .coveragerc

Fix cleaning of channel names which c... 2 hours ago

Bump ruff to v0.3.0 (#12477) last week

MAINT: Add copyright to the top of ea... 4 months ago

Fix cleaning of channel names which c... 2 hours ago

MAINT: Include OpenMEEG in pre jobs... last week

Bump ruff to v0.3.0 (#12477) last week

MAINT: No longer bundle packages (#... 2 years ago

MNE:
Magnetoencephalography
(MEG) and

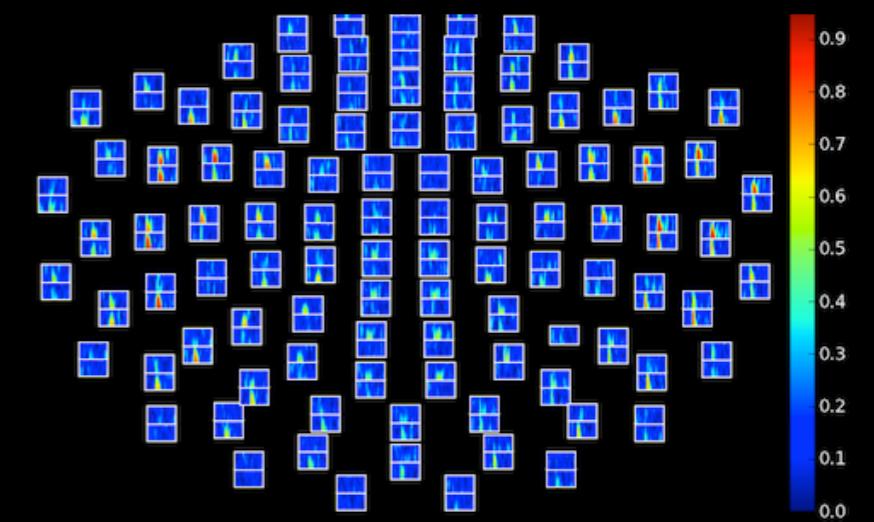
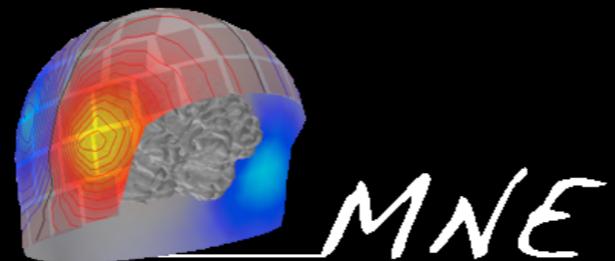
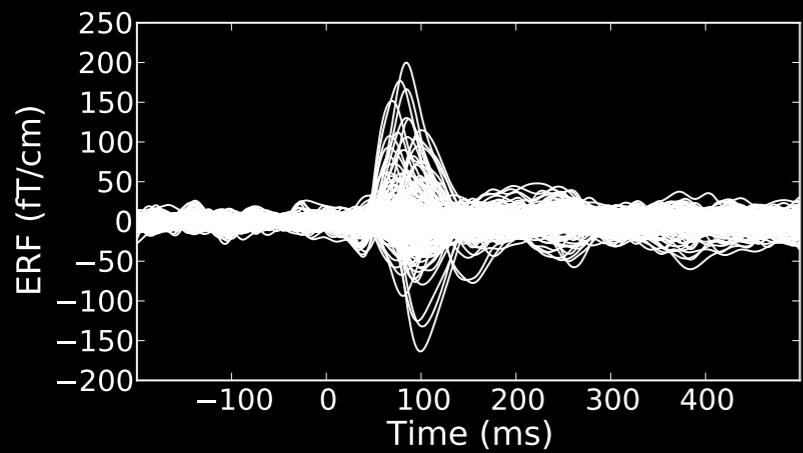
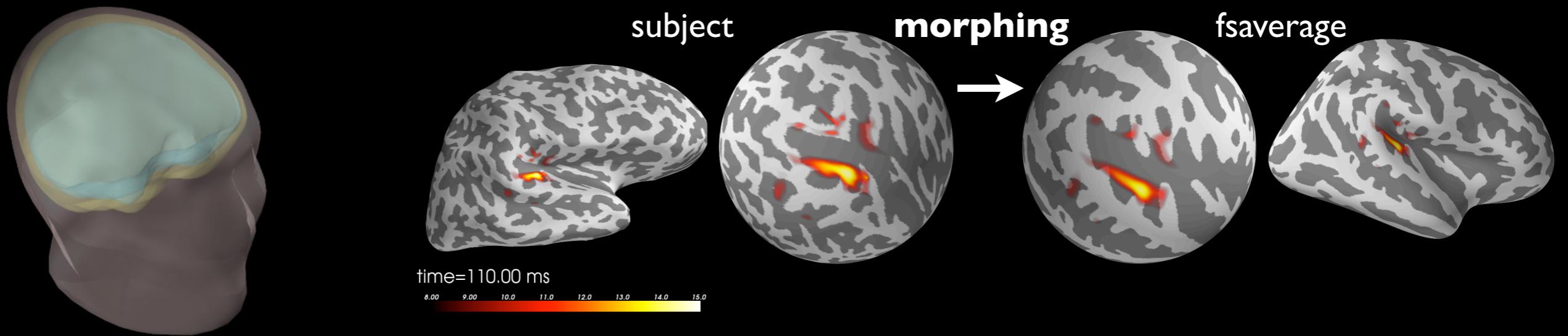
mne.tools

visualization python
machine-learning statistics
neuroscience meg eeg
neuroimaging
electroencephalography
magnetoencephalography
electrocorticography ecog

