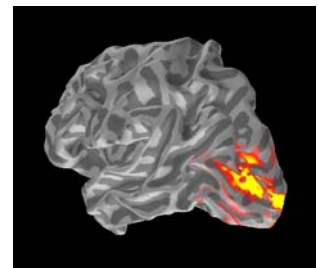
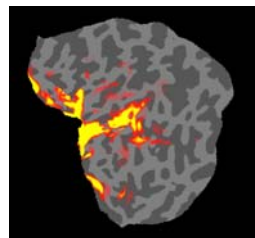
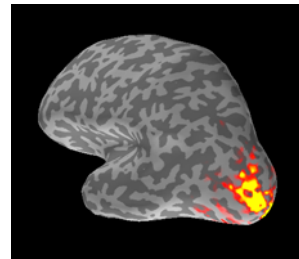
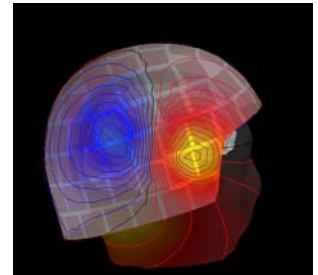


MNE software User's Guide

Version 2.5

December 2006



Matti Hämäläinen

MGH/HMS/MIT Athinoula A. Martinos Center for Biomedical Imaging
Massachusetts General Hospital
Charlestown, Massachusetts, USA

This document contains copyrighted information. The author reserves the right to make changes in the specifications or data shown herein at any time without notice or obligation. The author makes no warranty of any kind with regard to this document. The author shall not be liable for errors contained herein or direct, indirect, incidental or consequential damages in connection with the furnishing, performance, or use of this document.

Printing History	Identifier	Version	Date
1st edition	MSH-MNE	1.1	August 2001
2nd edition	MSH-MNE	1.2	April 2002
3rd edition	MSH-MNE	1.3	July 2002
4th edition	MSH-MNE	1.4	October 2002
5th edition	MSH-MNE	1.5	November 2002
6th edition	MSH-MNE	1.6	December 2002
7th edition	MSH-MNE	1.7	March 2003
8th edition	MSH-MNE	2.1	April 2005
9th edition	MSH-MNE	2.2	August 2005
10th edition	MSH-MNE	2.4	December 2005
11th edition	MSH-MNE	2.5	December 2006

Contents

Chapter 1	Introduction	9
Chapter 2	Software Overview	11
2.1	List of components	11
2.2	File formats	16
2.3	Conventions	16
2.4	Common options	16
2.5	User environment	17
Chapter 3	The Cookbook	19
3.1	Overview	19
3.2	Selecting the subject	20
3.3	Cortical surface reconstruction with FreeSurfer	20
3.4	Setting up the anatomical MR images for MRIlab	20
3.5	Setting up the source space	21
3.6	Creating the BEM model meshes	23
	Setting up the triangulation files	23
3.7	Setting up the boundary-element model	24
3.8	Setting up the MEG analysis directory	27
3.9	Preprocessing the raw data	27
	Cleaning the digital trigger channel	28
	Fixing channel information	28
	Designating bad channels	29
	Downsampling the MEG/EEG data	29
	Filtering the MEG/EEG data	30
	Off-line averaging	30
3.10	Aligning the coordinate frames	30
3.11	Computing the forward solution	31
3.12	Setting up the noise-covariance matrix	33
3.13	Calculating the inverse operator decomposition	34
3.14	Analyzing the data	37
Chapter 4	Processing raw data	39
4.1	Overview	39
4.2	Command-line options	39
	Common options	39
	Batch-mode options	41
4.3	The user interface	43
4.4	Description of menus	44
	The File menu	44

