

# Intracranial Electrode Location and Analysis in MNE-Python

Alexander P. Rockhill<sup>1</sup>, Eric Larson<sup>2</sup>, Brittany Stedelin<sup>3</sup>, Alessandra Mantovani<sup>3</sup>, Ahmed M. Raslan<sup>3</sup>, Alexandre Gramfort<sup>4</sup>, and Nicole C. Swann<sup>1</sup>

<sup>1</sup> Department of Human Physiology, University of Oregon, Eugene OR, USA <sup>2</sup> Institute for Learning and Brain Sciences, University of Washington, Seattle, WA, USA <sup>3</sup> Department of Neurological Surgery, Oregon Health & Science University, Portland, Oregon. <sup>4</sup> Université Paris-Saclay, Inria, CEA, Palaiseau, France

DOI: [10.21105/joss.03897](https://doi.org/10.21105/joss.03897)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Prashant K Jha](#) ↗

## Reviewers:

- [@TomDonoghue](#)
- [@janfreyberg](#)

Submitted: 22 October 2021

Published: 20 November 2021

## License

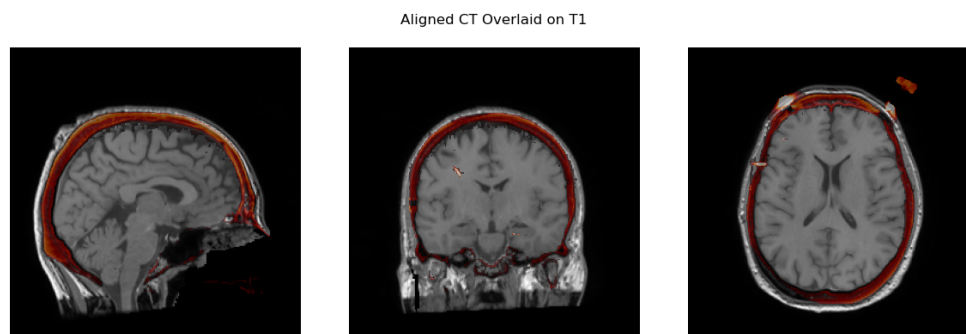
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

Intracranial electrophysiology analysis requires precise location and anatomical labeling of electrode recording contacts before a signal processing analysis of the data can be interpreted. Signal processing techniques are common to other electrophysiology modalities, such as magnetoencephalography (MEG) and electroencephalography (EEG), so ideally locating and labelling intracranial electrodes would be an integrated part of a single analysis software. We added intracranial electrode localization and intracranial-specific analyses to MNE-Python ([Gramfort et al., 2013](#)) so that its suite of signal processing and analysis tools can be used more easily by intracranial electrophysiology researchers.

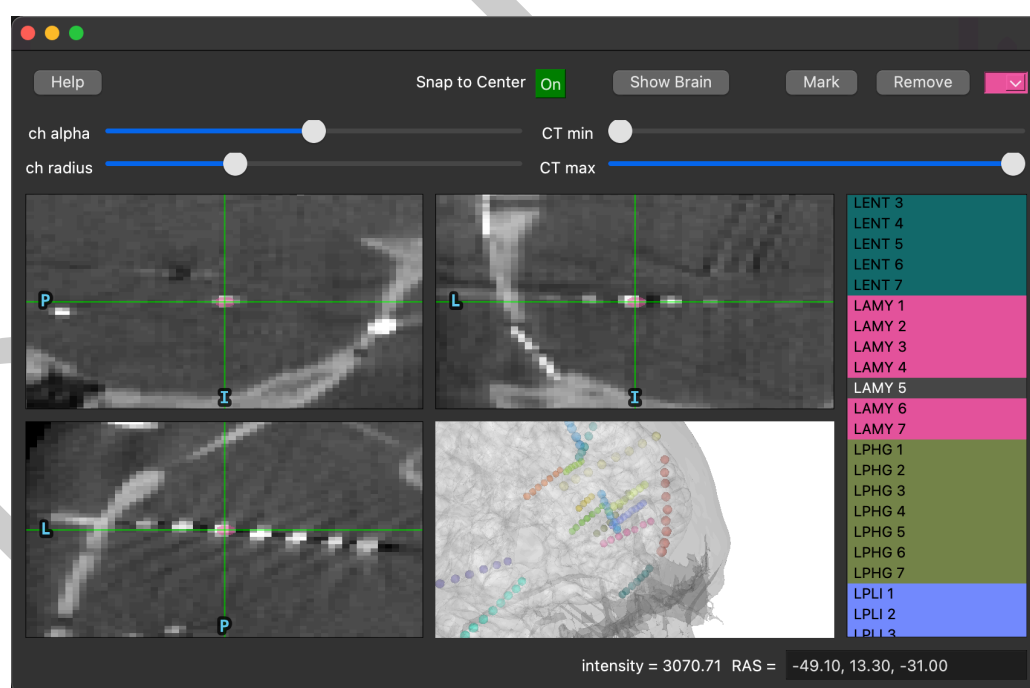
For brain electrophysiology research using stereoelectroencephalography (sEEG) and electrocorticography (ECoG), the raw electrophysiology, neurosurgical planning and neuroimaging needs to be integrated before meaningful interpretation of signal processing analyses can occur. Typically, for a patient with an intracranial electrophysiology recording, two anatomical images are collected. A preoperative magnetic resonance (MR) image is collected to obtain detailed individual brain anatomy, and a postoperative computed tomography (CT) image is collected to get a high-resolution image with 3D contact locations (although this can also be provided by a postoperative MR). The surgical plans then link the names of the contacts in the recording with their positions.

