

SIGNAL PROCESSING IN MNE: DAY 3

1 – WORKING IN SOURCE SPACE FORWARD AND INVERSE MODELLING



09.02.2024

Thomas S. Binns

thomas-samuel.binns@charite.de

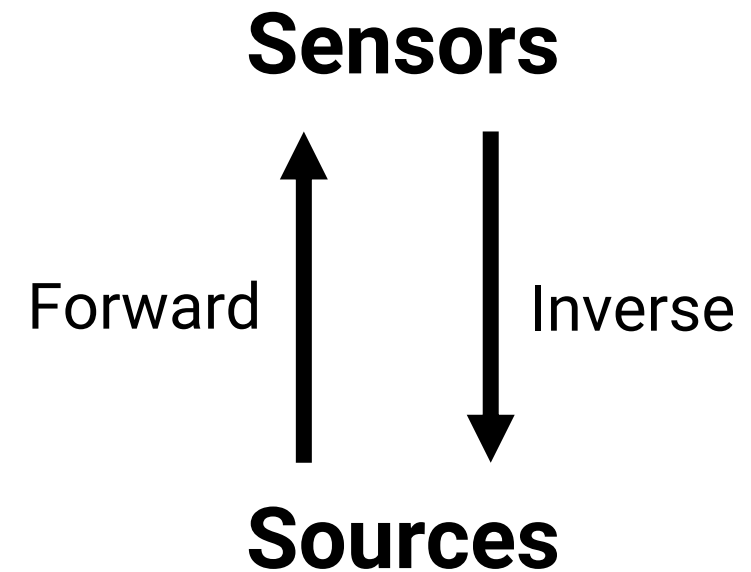
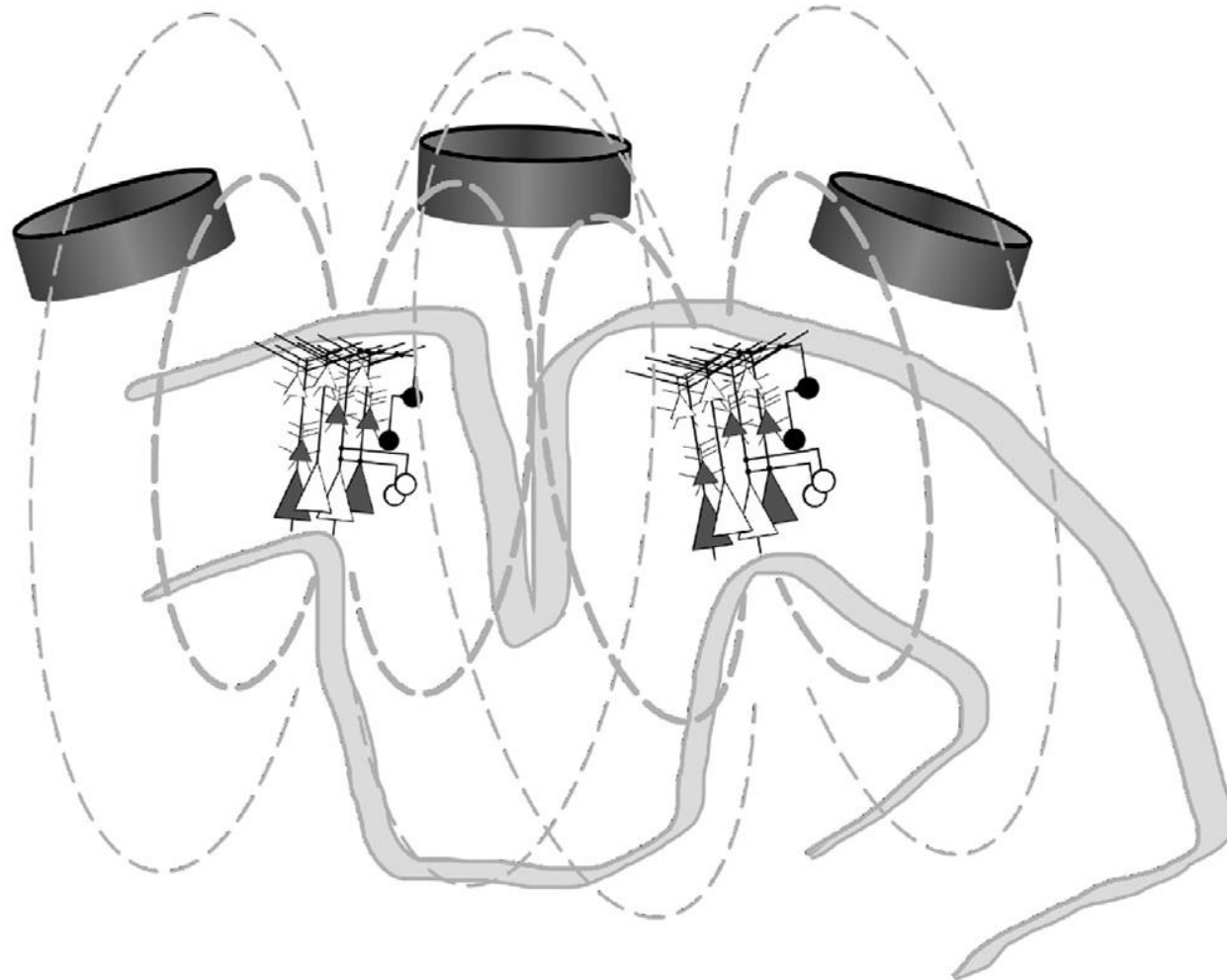
EINSTEIN
CENTER
Neurosciences