Open	44
Open evoked	45
Save	45
Change working directory	46
Read projection	46
Save projection	46
Apply bad channels	46
Save events (text)	46
Save channel selections	46
Load derivations	47
Quit	47
The Adjust menu	47
Filter	48
Scales	49
Selection	51
Compensation	53
Projection	54
Colors	54
Full view layout	54
Averaging preferences	54
Managing averages	55
The Process menu	56
Averaging	57
Computation of a covariance matrix	57
Computation of a covariance matrix from raw data	57
Estimation of noise subspace for SSP	57
The Windows menu	58
The Help menu	58
4.5 The raw data display	59
4.6 The toolbar	60
4.7 Topographical data displays	60
4.8 Description files for off-line averaging	61
Overall format	61
Common parameters	62
Category definition	63
4.9 Description files for covariance matrix computation	64
Overall format	64
Common parameters	65
Covariance definitions	66
4.10 Event files	67
4.11 The Signal-Space Projection (SSP) method	67
General concepts	67
Estimation of the noise subspace	69
EEG average electrode reference	70
4.12 Covariance matrix computation	70
Continuous raw data	70
Epochs	71
Combination of covariance matrix estimates	72
Projections included with the covariance matrix	72

Chapter 5 The forward solution	73
5.1 Overview	73
5.2 The head and device coordinate systems	73
5.3 Creating a source space	74
5.4 Creating the BEM meshes	76
Command-line options	76
Surface options	77
Tessellation file format	78
Topology checks	78
Examples	79
Importing tessellations from BrainSuite	79
5.5 Computing the BEM geometry data	80
5.6 Coil geometry information	81
The coil coordinate system	81
Calculation of the magnetic field	82
Implemented coil geometries	83
The coil definition file	87
Creating the coil definition file	89
5.7 Computing the forward solution	89
Purpose	89
Command line options	89
Implementation of software gradient compensation	91
The EEG sphere model definition file	92
EEG forward solution in the sphere model	92
Chapter 6 The current estimates	95
6.1 Overview	95
6.2 Minimum-norm estimates	95
The linear inverse operator	95
Regularization	96
Whitening and scaling	96
Regularization of the noise-covariance matrix	97
Computation of the solution	98
Noise normalization	99
Predicted data	99
Cortical patch statistics	100
The orientation constraints	100
Depth weighting	101
fMRI-biased estimates	101
6.3 Effective number of averages	101
6.4 Inverse-operator decomposition	103
6.5 Producing movies and snapshots	106
General options	107
Input files	107
Times and baseline	107
Options controlling the estimates	108
Visualization options	109
Thresholding	111

Output files	111
Label processing	112
Using stc file input	113
6.6 Computing inverse from raw data	114
Command-line options	114
Implementation details	116

Chapter 7 Interactive analysis	117
---------------------------------------	------------

7.1 Overview	117
7.2 Command line options	117
7.3 The main window	118
7.4 The menus	119
The File menu	119
The Adjust menu	120
The Windows menu	121
The Labels menu	121
The Help menu	122
7.5 Loading data	123
7.6 Loading epochs from a raw data file	125
7.7 Data displays	126
The topographical display	126
The sample channel display	127
Scale settings	127
7.8 The surface display	129
The surface selection dialog	130
The patch selection dialog	131
Controlling the surface display	131
Defining viewing orientations	133
Adjusting lighting	134
Producing output files	134
Image output modes	136
7.9 Morphing	137
7.10 The viewer	137
Overview	137
Viewer options	140
7.11 Magnetic field and electric potential maps	142
Overview	142
Mathematical description	142
Field mapping preferences	144
7.12 Working with current estimates	145
Preferences	145
The SNR display	148
7.13 Inquiring timecourses	149
Timecourses at vertices	149
Timecourses at labels	149
The timecourse manager	150
Label timecourse files	151
Creating new label files	152
7.14 Overlays	153