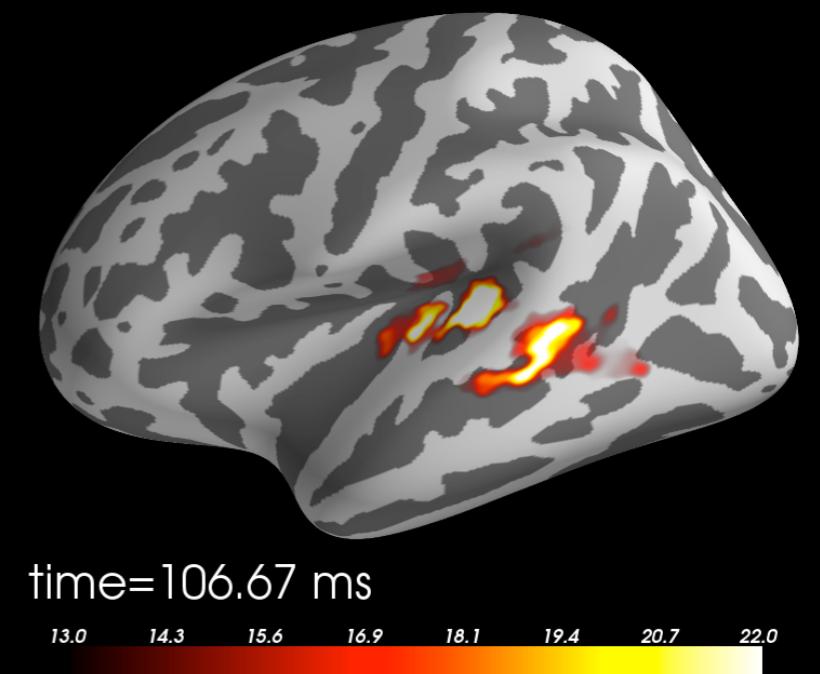
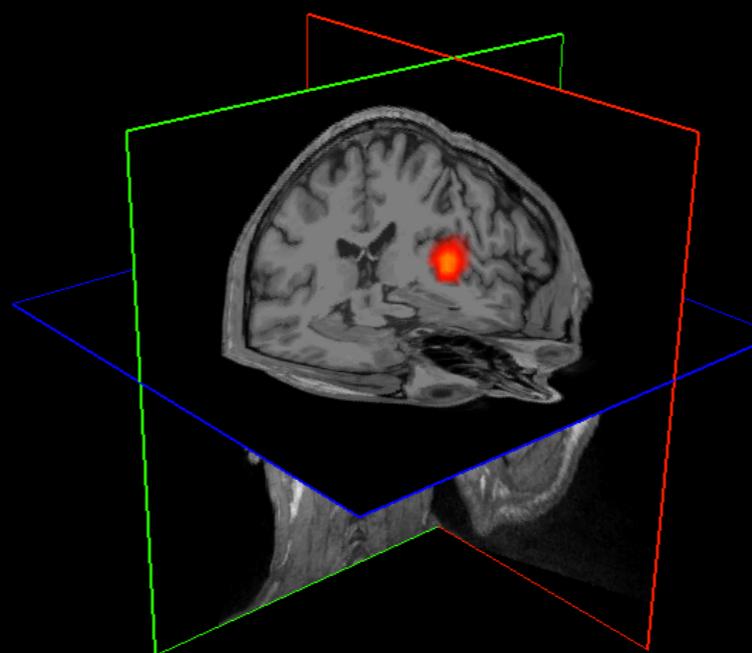
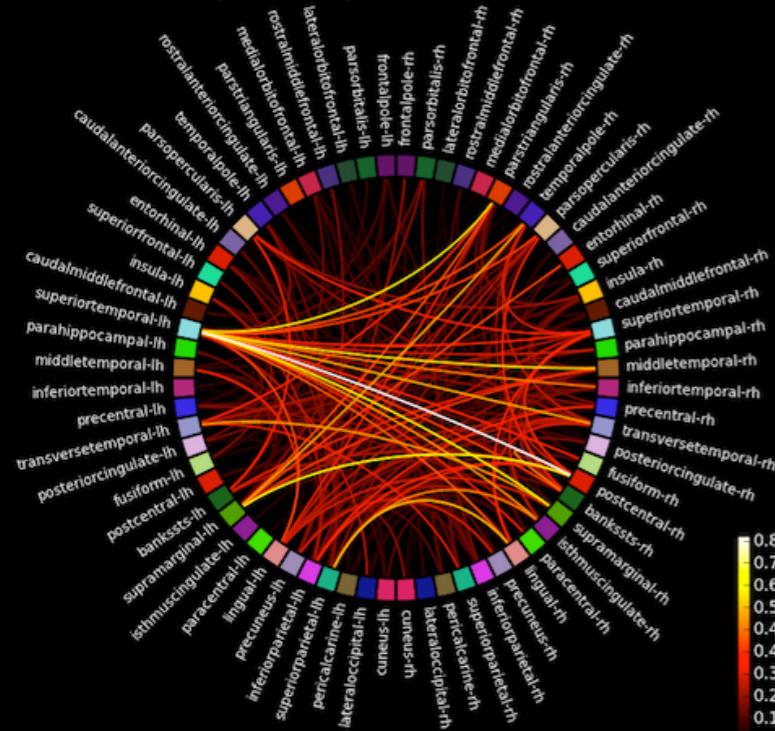
Readme
BSD-3-Clause license

