MNE for MEG/EEG data processing: What's up?

A. Gramfort[1], M. Luessi[2], E. Larson[3], D.A. Engemann[4], D. Strohmeier[5]
C. Brodbeck[6], R. Goj[7], M. Jas[8], T. Brooks[6], L. Parkkonen[8], M. Hämäläinen[2]
[1]Telecom ParisTech, CNRS LTCI - CEA/Neurospin [2]MGH, Harvard Med. School, USA [3]U. of Washington, USA
[4]CNS (INM-3), Juelich Research Center, Germany [5]TU Ilmenau, Germany [7]Stirling, UK [8]AALTO, Helsinki [6]New York U., USA

The MNE Software Vision

- State-of-the-art methods, documented and tested
- Open development: collaboration between several centers
- Share the best practices, promote reproducible research

Software Features

Preprocessing

- Review raw data, filter, correct ECG / EOG with SSPs, ICA

Forward & inverse modeling

- FreeSurfer structural data: Automatic forward modeling
- MNE – dSPM – sLORETA – (TF-)MxNE – LCMV – DICS

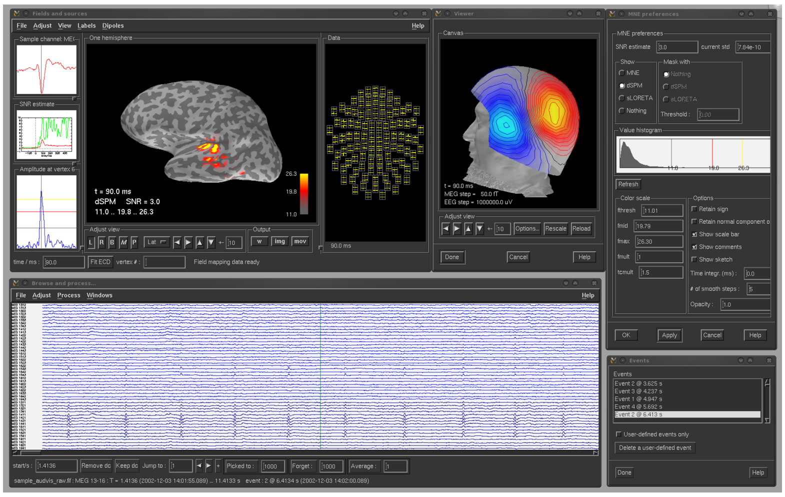
Statistics (sensor and source spaces)

- Time-Frequency (Phase-Locking, Induced Power)
- Parametric and non-parametric stats, with clustering
- Decoding/MVPA (SVM, ...)

Connectivity (sensor and source spaces)

- Functional and effective connectivity measures

The MNE Software Family
MNE-C – MNE-Matlab – MNE-Python



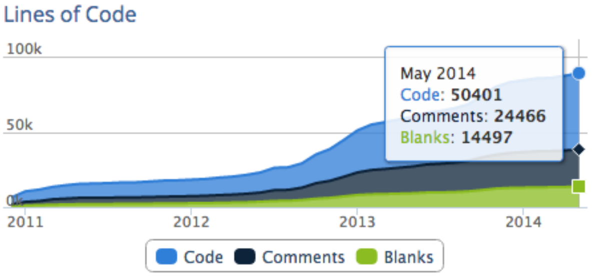
http://martinos.org/mne   **http://github.com/mne-tools**

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

A. Gramfort, M. Luessi, E. Larson, D. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, M. Hämäläinen, **MEG and EEG data analysis with MNE-Python**, Frontiers in Neuroscience, 2013

MNE-Python

- Python: general-purpose, high-level language
- Free: can run on a cluster without license problems
- Permissive BSD license: allows use in commercial products
- 46 contributors in last 12 months ≈ 13 person-years of effort

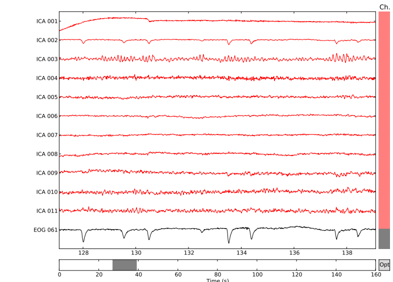


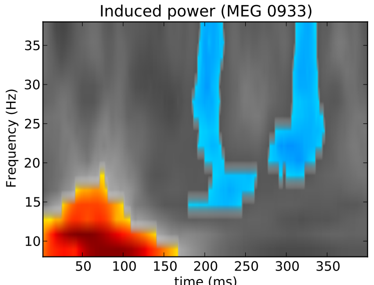
Learn more

- Mailing list: mne_analysis@nmr.mgh.harvard.edu
- http://martinos.org/mne/ (general doc)
- http://martinos.org/mne/dev/whats_new.html
- http://mne-tools.github.com/mne-python-intro-slides
- http://martinos.org/mne/auto_examples/ (> 70 demos)

From raw to dSPM in < 30 lines of code

```
import mne

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

# band-pass filter data in beta band, and save it
raw.filter(13.0, 30.0, filter_length=4096, n_jobs='cuda')
raw.save(fname[:-4] + '_beta.fif')

# extract epochs
picks = mne.pick_types(raw.info, meg=True, eeg=True, eog=True)
events = mne.find_events(raw)
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, and plot evoked
evoked = epochs.average()
cov = mne.compute_covariance(epochs, tmax=0)
evoked.plot()

# 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)
stc_avg.plot()
```


ICA 001
ICA 002
ICA 003
ICA 004
ICA 005
ICA 006
ICA 007
ICA 008
ICA 009
ICA 010
ICA 011
EOG 061
Ch.
Time (s)
Opt


Induced power (MEG 0933)
Frequency (Hz)
time (ms)