7.15	Importing current dipoles	155
7.16	Coordinate frame alignment	157
	Using a high-resolution head surface tessellation	159
7.17	Viewing continuous HPI data	160
Chapter 8 Morphing and averaging		163
8.1	Overview	163
8.2	The morphing maps	163
8.3	About smoothing	164
8.4	Precomputing the morphing maps	165
8.5	Morphing label data	166
8.6	Averaging	167
	Overview	167
	The averager	168
	The description file	168
Chapter 9 Data conversion		171
9.1	Overview	171
9.2	Importing data from other MEG/EEG systems	171
	Importing 4-D Neuroimaging data	171
	Importing CTF data	172
	Importing CTF Polhemus data	174
	Applying software gradient compensation	175
	Importing Magnes compensation channel data	177
	Creating software gradient compensation data	178
	Importing EEG data saved in the Tufts University format	179
	Importing KIT MEG system data	179
9.3	Listing source space data	182
9.4	Listing BEM mesh data	183
9.5	Converting an ncov covariance matrix file to fiff	184
9.6	Converting a lisp covariance matrix to fiff	185
9.7	The MNE data file conversion tool	185
	Command-line options	186
	Guide to combining options	188
	Matlab data structures	189
	Example matlab programs	192
9.8	Converting surface data to Matlab format	192
	command-line options	192
	Matlab data structure	193
	Conversion of both hemispheres	193
9.9	Converting raw data to Matlab format	194
	Command-line options	194
	Matlab data structures	195
9.10	Converting epochs to Matlab format	197
	Command-line options	197
	The binary epoch data file	199
	Matlab data structures	199

Chapter 10	The Matlab toolbox	203
10.1	Overview	203
10.2	Some data structures	209
10.3	On-line documentation for individual routines	222
Chapter 11	Miscellaneous utilities	223
11.1	Overview	223
11.2	Finding software versions	223
11.3	Listing contents of a fif file	223
11.4	Data file conditioning utilities	224
	Designating bad channels: mne_mark_bad_channels	224
	Fixing the encoding of the trigger channel: mne_fix_stim14	224
	Updating EEG location info: mne_check_eeg_locations	225
	Updating magnetometer coil types: mne_fix_mag_coil_types	225
	Modifying channel names and types: mne_rename_channels	226
	Modifying trigger channel data: mne_add_triggers	227
	Purpose	227
	Command line options	227
11.5	Creating a derivation file	228
	Purpose	228
	Command-line options	228
	Derivation file formats	229
11.6	Creating a custom EEG layout	230
	Purpose	230
11.7	Adding topology information to a source space	231
	Purpose	231
	Command line options	231
11.8	Averaging forward solutions	232
	Purpose	232
	Command line options	232
11.9	Converting covariance data into an SSP operator	232
	Purpose	232
	Command line options	233
11.10	Fitting a sphere to a surface	234
	Purpose	234
	Command line options	234
11.11	Applying signal-space projection to raw data	234
	Purpose	234
	Command line options	235
11.12	Computing sensitivity maps	235
	Purpose	235
	Command line options	235
	Available sensitivity maps	236
11.13	Transforming locations	237
	Purpose	237
	Command line options	237
11.14	Inquiring and changing baselines	238
11.15	Data simulator	239

	Purpose	239
	Command-line options	239
	Noise simulation	240
	Simulated data	240
11.16	Converting parcellation data into labels	241
Chapter 12	The sample data set	243
12.1	Purpose	243
12.2	Overview	243
12.3	Setting up	244
12.4	Contents of the data set	245
12.5	Setting up subject-specific data	246
	Structural MRIs	246
	Source space	246
	Boundary-element models	247
12.6	Setting up a custom EEG layout	247
12.7	Previewing the data	247
12.8	Off-line averaging	249
	Using the averaging script interactively	249
	Using the averaging script in batch mode	249
12.9	Viewing the off-line average	250
	Loading the averages	250
	Inspecting the auditory data	250
	Inspecting the visual data	251
12.10	Computing the noise-covariance matrix	251
12.11	MEG-MRI coordinate system alignment	252
	Using mne_analyze	252
	Initial alignment	252
	Refining the coordinate transformation	253
	Saving the transformation	254
	Using MRILab	255
12.12	The forward solution	255
12.13	The inverse operator decomposition	256
12.14	Interactive analysis	256
	Getting started	257
	Load surfaces	257
	Load the data	257
	Show field and potential maps	257
	Show current estimates	257
	Labels and timecourses	257
	Morphing	258
Chapter 13	Useful reading	259
13.1	General MEG reviews	259
13.2	Cortical surface reconstruction and morphing	259
13.3	Forward modelling	259
13.4	Signal-space projections	260
13.5	Minimum-norm estimates	260

