The MNE package for M/EEG data processing

A. Gramfort[†‡], M. Luessi[†], M. Hämäläinen[†], YOU ?
gramfort@nmr.mgh.harvard.edu
[†] MGH, Harvard Medical School, USA [‡] Parietal Project Team, INRIA, France

Features

Preprocessing

- Filter, review raw data, correct ECG / EOG with SSPs

Forward modeling

- Automatic BEM with Freesurfer reconstruction

Inverse modeling

- MNE - dSPM - sLORETA - LCMV - MxNE

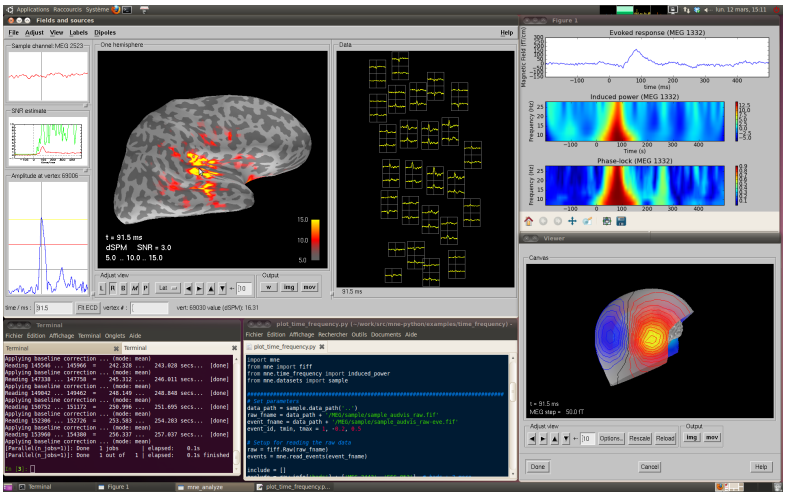
Statistics

- Parametric and non-parametric (sensor & source space)

Project vision & Goals

- State of the art pipeline
- Well documented, tested and easy to use
- Open development between different labs
- Sharing best practices for analysis
- Make reproducible research with M/EEG

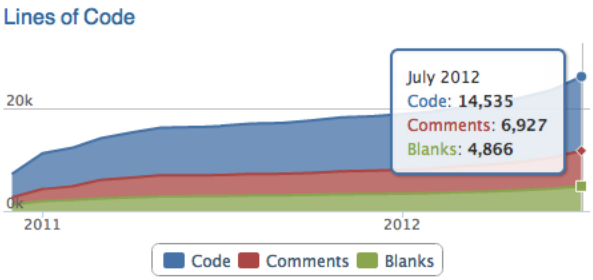
MNE-suite:
C / Unix & Matlab & Python



http://martinos.org/mne **http://github.com/mne-tools**

MNE-Python

- Python: general-purpose, high-level language
- Free (can run on a cluster without license problem)
- Permissive BSD license (allows use in commercial products)
- Open http://github.com/mne-tools/mne-python
- 14 contributors so far



From raw to dSPM in < 30 lines of code

```
fname = 'raw.fif'
raw = mne.fiff.Raw(fname)
raw.info['bads'] = ['MEG 2443', 'EEG 053']  # mark bad channels

# filter data in beta band
raw.filter(13.0, 30.0, filter_length=4096, n_jobs=8)

# save filtered raw data
raw.save(fname[:-4] + '_beta.fif')

# extract epochs
picks = mne.fiff.pick_types(raw.info, meg=True, eeg=True, eog=True,
                            exclude=raw.info['bads'])
events = mne.find_events(raw, stim_channel='STI 014')
epochs = mne.Epochs(raw, events, event_id=1, tmin=-0.2, tmax=0.5, proj=True,
                    picks=picks, baseline=(None, 0), preload=True,
                    reject=dict(grad=4000e-13, mag=4e-12, eog=150e-6))

# compute evoked response and noise covariance
evoked = epochs.average()
cov = mne.compute_covariance(epochs, tmax=0)

# Plot evoked
from mne.viz import plot_evoked
plot_evoked(evoked)

# Inverse modeling

# compute inverse operator
fwd_fname = 'sample_audvis-meg-eeg-oct-6-fwd.fif'
fwd = mne.read_forward_solution(fwd_fname, surf_ori=True)
inv = mne.minimum_norm.make_inverse_operator(raw.info, fwd, cov, loose=0.2)

# compute inverse solution
stc = mne.minimum_norm.apply_inverse(evoked, inv, lambda2=1 / 3.0 ** 2,
      method='dSPM')

# morph it to average brain for group study
stc_avg = mne.morph_data('sample', 'fsaverage', stc, 5, smooth=5)
```

Learn more

- Mailing list: mne_analysis@nmr.mgh.harvard.edu
- http://martinos.org/mne/ (general doc)
- http://martinos.org/mne/python_tutorial.html
- http://martinos.org/mne/auto_examples/ (> 40 demo)
- http://mne-tools.github.com/mne-python-intro-slides