To integrate these electrophysiology and imaging data, first, the CT must be aligned to the MR in order to determine the positions of the electrode contacts in the CT relative to brain structures in the MR. In MNE-Python, these are aligned using a mutual information histogram-matching approach using DIPY ([Garyfallidis et al., 2014](#)) that works without manual pre-alignment using the function `mne.transforms.apply_volume_registration`.

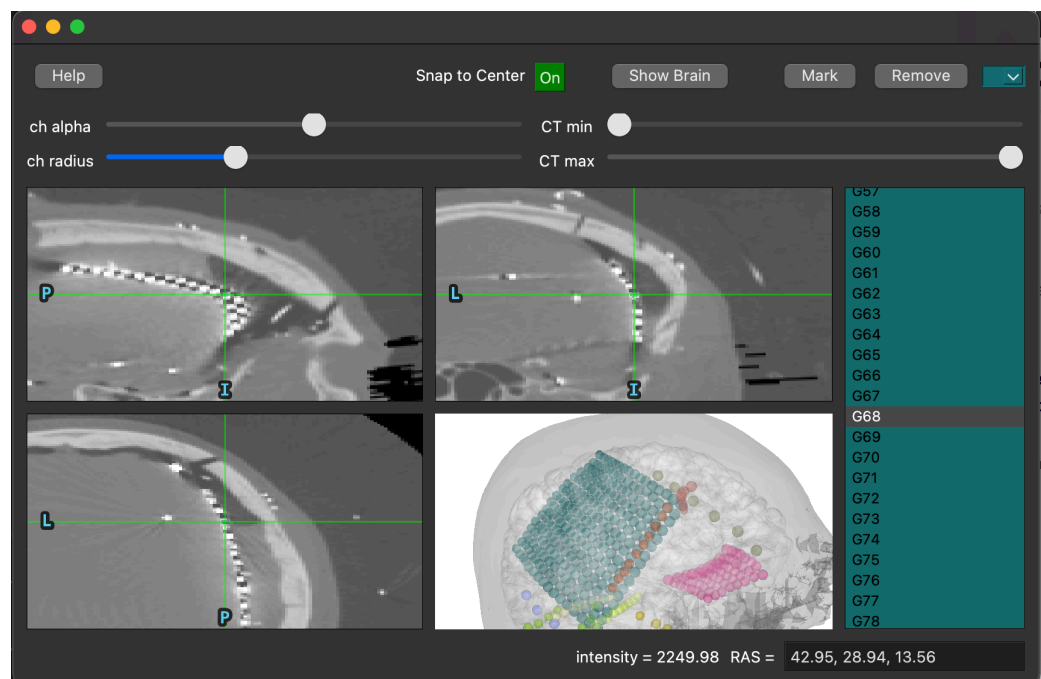


**Figure 1:** Figure 1. A CT image correctly aligned to the MR image. The darker areas in the MR are the skull whereas the lighter, outermost regions are subcutaneous fat. The CT is shown with a red scale, and red areas shown in the slice are the skull which has high intensity in the CT.