13.6 fMRI-weighted estimates	261
Appendix A Creating the BEM meshes	263
A.1 Using the watershed algorithm	263
A.2 Using FLASH images	264
Organizing MRI data into directories	264
Creating the surface tessellations	265
Inspecting the meshes	266
A.3 Using seglab	266
A.4 Other options	267
Appendix B Setup at the Martinos Center	269
B.1 User environment	269
B.2 Using Neuromag software	269
Using Neuromag software on the HP workstations	269
Using Neuromag software on LINUX workstations	270
B.3 Mature software	271
mne_compute_mne	271
Using graph for signal processing	274
Filtering	274
Computing a noise-covariance matrix from raw data	274
Appendix C Installation and configuration	277
C.1 Installation	277
Create a directory to hold the software	277
Unpack the archive	277
C.2 Configuration	277
Set up a link to Matlab	277
Run the configuration script	278
Modify the login scripts	278
Add open source software	278
Optional: Remove unnecessary binaries	279
C.3 Licensing	279
C.4 Obtain FreeSurfer	280
C.5 How to get started	280
Appendix D Release notes	281
D.1 Release notes for MNE software 2.4	281
Manual	281
General software changes	281
File conversion utilities	281
mne_browse_raw	282
mne_analyze	282
mne_make_movie	282
Averaging	282
D.2 Release notes for MNE software 2.5	282

Manual	282
mne_browse_raw	283
mne_epochs2mat	283
mne_analyze	283
mne_ctf2fiff	284
mne_make_movie	284
mne_surf2bem	284
mne_forward_solution	284
mne_inverse_operator	285
mne_compute_raw_inverse	285
Time range settings	285
mne_change_baselines	285
New utilities	285
mne_show_fiff	285
mne_make_cor_set	285
mne_compensate_data	286
mne_insert_4D_comp	286
mne_ctf_dig2fiff	286
mne_kit2fiff	286
mne_make_derivations	286
BEM mesh generation	286
Matlab toolbox	286
 Appendix E Licence agreement	 289
E.1 License agreement	289

CHAPTER 1 Introduction

This document describes a set of programs for preprocessing and averaging of MEG and EEG data and for constructing cortically-constrained minimum-norm estimates. This suite is in the sequel referred to as *MNE software*. The software is based on anatomical MRI processing, forward modelling, and source estimation methods published in Dale, Fischl, Hämäläinen, and others. The software depends on anatomical MRI processing provided by the FreeSurfer software.

Chapter 2 of this manual gives an overview of the software modules included with MNE software. Chapter 3 is a concise cookbook describing a typical workflow for a novice user employing the convenience scripts as far as possible. Chapters 4 to 11 give more detailed information about the software modules. Chapter 12 discusses processing of the sample data set included with the MNE software. Chapter 13 lists some useful background material for the methods employed in the MNE software.

Appendix A is an overview of the BEM model mesh generation methods, Appendix B contains information specific to the setup at Martinos Center of Biomedical Imaging, Appendix C is a software installation and configuration guide, Appendix D summarizes the software history, and Appendix E contains the End-User License Agreement.

Note: The most recent version of this manual is available at `$MNE_ROOT/doc/MNE-manual-<version>.pdf`. For the present manual, `<version> = 2.5`. For definition of the `MNE_ROOT` environment variable, see Section 2.5.

I want to thank all MEG users at the Martinos Center for their collaboration during the creation of this software as well as for useful comments on the software and its documentation.

The development of this software has been supported by the NCRR grant P41RR14075-06, by Department of Energy under Award Number DE-FG02-99ER62764 to The MIND Institute, and NIH grants 1R01 HD40712-A1, 1R01 NS44319-01, and 2R01 NS37462-05

CHAPTER 2 Software Overview

2.1 List of components

The principal components of the MNE Software and their functions are listed in Table 2.1. Documented software is listed in italics. Table 2.2 lists various supplementary utilities.