**Brain and
Data Science**
QAI LABS



Sensor vs. source space



Forward modelling




[Install](#) [Documentation](#) [API Reference](#) [Get help](#) [Development](#)

File I/O	✓
Creating data objects from arrays	✓
Exporting	✓
Datasets	✓
Visualization	✓
Preprocessing	✓
Events	✓
Sensor Space Data	✓
Covariance computation	✓
MRI Processing	✓
Forward Modeling	✓
Inverse Solutions	✓
Source Space Data	✓
Time-Frequency	✓
Connectivity Estimation	✓
Statistics	✓
Simulation	✓
Decoding	✓
Realtime	✓

Forward Modeling

<code>Forward</code>	Forward class to represent info from forward solution.
<code>SourceSpaces</code> (source_spaces[, info])	Represent a list of source space.
<code>add_source_space_distances</code> (src[, ...])	Compute inter-source distances along the cortical surface.
<code>apply_forward</code> (fwd, stc, info[, start, stop, ...])	Project source space currents to sensor space using a forward operator.
<code>apply_forward_raw</code> (fwd, stc, info[, start, ...])	Project source space currents to sensor space using a forward operator.
<code>average_forward_solutions</code> (fwds[, weights, ...])	Average forward solutions.
<code>convert_forward_solution</code> (fwd[, surf_ori, ...])	Convert forward solution between different source orientations.
<code>decimate_surface</code> (points, triangles, n_triangles)	Decimate surface data.
<code>dig_mri_distances</code> (info, trans, subject[, ...])	Compute distances between head shape points and the scalp surface.

Inverse modelling

MNE  [Install](#) [Documentation](#) [API Reference](#) [Get help](#) [Development](#)  

File I/O

Creating data objects from arrays

Exporting

Datasets

Visualization

Preprocessing

Events

Sensor Space Data

Covariance computation

MRI Processing

Forward Modeling

Inverse Solutions

Source Space Data

Time-Frequency

Connectivity Estimation

Statistics

Simulation

Decoding

Realtime

MNE-Report

Inverse Solutions

`mne.minimum_norm`:

Linear inverse solvers based on L2 Minimum Norm Estimates (MNE).

<code>InverseOperator</code>	InverseOperator class to represent info from inverse operator.
<code>apply_inverse</code> (evoked, inverse_operator[, ...])	Apply inverse operator to evoked data.
<code>apply_inverse_cov</code> (cov, info, inverse_operator)	Apply inverse operator to covariance data.
<code>apply_inverse_epochs</code> (epochs, ...[, method, ...])	Apply inverse operator to Epochs.
<code>apply_inverse_raw</code> (raw, inverse_operator, lambda2)	Apply inverse operator to Raw data.
<code>apply_inverse_tfr_epochs</code> (epochs_tfr, ...[, ...])	Apply inverse operator to EpochsTFR.
<code>compute_source_psd</code> (raw, inverse_operator[, ...])	Compute source power spectral density (PSD).
<code>compute_source_psd_epochs</code> (epochs, ...[, ...])	Compute source power spectral density (PSD) from Epochs.
<code>compute_rank_inverse</code> (inv)	Compute the rank of a linear inverse operator (MNE, dSPM, etc.).

Onto the notebook...

Conclusion

- Many tools for subject-specific forward and inverse modelling offered in MNE
- Variety of inverse modelling methods supported
- In-depth tutorials in MNE for using these tools