A young girl with light brown hair in two pigtails, wearing a pink long-sleeved shirt, is looking upwards with her arms outstretched. An orange speech bubble originates from her hands and contains the text.

What can I do
with MNE?



All-to-All Connectivity left-Auditory Condition



Section Navigation

Tutorials

- ^ Introductory tutorials
- ^ Reading data for different recording systems
- ^ Working with continuous data
- ^ Preprocessing
- ^ Segmenting continuous data into epochs
- ^ Estimating evoked responses
- ^ Time-frequency analysis
- ^ Forward models and source spaces
- ^ Source localization and inverses
- ^ Statistical analysis of sensor data
- ^ Statistical analysis of source estimates
- ^ Machine learning models of neural activity
- ^ Clinical applications
- ^ Simulation
- ^ Visualization tutorials
- ^ Examples
- Glossary
- Implementation details
- Design philosophy
- Example datasets
- Command-line tools
- Migrating from other analysis software
- The typical M/EEG workflow

Tutorials

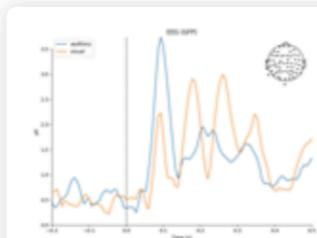
These tutorials provide narrative explanations, sample code, and expected output for the most common MNE-Python analysis tasks. The emphasis here is on thorough explanations that get you up to speed quickly, at the expense of covering only a limited number of topics. The sections and tutorials are arranged in a fixed order, so in theory a new user should be able to progress through in order without encountering any cases where background knowledge is assumed and unexplained. More experienced users (i.e., those with significant experience analyzing EEG/MEG signals with different software) can probably skip around to just the topics they need without too much trouble.

Note

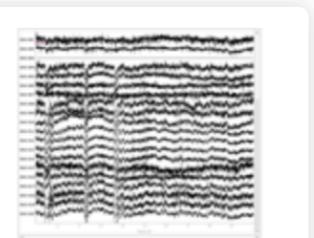
If tutorial-scripts contain plots and are run locally, using the interactive flag with `python -i tutorial_script.py` keeps them open.

Introductory tutorials

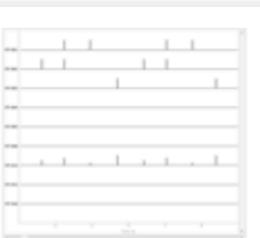
These tutorials cover the basic EEG/MEG pipeline for event-related analysis, introduce the [mne.Info](#), [events](#), and [mne.Annotations](#) data structures, discuss how sensor locations are handled, and introduce some of the configuration options available.



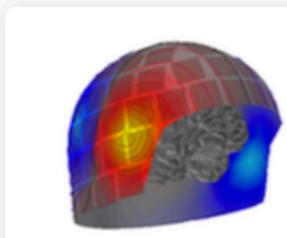
Overview of MEG/EEG analysis with MNE-Python



Modifying data in-place



Parsing events from raw data



The Info data structure

On this page

- Introductory tutorials
- Reading data for different recording systems
- Working with continuous data
- Preprocessing
- Segmenting continuous data into epochs
- Estimating evoked responses
- Time-frequency analysis
- Forward models and source spaces
- Source localization and inverses
- Statistical analysis of sensor data
- Statistical analysis of source estimates
- Machine learning models of neural activity
- Clinical applications
- Simulation
- Visualization tutorials

[Edit on GitHub](#)

Section Navigation

Tutorials

- ^ Introductory tutorials
- ^ Reading data for different recording systems
- ^ Working with continuous data
- ^ Preprocessing
- ^ Segmenting continuous data into epochs
- ^ Estimating evoked responses
- ^ Time-frequency analysis
- ^ Forward models and source spaces
- ^ Source localization and inverses
- ^ Statistical analysis of sensor data
- ^ Statistical analysis of source estimates
- ^ Machine learning models of neural activity
- ^ Clinical applications
- ^ Simulation
- ^ Visualization tutorials
- Examples
- Glossary
- Implementation details
- Design philosophy
- Example datasets
- Command-line tools
- Migrating from other analysis software
- The typical M/EEG workflow

Tutorials

These tutorials provide narrative explanations, sample code, and expected output for the most common MNE-Python analysis tasks. The emphasis here is on thorough explanations that get you up to speed quickly, at the expense of covering only a limited number of topics. The sections and tutorials are arranged in a fixed order, so in theory a new user should be able to progress through in order without encountering any cases where background knowledge is assumed and unexplained. More experienced users (i.e., those with significant experience analyzing EEG/MEG signals with different software) can probably skip around to just the topics they need without too much trouble.