Name	Purpose
<i>mne_analyze</i>	An interactive analysis tool for computing source estimates, see Chapter 7.
<i>mne_average_estimates</i>	Average data across subjects, see Section 8.6.2.
<i>mne_browse_raw</i>	Interactive raw data browser. Includes filtering, offline averaging, and computation of covariance matrices, see Chapter 4.
<i>mne_compute_mne</i>	Computes the minimum-norm estimates, see Section B.3.1. Most of the functionality of <i>mne_compute_mne</i> is included in <i>mne_make_movie</i> .
<i>mne_compute_raw_inverse</i>	Compute the inverse solution from raw data, see Section 6.6.
<i>mne_convert_mne_data</i>	Convert MNE data files to other file formats, see Section 9.7.
<i>mne_do_forward_solution</i>	Convenience script to calculate the forward solution matrix, see Section 3.11.
<i>mne_do_inverse_operator</i>	Convenience script to compute the inverse operator decomposition, see Section 3.13.
<i>mne_forward_solution</i>	Calculate the forward solution matrix, see Section 5.7.
<i>mne_inverse_operator</i>	Compute the inverse operator decomposition, see Section 6.4.
<i>mne_make_movie</i>	Make movies in batch mode, see Section 6.5.
<i>mne_make_source_space</i>	Create a <i>fif</i> source space description file, see Section 5.3.

Table 2.1 The software components.

Name	Purpose
<i>mne_process_raw</i>	A batch-mode version of <i>mne_browse_raw</i> , see Chapter 4.
<i>mne_redo_file</i>	Many intermediate result files contain a description of their ‘production environment’. Such files can be recreated easily with this utility. This is convenient if, for example, the selection of bad channels is changed and the inverse operator decomposition has to be recalculated.
<i>mne_redo_file_nocwd</i>	Works like <i>mne_redo_file</i> but does not try to change in to the working directory specified in the ‘production environment’
<i>mne_setup_forward_model</i>	Set up the BEM-related fif files, see Section 3.7.
<i>mne_setup_mri</i>	Convenience script to create the <i>fif</i> files describing the anatomical MRI data, see Section 3.4.
<i>mne_setup_source_space</i>	Convenience script to create a source space description file, see Section 3.5.
<i>mne_show_environment</i>	Show information about the production environment of a file.

Table 2.1 The software components.

Name	Purpose
<i>mne_add_patch_info</i>	Add patch information to a source space file, see Section 11.7.
<i>mne_add_to_meas_info</i>	Utility to add new information to the measurement info block of a fif file. The source of information is another fif file.
<i>mne_add_triggers</i>	Modify the trigger channel STI 014 in a raw data file, see Section 11.4.6. The same effect can be reached by using an event file for averaging in <i>mne_process_raw</i> and <i>mne_browse_raw</i> .
<i>mne_annot2labels</i>	Convert parcellation data into label files, see Section 11.16.
<i>mne_average_forward_solutions</i>	Calculate an average of forward solutions, see Section 11.8.
<i>mne_change_baselines</i>	Change the dc offsets according to specifications given in a text file, see Section 11.14.

Table 2.2 Utility programs.

Name	Purpose
<code>mne_change_nave</code>	Change the number of averages in an evoked-response data file. This is necessary if the file was derived from several files. For example, xplotter software cannot decide the equivalent number of averages when a weighted average of several data files is made.
<code>mne_check_eeg_locations</code>	Checks that the EEG electrode locations have been correctly transferred from the Polhemus data block to the channel information tags, see Section 11.4.3.
<code>mne_check_surface</code>	Check the validity of a FreeSurfer surface file. This program simply checks for topological errors in surface files.
<code>mne_compensate_data</code>	Change the applied software gradient compensation in an evoked-response data file, see Section 9.2.4.
<code>mne_convert_dfs</code>	Convert dfs triangulation files created by BrainSuite software to fif and FreeSurfer formats, see Section 5.4.6.
<code>mne_convert_lspcov</code>	Convert the LISP format noise covariance matrix output by graph into fif, see Section 9.6.
<code>mne_convert_ncov</code>	Convert the ncov format noise covariance file to fif, see Section 9.5.
<code>mne_convert_surface</code>	Convert FreeSurfer and text format surface files into Matlab mat files, see Section 9.8.
<code>mne_cov2proj</code>	Pick eigenvectors from a covariance matrix and create a signal-space projection (SSP) file out of them, see Section 11.10.
<code>mne_create_comp_data</code>	Create a fif file containing software gradient compensation information from a text file, see Section 9.2.6.
<code>mne_ctf2fiff</code>	Convert a CTF ds folder into a fif file, see Section 9.2.2.
<code>mne_ctf_dig2fiff</code>	Convert text format digitization data to fif format, see Section 9.2.3.
<code>mne_dicom_essentials</code>	List essential information from a DICOM file. This utility is used by the script <code>mne_organize_dicom</code> , see Section A.2.1.
<code>mne_dump_triggers</code>	List events (state changes) on the trigger channel STI 014. This functionality is now included with <code>mne_browse_raw</code> and <code>mne_process_raw</code> .
<code>mne_epochs2mat</code>	Apply bandpass filter to raw data and extract epochs for subsequent processing in Matlab, see Section 9.10.