33 Next, electrode contact locations are selected in merged CT-MR coordinates using a graphical  
34 user interface (GUI) implemented in PyQt (`mne.gui.locate_ieeg`). In the GUI, users click  
35 or scroll through slices to find the location of contacts. When the user navigates to the location  
36 of a channel, the channel name can be selected from a menu listing the names extracted from  
37 the recording file. Then, the user can mark the channel as associated with that location. The  
38 channel is then rendered in the 3D view as well as colored on the slice plots.

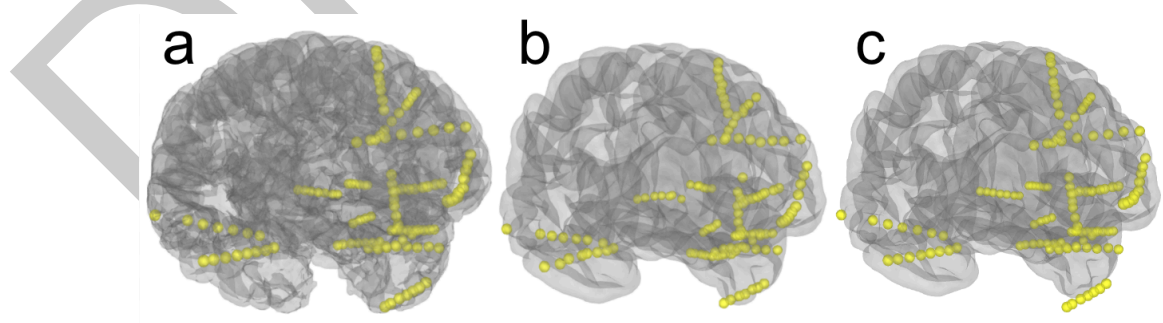


**Figure 2:** Figure 2. A screenshot of the GUI during a sEEG electrode location. For sEEG, the 3D brain view is typically similar to the surgical plans, making it easier to determine which electrode is which. With only 2D slice plots, locations and directions must be pieced together which is substantially more difficult.



**Figure 3:** Figure 3. A screenshot of the GUI during an ECoG electrode location. For ECoG, the 3D brain view makes it easier to select each contact in a large grid. Without the 3D view, it is difficult to continue in one row without jumping to another row incorrectly when viewing in 2D. These mistakes make it much slower to identify contacts.

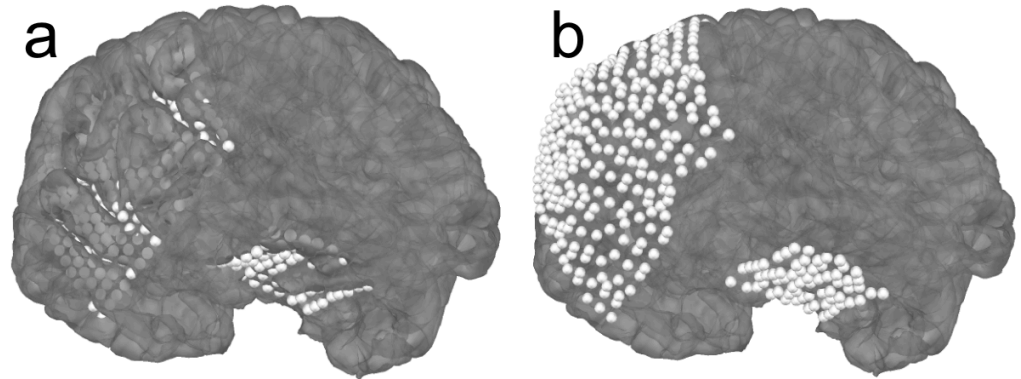
39 Located electrode contacts in merged CT-MR coordinates for an individual subject can be  
40 morphed to a template brain for group analysis using `mne.transforms.compute_volume_`  
41 `registration` and `mne.transforms.apply_volume_registration`. This is done using a  
42 symmetric diffeomorphic registration (SDR; Garyfallidis et al., 2014). This non-linear mapping  
43 tends to be more accurate than a linear transformation such as the Talairach transform (Dale  
44 et al., 1999) as shown in Figure 4.



**Figure 4:** Figure 4. a) Example sEEG electrodes in a 3D rendering of individual subject anatomy. b) The electrodes mapped to the Freesurfer fsaverage brain using the SDR morph. c) The electrodes mapped to the Freesurfer fsaverage brain using the Talairach transform. The linear transformation in c is less accurate than the SDR transform in b which can be seen by the electrode in the temporal pole floating outside the pial surface in c.