Note

If tutorial-scripts contain plots and are run locally, using the interactive flag with `python -i tutorial_script.py` keeps them open.

Introductory tutorials

These tutorials cover the basic EEG/MEG pipeline for event-related analysis, introduce the [mne.Info](#),

Total running time of the script: (0 minutes 46.432 seconds)

Estimated memory usage: 772 MB

[Download Jupyter notebook: 10_overview.ipynb](#)

[Download Python source code: 10_overview.py](#)

< Previous
[Introductory tutorials](#)

Next >
[Modifying data in-place](#)

On this page

- Introductory tutorials
- Reading data for different recording systems
- Working with continuous data
- Preprocessing
- Segmenting continuous data into epochs
- Estimating evoked responses
- Time-frequency analysis
- Forward models and source spaces
- Source localization and inverses
- Statistical analysis of sensor data
- Statistical analysis of source estimates
- Machine learning models of neural activity

A close-up photograph of a young girl with light brown hair styled in two pigtails. She is wearing a long-sleeved pink shirt. Her arms are extended wide to her sides, palms facing forward. She is looking upwards and slightly to the right with a neutral expression. The background is plain white.

And if I need
help?

Getting help

<https://mne.discourse.group/>

MNE

Topics More Categories Support & Discussions Announcements Job Opportunities Mailing List Archive (re... All categories Tags list-archive eeg preprocessing visualization epochs All tags

categories ► tags ► Categories Latest Top

Category	Topics	Latest
Support & Discussions Post your questions here – ideally after searching for existing answers!	52 / month	 MNE-Python 1.6.0 released on 2023-11-20 Announcements Dec 2023
Announcements Announcements of new software releases, workshops, etc.	1 / month	 How to correctly format your question Support & Discussions Nov 2023
Job Opportunities This is the place to post about interesting job opportunities.	1 / month	 MNE Office Hours on Discord Announcements Aug 2023
Mailing List Archive (rea... An archive of the mne_analysis mailing list.	2.4k	 Welcome to the MNE Forum Announcements Dec 2020
K Repeated measures ANOVA permutation test for multiple subjects?	2h	 Support & Discussions

A young girl with light brown hair in two pigtails, wearing a pink long-sleeved shirt, is looking upwards with her arms outstretched. An orange speech bubble originates from her right arm.

And if I want to
help?