Table 2.2 Utility programs.

Name	Purpose
<code>mne_evoked_data_summary</code>	List summary of averaged data from a fif file to the standard output.
<code>mne_fit_sphere_to_surf</code>	Fit a sphere to a surface given in either fif or FreeSurfer format, see Section 11.10.
<code>mne_fix_mag_coil_types</code>	Update the coil types for magnetometers in a fif file, see Section 11.4.4.
<code>mne_fix_stim14</code>	Fix coding errors of trigger channel STI 014, see Section 3.9.1.
<code>mne_flash_bem</code>	Create BEM tessellation using multi-echo FLASH MRI data, see Section A.2.
<code>mne_insert_4D_comp</code>	Read Magnes compensation channel data from a text file and merge it with raw data from other channels in a fif file, see Section 9.2.5
<code>mne_list_bem</code>	List BEM information in text format, see Section 9.4.
<code>mne_list_coil_def</code>	Create the coil description file. This is run automatically at when the software is set up, see Section 5.6.5.
<code>mne_list_proj</code>	List signal-space projection data from a fif file.
<code>mne_list_source_space</code>	List source space information in text format suitable for importing into Neuromag MRILab software, see Section 9.3.
<code>mne_list_surface_nodes</code>	List vertices from a FreeSurfer surface file.
<code>mne_list_versions</code>	List versions and compilation dates of MNE software modules, see Section 11.2.
<code>mne_make_cor_set</code>	Used by <code>mne_setup_mri</code> to create fif format MRI description files for images in COR format, see Section 3.4.
<code>mne_make_derivations</code>	Create a channel derivation data file, see Section 11.5.
<code>mne_make_eeg_layout</code>	Make a topographical trace layout file using the EEG electrode locations from an actual measurement, see Section 11.6.
<code>mne_make_morph_maps</code>	Precompute the mapping data needed for morphing between subjects, see Section 8.4.
<code>mne_make_uniform_stc</code>	Create a spatially uniform stc file for testing purposes.
<code>mne_mark_bad_channels</code>	Update the list of unusable channels in a data file, see Section 11.4.1.

Table 2.2 Utility programs.