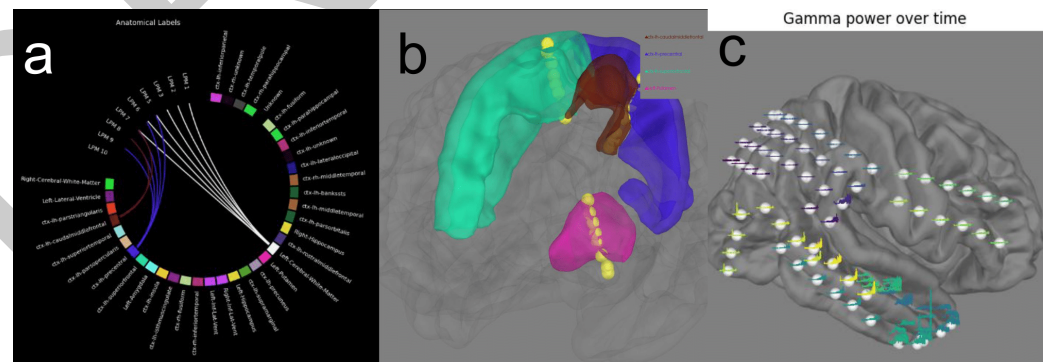
45 For ECoG electrodes, “brain shift,” caused by changes in pressure during the craniotomy,  
46 can also be accounted for using MNE-Python. This shift causes grid and strip electrodes to  
47 be deeper in the post-operative CT than they were in the preoperative MR. This makes it  
48 so that the electrodes are inside the pial surface in the preoperative MRI. In MNE-Python,

49 using `mne.preprocessing.ieeg.project_sensors_onto_brain`, this is compensated for  
50 by projecting the grid to the leptomenigeal surface.



**Figure 5:** Figure 5. a) A 3D rendering of an ECoG grid without correction for “brain shift.” In this case, the electrode contacts are under the pial surface of the preoperative brain because the change in pressure shifted the brain during the operation. b) The same 3D rendering with the “brain shift” correction.

51 Once the electrode locations are found both in relation to individual subject anatomy and  
52 in relation to a template brain, there are several visualization functions specific to sEEG  
53 and ECoG in MNE-Python. For sEEG, plotting the anatomical labels that the electrode  
54 shaft passes through indicates which areas are being recorded from using `mne.viz.plot_`  
55 `channel_labels_circle` (see Figure 6a). These areas can also be rendered in different  
56 colors in 3D for precise visualization of the trajectory and location of the electrodes using  
57 `mne.viz.Brain.add_volume_labels` as shown in Figure 6b. For ECoG, viewing a time  
58 series superimposed on a view of the 3D rendering enables the data to be displayed in relation  
59 to nearby channels using `mne.viz.snapshot_brain_montage` or `mne.stc_near_sensors`,  
60 as shown in Figure 6c.



**Figure 6:** Figure 6. a) The anatomical labels for an sEEG electrode shaft are shown as the contacts progress from deep (starting with 1) to superficial regions. b) The anatomical surfaces that sEEG electrodes pass through are rendered along with the trajectory of the electrode shaft. c) A time-frequency decomposition of the ECoG data rendered on top of a 3D image of the brain showing the location of the grid implant.

61 These intracranial location and analysis steps are shown in the MNE-Python tutorials “Lo-  
62 cating intracranial electrode contacts,” “Working with sEEG data” and “Working with ECoG  
63 data.”

## Statement of need

Integrating intracranial electrode location and analysis into MNE-Python allows researchers to go from raw data to completed analyses and visualizations of results in an all-in-one package that follows modern coding best practices, including unit tests, continuous integration, and thorough documentation. This integration provides immediate access to a wide variety of algorithms for different analyses of interest.

Previous work on intracranial software has typically come in the form of standalone packages (Groppe et al., 2017; Hamilton et al., 2017). However, these are difficult to maintain as the group of developers and number of users tends to be relatively smaller, thereby hampering software maintenance as the developers transition to other projects. MNE-Python is a general-purpose electrophysiology analysis package which makes it easier to retain a larger core group of developers and makes it more likely that this package and the intracranial functionality will be maintained and improved in the long-term. MNE-Python also has stability due to the funding it receives directly for development from institutions such as National Institutes of Health, the Chan-Zuckerberg open-source initiative, the European Research Council and Agence Nationale de la Recherche in France.