Section Navigation

Contributing guide

What's new

Roadmap

Project Governance

Contributing guide

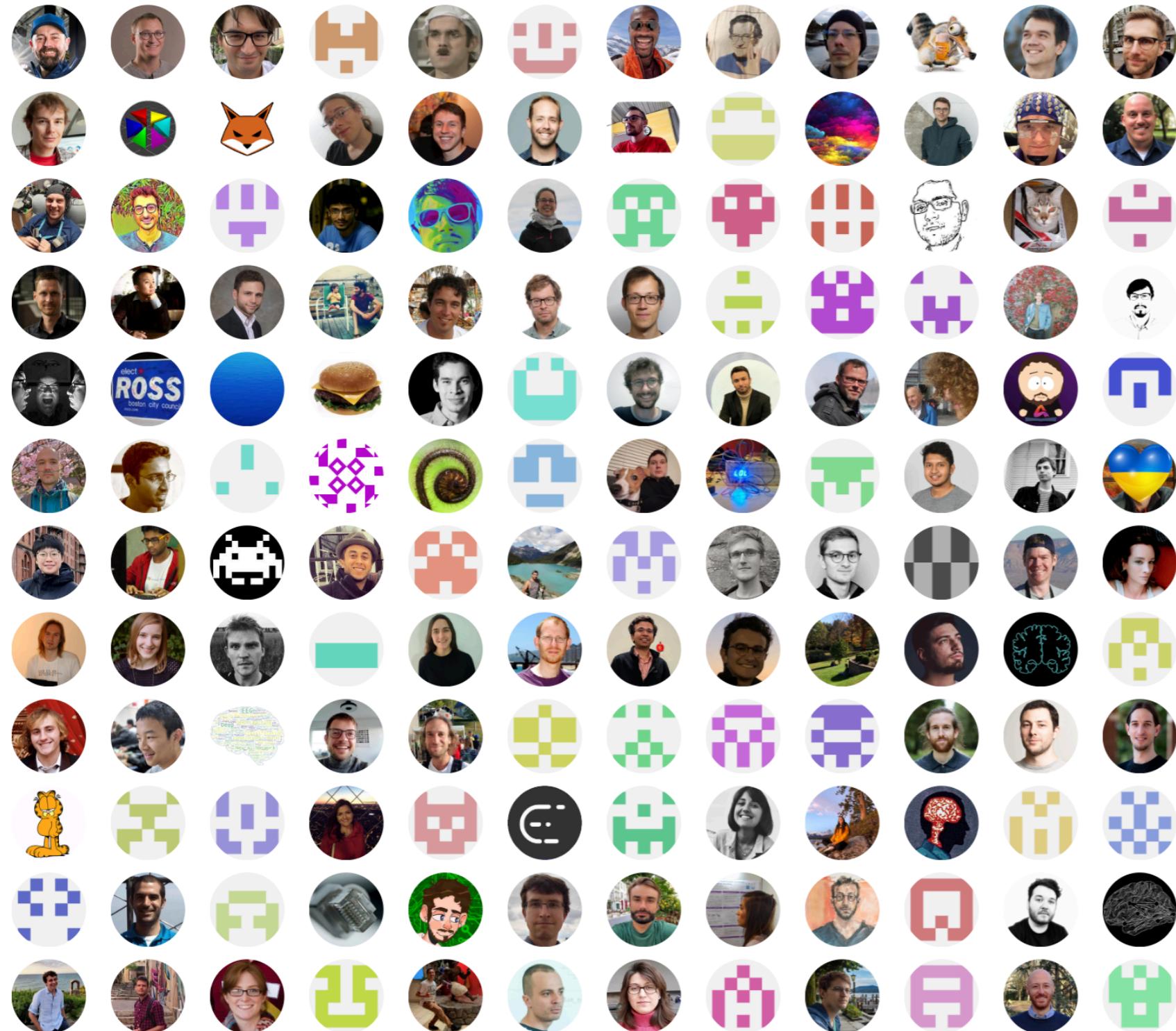
Thanks for taking the time to contribute! MNE-Python is an open-source project sustained mostly by volunteer effort. We welcome contributions from anyone as long as they abide by our [Code of Conduct](#).

There are lots of ways to contribute, such as:

- ⌚ Use the software, and when you find bugs, tell us about them! We can only fix the bugs we know about.
- 💬 Answer questions on [our user forum](#).
- 💬 Tell us about parts of the documentation that you find confusing or unclear.
- ✋ Tell us about things you wish MNE-Python could do, or things it can do but you wish they were easier.
- ♿ Improve the accessibility of our website.
- 🛠 Fix bugs.
- 🛠 Fix mistakes in our function documentation strings.
- 🛠 Implement new features.
- ✍ Improve existing tutorials or write new ones.
- ✚ Contribute to one of the many Python packages that MNE-Python depends on.

