

Kurzanleitung MNE-Python Pipeline

Installation:

https://www.martinos.org/mne/stable/install_mne_python.html

Zum Start in Anaconda Prompt „activate mne“ Tippen, dann ist man im richtigen Environment.

Oder über Anaconda Navigator mne auswählen.

Ich habe in die Pipeline autoreject eingebaut. Zu Beginn muss für diese Pipeline also auch noch dieses Package installiert werden:

1. Activate mne
2. Pip install autoreject

Lege einen **Skript-Ordner** an, in dem du sowohl die **Pipeline_<Dein Projekt>.py-Datei**, als auch den Ordner **Pipeline_Functions** ablegst. Das ist wichtig, damit die Pipeline_Functions gefunden werden können

Ich arbeite gerne mit **Spyder**, um die Pipeline zu bearbeiten und auszuführen. Dazu im Anaconda Prompt nach „activate mne“ „spyder“ eingeben.

Ordner einrichten (in Pipeline_Pinprick.py):

1. Bei **home_path** den Pfad für den Ordner auf eurem Computer/Festplatte mit allen Labor-Daten eingeben (OS ist egal, wenn du nur ein Betriebssystem nutzt)
2. Bei **project_name** den Pfad des Projektordners, mit dem ihr jetzt Daten analysieren wolltet eingeben
3. Bei **subjects_dir** den Pfad zum Ordner mit den schon von Freesurfer-segmentierten MRT-Sequenzen eingeben
join() kombiniert die durch Komma getrennten eingegebenen **Strings (eingerahmt von Anführungszeichen, wichtig!)** und fügt dazwischen „\\“ ein, also wie ein Pfad

Dateien einpflegen (Subject Organisation):

Funktionen mit „1“ statt „0“ anschalten

1. **Add_subjects:**
 - a. Hier Dateinamen (**ohne .fif**) eingeben und mit „**add**“ zu Liste hinzufügen (**sub_list.py** wird beim ersten Mal automatisch im Ordner Daten erstellt)
 - b. Mit **read**(Ausgabe in Konsole) und **delete_last** können Fehler verbessert werden
 - c. Am Ende **populate_dir** drücken um die Ordnerstrukturen für die Dateien zu erstellen
 - d. Jetzt sollten die Dateien manuell direkt in die den Dateinamen entsprechenden Ordner kopiert werden (der anat-Ordner würde erst relevant, wenn man die Pipeline auch für die Freesurfer-Segmentierung nutzen möchte)
 - e. In den Ordner **empty_room_data** können werden die Leerraummessungen kopiert. Dabei wäre es wichtig, dass diese Dateien vom Namen her den einzelnen Messtagen/Probanden zugeordnet werden können(**nicht immer derselbe Name für Leerraummessungen!**)

2. **Add_mri_subjects:**

- a. Hier die Namen der Segmentierungs-Ordner im Freesurfer-Ordner eingeben (also z.B. P200fs)

3. **Assign Subject to MRI-Subject:**

- a. Dateiname einer Messung auswählen
- b. Zugehörige Freesurfer-Segmentierung(Ordner-Name) eintippen
- c. **Assign** drücken(Dateiname muss in der Liste ausgewählt sein)

4. **Assign ERM to Subject:**

- a. Dateiname (**ohne .fif**) der Leeraummessung eingeben
- b. Zugehörige Messung auswählen (wahrscheinlich verwendet man für die Messungen eines Tages mehrmals dieselbe Leeraummessung=
- c. **Assign** drücken(Dateiname muss in der Liste ausgewählt sein)

5. **Assign bad channels to Subject:**

- a. Dateiname auswählen
- b. "plot_raw" drücken
- c. Schlechte Kanäle identifizieren(bei anklicken werden sie rot) und so eintragen:MEG 001,MEG 006,MEG 025,...
- d. Assign drücken(Dateiname muss in der Liste ausgewählt sein)
- e. Für diesen Schritt musst du schon die FIF-Dateien in die entsprechenden Ordner kopiert haben

6. **Coregistration:**

- a. Dafür müssen die MRT-Bilder von **Freesurfer** segmentiert im vorher als subjects_dir deklarierten Ordner liegen
- b. Außerdem müssen schon die **Bash-Funktionen**(„apply_watershed“ bis „make_morph_map“) durchgeführt worden sein (s. Funktionen auswählen)
- c. Tutorial hier: <https://www.slideshare.net/mne-python/mnepython-coregistration>

Dateien/Freesurfer-Ordner für Pipeline-Durchgang auswählen:

1. Als **STRING!!**(also mit Anführungszeichen) die Zeilennummern aus sub_list.py/mri_sub_list.py eingeben (am besten in einem separaten Fenster in Atom oder Spyder öffnen)
2. Kombinationen aus Abschnitten(x-y) oder Einzelauswahl(x,y,...) sind möglich
3. ‚all‘ wählt alle Dateien/Ordner aus

Funktionen auswählen:

1. 1 schaltet Funktion ein
2. 0 schaltet Funktion aus
3. **Die Source Space Operations greifen auf Bash-Funktionen zurück:**
 - a. Um die bash-Funktionen nutzen zu können, muss man MNE-Python und Freesurfer auf Linux installiert haben (ob es auf Mac funktioniert weiß ich nicht)
 - b. Ich habe auf Windows Ubuntu aus dem Windows Store installiert. Darin laufen sowohl Freesurfer als auch ein MNE-Python, mit dem ich die Bash-Funktionen dieser Pipeline nutze

Parameter einstellen:

1. Je nach Parameter Number oder String gefordert

Auf Run(F5) klicken und es läuft

(hoffentlich ohne Error)

Diese Pipeline wurde auf Grundlage von Lau Andersens Pipeline erstellt:

Andersen, L. M. (2018). Group analysis in MNE-python of evoked responses from a tactile stimulation paradigm: A pipeline for reproducibility at every step of processing, going from individual sensor space representations to an across-group source space representation. Frontiers in Neuroscience, 12(JAN). <https://doi.org/10.3389/fnins.2018.00006>

Die Pipeline ist immer noch im Entwicklungszustand.

Schreibt mir gerne, wenn ihr Fragen oder Ideen habt und euch Fehler auffallen.

martin.schulz@stud.uni-heidelberg.de