Other general purpose packages, written in MATLAB, provide similar functionality (Oostenveld et al., 2011; Tadel et al., 2011) but can be difficult for researchers to integrate into Python analyses and requires a MATLAB license. Compared to alternatives, the stability of the MNE-Python code base benefits from extensive unit tests and continuous integration across Windows, Mac OS and Ubuntu operating systems. Previous work in Python also provides similar functionality (Hamilton et al., 2017) but was not designed for sEEG electrode location. Adding the ability to control various aspects of the visualization (e.g., electrode contact marker size, opacity, multiple views, etc.) makes this task much easier, both for sEEG and ECoG. Additionally, compared to this previous work, MNE-Python has added key features. First it utilizes an SDR morph, which is orders of magnitude faster (approximately 15 minutes compared to 15 hours) and is comparably accurate (Avants et al., 2008). Second, it has a “snap to center” of the electrode contact feature to both increase accuracy and repeatability. This feature uses the center of mass of all voxels that monotonically decrease from the highest intensity voxel nearby the selected location. Next, it has an integrated 3D view which updates upon selection. Finally, it uses a predetermined list of channel names from the recording file to speed up the localization process and to eliminate matching errors. In general, the intracranial electrode location and analysis in MNE-Python is implemented using coding best-practices, it is feature complete for intracranial analysis, and it has user-friendly features. Importantly it is poised to be a well-maintained tool into the future.

## Acknowledgements

We acknowledge Liberty Hamilton for her work (Hamilton et al., 2017) which was especially helpful to use as a comparison and to build off conceptually in the development of this project.

## References

- Avants, B. B., Epstein, C. L., Grossman, M., & Gee, J. C. (2008). Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Medical Image Analysis*, 12(1), 26–41. <https://doi.org/10.1016/j.media.2007.06.004>
- Dale, A. M., Fischl, B., & Sereno, M. I. (1999). Cortical Surface-Based Analysis. *NeuroImage*, 9(2), 179–194. <https://doi.org/10.1006/nimg.1998.0395>



- 109 Garyfallidis, E., Brett, M., Amirbekian, B., Rokem, A., Walt, S. van der, Descoteaux, M.,  
110 Nimmo-Smith, I., & Dipy Contributors. (2014). Dipy, a library for the analysis of diffusion  
111 MRI data. *Frontiers in Neuroinformatics*, 8. <https://doi.org/10.3389/fninf.2014.00008>
- 112 Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj,  
113 R., Jas, M., Brooks, T., Parkkonen, L., & Hämäläinen, M. S. (2013). MEG and EEG data  
114 analysis with MNE-Python. *Frontiers in Neuroscience*, 7(267), 1–13. <https://doi.org/10.3389/fnins.2013.00267>
- 116 Groppe, D. M., Bickel, S., Dykstra, A. R., Wang, X., Mégevand, P., Mercier, M. R., Lado,  
117 F. A., Mehta, A. D., & Honey, C. J. (2017). iELVis: An open source MATLAB toolbox  
118 for localizing and visualizing human intracranial electrode data. *Journal of Neuroscience*  
119 *Methods*, 281. <https://doi.org/10.1016/j.jneumeth.2017.01.022>
- 120 Hamilton, L. S., Chang, D. L., Lee, M. B., & Chang, E. F. (2017). Semi-automated anatom-  
121 ical labeling and inter-subject warping of high-density intracranial recording electrodes in  
122 electrocorticography. *Frontiers in Neuroinformatics*, 11. <https://doi.org/10.3389/fninf.2017.00062>
- 124 Oostenveld, R., Fries, P., Maris, E., & Schoffelen, J.-M. (2011). FieldTrip: Open Source Soft-  
125 ware for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data. *Com-  
126 putational Intelligence and Neuroscience*, 2011. <https://doi.org/10.1155/2011/156869>
- 127 Tadel, F., Baillet, S., Mosher, J. C., Pantazis, D., & Leahy, R. M. (2011). Brainstorm: A  
128 User-Friendly Application for MEG/EEG Analysis. *Computational Intelligence and Neuro-  
129 science*, 2011. <https://doi.org/10.1155/2011/879716>

DRAFT