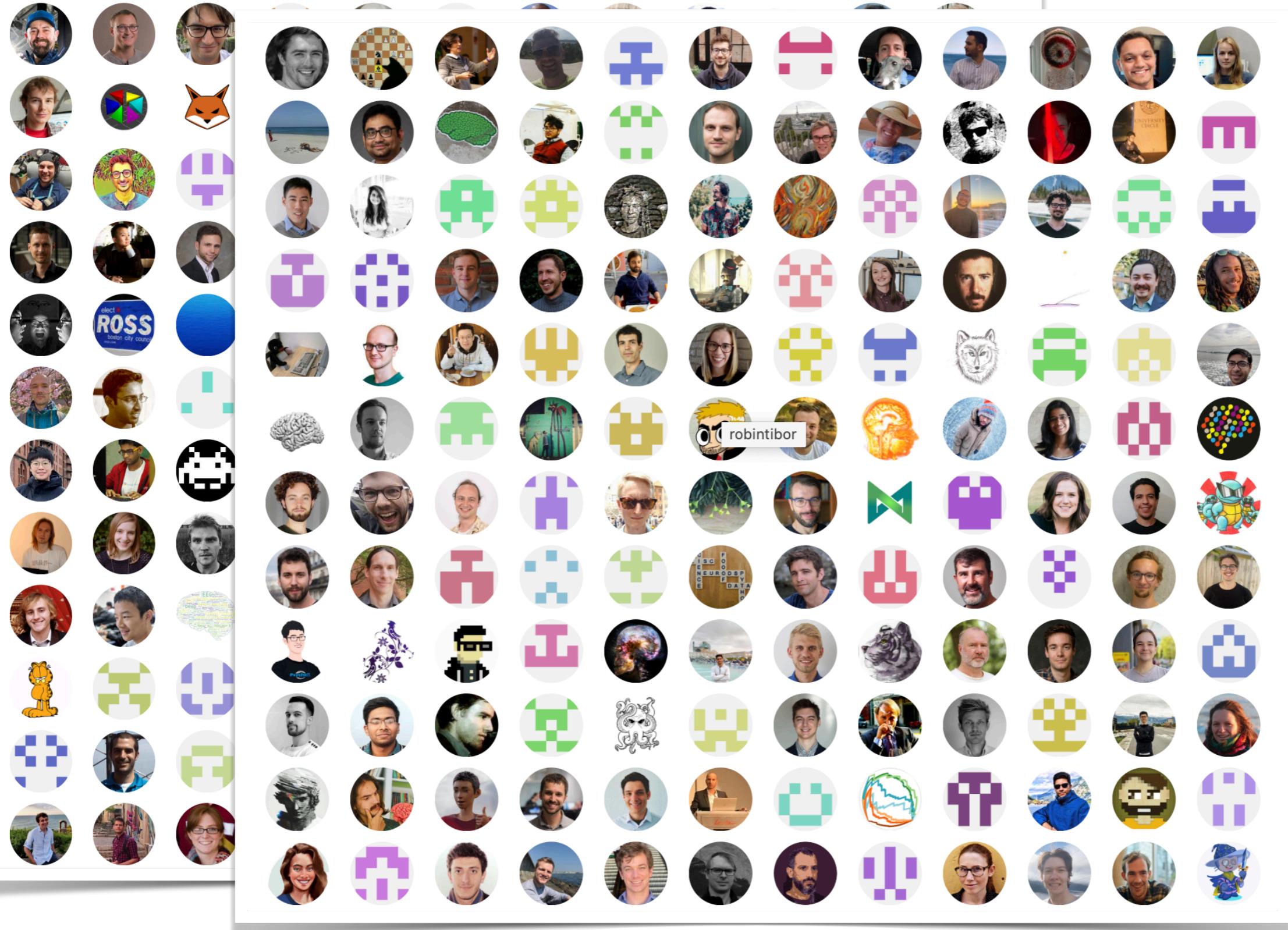
To *report bugs*, *request new features*, or *ask about* confusing documentation, it's usually best to open a new issue on [our user forum](#) first; you'll probably get help fastest that way, and it helps keep our GitHub issue tracker focused on things that we *know* will require changes to our software (as opposed to problems that can be fixed in the user's code). We may ultimately ask you to open an issue on GitHub too, but starting on the forum helps us keep things organized. For fastest results, be sure to include information about your operating system and MNE-Python version, and (if applicable) include a reproducible code sample that is as short as possible and ideally uses one of [our example datasets](#).

MNE People



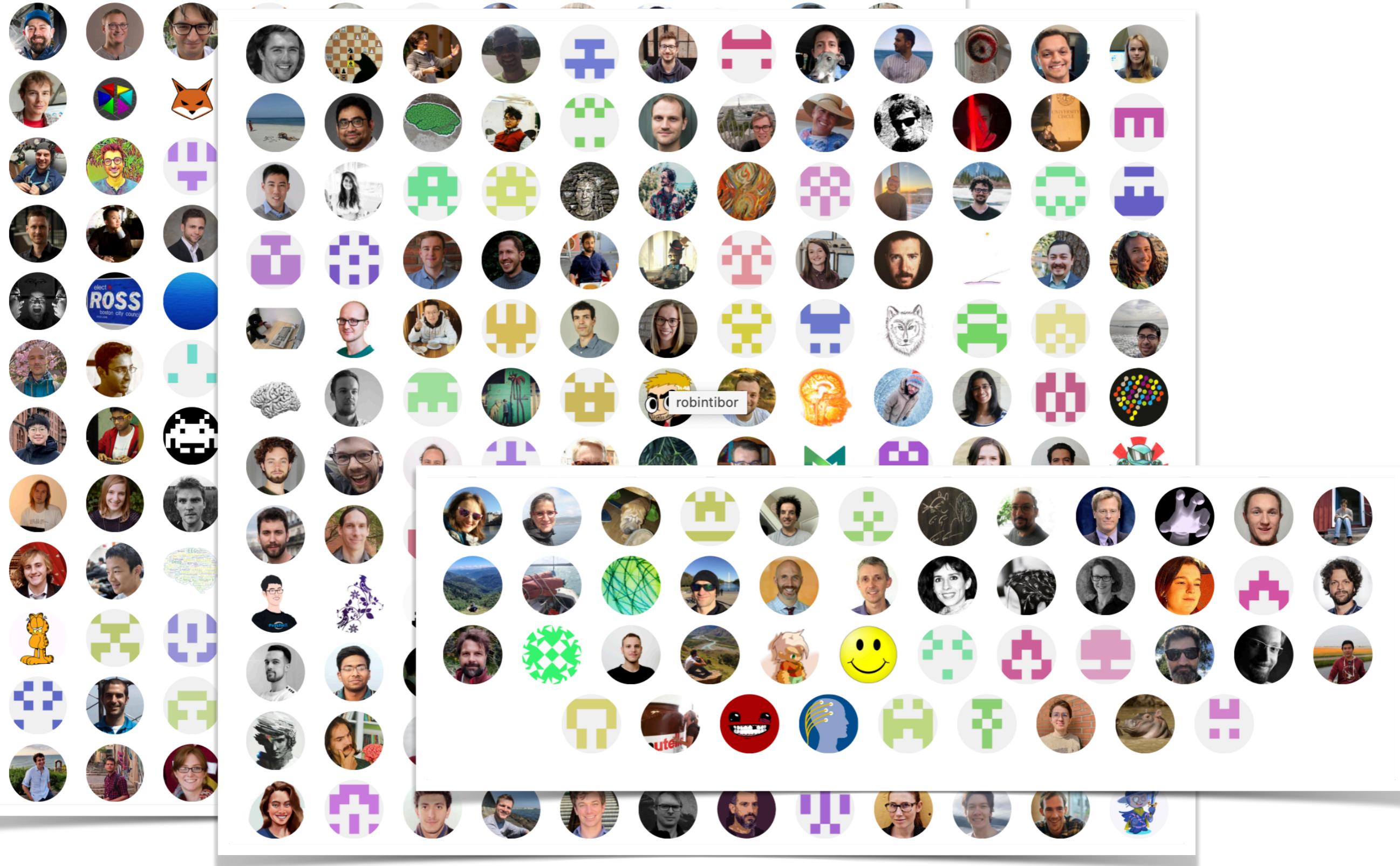
<https://github.com/mne-tools/mne-python/graphs/contributors>

MNE People



<https://github.com/mne-tools/mne-python/graphs/contributors>

MNE People



<https://github.com/mne-tools/mne-python/graphs/contributors>

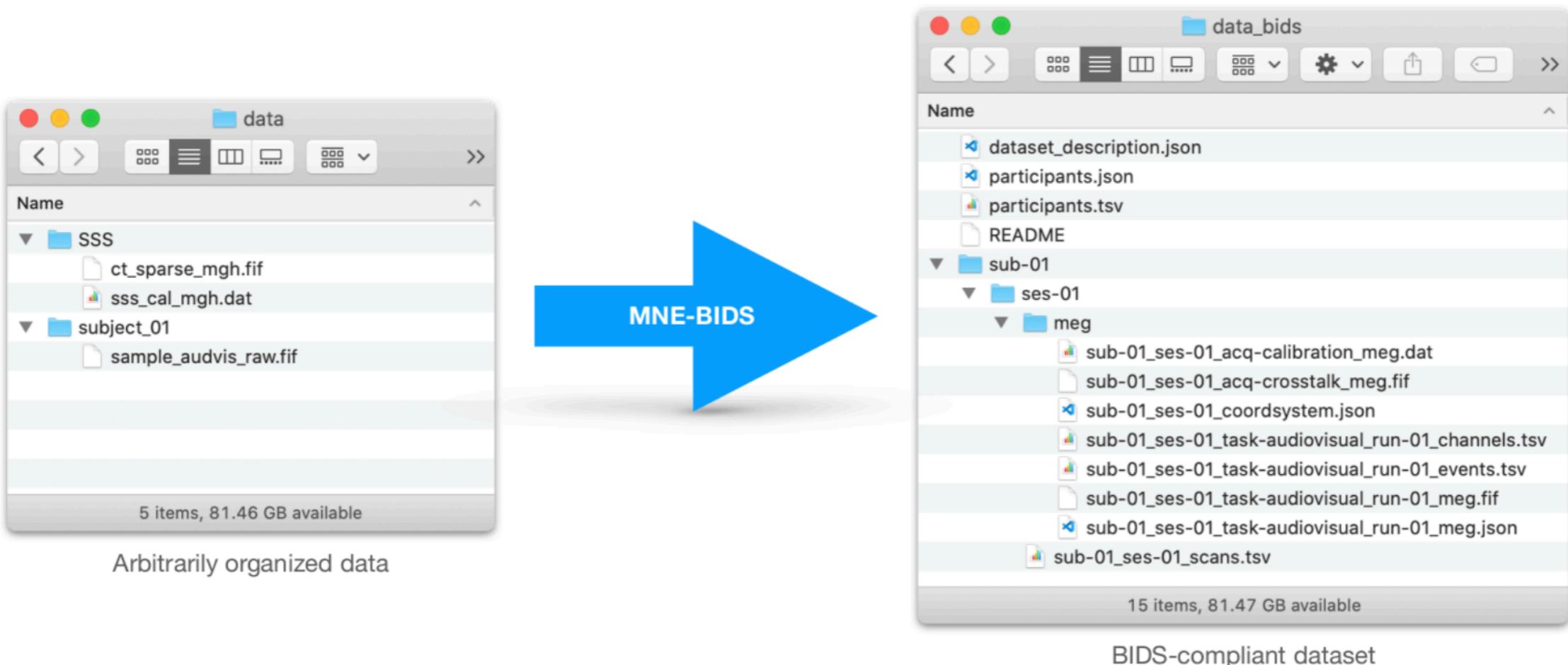
This is it?





MNE-BIDS

MNE-BIDS is a Python package that allows you to read and write [BIDS](#)-compatible datasets with the help of [MNE-Python](#).



Why?

MNE-BIDS links BIDS and MNE-Python with the goal to make your analyses faster to code, more robust, and facilitate data and code sharing with co-workers and collaborators.

What is MNE-BIDS-Pipeline?

MNE-BIDS-Pipeline is a full-fledged processing pipeline for your MEG and EEG data.

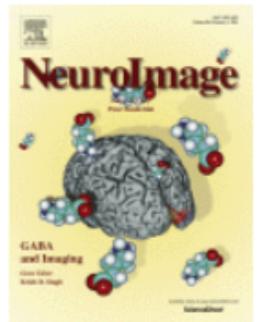
- It operates on data stored according to the Brain Imaging Data Structure (BIDS).
- Under the hood, it uses [MNE-Python](#).

[Get started](#)[Learn more](#)

<https://mne.tools/mne-bids-pipeline/stable/>

What the pipeline offers

- Automated processing of MEG and EEG data from raw data to inverse solutions.
- Configuration via a simple text file.
- Extensive processing and analysis summary reports.
- Process just a single participant, or as many as several hundreds of participants – in parallel.
- Execution via an easy-to-use command-line utility.
- Helpful error messages in case something goes wrong.
- Data processing as a sequence of standard processing steps.
- Steps are cached to avoid unnecessary recomputation.
- Data can be "ejected" from the pipeline at any stage. No lock-in!
- Runs on your laptop, on a powerful server, or on a high-performance cluster via Dash.



MNE software for processing MEG and EEG data

Alexandre Gramfort^{a, b, c, d, e}  , Martin Luessi^{b, c}, Eric Larson^f, Denis A. Engemann^{g, h}, Daniel Strohmeierⁱ, Christian Brodbeck^j, Lauri Parkkonen^{k, l}, Matti S. Hämäläinen^{b, c}

METHODS ARTICLE

Front. Neurosci., 26 December 2013 | <https://doi.org/10.3389/fnins.2013.00267>

MEG and EEG data analysis with MNE-Python

 **Alexandre Gramfort**^{1,2,3*},  **Martin Luessi**²,  **Eric Larson**⁴,  **Denis A. Engemann**^{5,6},  **Daniel Strohmeier**⁷,  **Christian Brodbeck**⁸,  **Roman Goj**⁹,  **Mainak Jas**^{10,11},  **Teon Brooks**⁸,  **Lauri Parkkonen**^{10,11} and  **Matti Hämäläinen**^{2,11}

Don't forget to give us academic credit



> J Open Source Softw. 2019;4(44):1896. doi: 10.21105/joss.01896. Epub 2019 Dec 18.

MNE-BIDS: Organizing electrophysiological data into the BIDS format and facilitating their analysis

Stefan Appelhoff ¹, Matthew Sanderson ², Teon L Brooks ³, Marijn van Vliet ⁴,
Romain Quentin ⁵, Chris Holdgraf ⁶, Maximilien Chaumon ⁷, Ezequiel Mikulan ⁸,
Kambiz Tavabi ⁹, Richard Höchenberger ¹⁰, Dominik Welke ¹¹, Clemens Brunner ¹²,
Alexander P Rockhill ¹³, Eric Larson ⁹, Alexandre Gramfort ¹⁴, Mainak Jas ¹⁵

[J Open Source Softw.](#) Author manuscript; available in PMC 2022 Aug 18.

PMCID: PMC9387757

Published in final edited form as:

NIHMSID: NIHMS1780638

J Open Source Softw. 2022; 7(70): 3897.

PMID: [35992635](#)

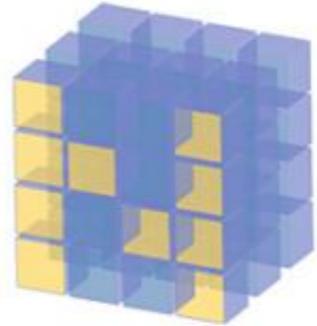
Published online 2022 Feb 14. doi: [10.21105/joss.03897](#)

Intracranial Electrode Location and Analysis in MNE-Python

Alexander P. Rockhill,¹ Eric Larson,² Brittany Stedelin,³ Alessandra Mantovani,³ Ahmed M. Raslan,³
Alexandre Gramfort,⁴ and Nicole C. Swann¹

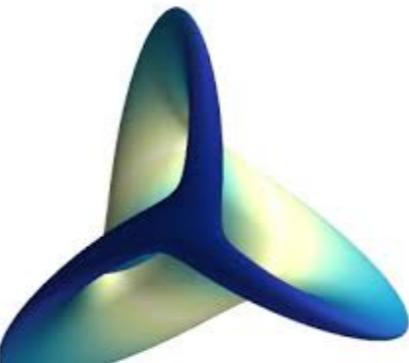
Don't forget to give us academic credit 

Thanks !



NumPy

matplotlib



Mayavi

GitHub



Travis CI



circleci



SPHINX
Sphinx-Gallery



ANACONDA®

Contact:

Alexandre Gramfort
<http://alexandre.gramfort.net>

GitHub : @agramfort



Twitter : @agramfort



```
>>> import mne
```