Name	Purpose
<code>mne_mdip2stc</code>	Convert a spatiotemporal dipole description given by the Neuromag source modelling module <i>xfit</i> into an stc file.
<code>mne_morph_labels</code>	Morph label file definitions between subjects, see Section 8.5.
<code>mne_organize_dicom</code>	Organized DICOM MRI image files into directories, see Section A.2.1.
<code>mne_prepare_bem_model</code>	Perform the geometry calculations for BEM forward solutions, see Section 5.5.
<code>mne_process_stc</code>	Manipulate stc files.
<code>mne_project_raw</code>	Apply a signal-space projection to a raw data file, see Section 11.11.
<code>mne_raw2mat</code>	Convert raw data into a Matlab file, see Section 9.9.
<code>mne_rename_channels</code>	Change the names and types of channels in a fif file, see Section 11.4.5.
<code>mne_sensitivity_map</code>	Compute a sensitivity map and output the result in a w-file, see Section 11.12.
<code>mne_sensor_locations</code>	Create a file containing the sensor locations in text format.
<code>mne_show_fiff</code>	List contents of a fif file, see Section 11.3
<code>mne_simu</code>	Simulate MEG and EEG data, see Section 11.15.
<code>mne_smooth_w</code>	Smooth a w file.
<code>mne_surf2bem</code>	Create a <i>fif</i> file describing the triangulated compartment boundaries for the boundary-element model (BEM), see Section 5.4.
<code>mne_surface2mat</code>	Create Matlab files from FreeSurfer surface files, see Section 9.8.3.
<code>mne_toggle_skips</code>	Change data skip tags in a raw file into ignored skips or vice versa.
<code>mne_transform_points</code>	Transform between MRI and MEG head coordinate frames, see Section 11.13.
<code>mne_tufts2fiff</code>	Convert EEG data from the Tufts University format to fif format, see Section 9.2.7.
<code>mne_view_manual</code>	Starts a PDF reader to show this manual from its standard location.
<code>mne_volume_source_space</code>	Make a source space containing the brain volume.

Table 2.2 Utility programs.

Name	Purpose
<i>mne_watershed_bem</i>	Do the segmentation for BEM using the watershed algorithm, see Section A.1.

Table 2.2 Utility programs.

2.2 File formats

The software suite employs the *fif* file format whenever possible. New tags have been added to incorporate MNE-specific information. FreeSurfer file formats are also employed when needed to represent cortical surface geometry data as well as spatiotemporal distribution of quantities on the surfaces. Of particular interest are the *w* files, which contain static overlay data on the cortical surface and *stc* files, which contain dynamic overlays (movies).

2.3 Conventions

When command line examples are shown, the backslash character (\) indicates a continuation line. It is also valid in the shells. In most cases, however, you can easily fit the commands listed in this manual on one line. The order of options for the commands is irrelevant. Entries to be typed literally are shown like `this`. *Italicized* text indicates conceptual entries. For example, *<dir>* indicates that a directory name should be written.

In the description of interactive software modules the notation *<menu>/<item>* is often used to denote menu selections. For example, *File/Quit* stands for the *Quit* button in the *File* menu.

All software modules employ the GNU-style double-dash (--) option convention, *i.e.*, the options to a command line should be preceded by two dashes.

2.4 Common options

Most of the program modules have two common options to obtain information about the software:

--help

Prints concise usage information.

--version

Prints the program module name, version number, and compilation date.

2.5 User environment

The MNE Software location is system dependent. Consult your system administrators to find the location where MNE software is installed. This location will be here referred to with the symbol $\langle MEG_{sw} \rangle$. There are two scripts for setting up user environment so that the software can be used conveniently:

```
 $\langle MEG_{sw} \rangle$ /mne/setup/mne/mne_setup_analysis_sh
```

and

```
 $\langle MEG_{sw} \rangle$ /mne/setup/mne/mne_setup_analysis_csh
```

compatible with the POSIX and csh/tcsh shells, respectively. Since the scripts set environment variables they should be ‘sourced’ to the present using the `.` (POSIX shell) and `source` (csh/tcsh) commands. Among other things these scripts set the environment variable `$MNE_ROOT` to point to $\langle MEG_{sw} \rangle$ /mne.

For creating source spaces using the traditional method with specified approximate grid spacing (see Sections 3.5 and Section 5.3) and BEM mesh generation using the watershed algorithm or on the basis of multi-echo FLASH MRI data (see Appendix A), the MNE software needs access to a *FreeSurfer* license and software. Therefore, to use these features it is mandatory that you set up the *FreeSurfer* environment as described in the *FreeSurfer* documentation.

Tip: A convenient way to provide a shortcut for MNE analysis setup is to create an alias called *mne_setup* in the shell startup file. For example, if you are using tcsh, add the following line to your `.tcshrc` file, located in your home directory:

```
alias mne_setup 'source\  
 $\langle MEG_{sw} \rangle$ /mne/setup/mne/mne_setup_analysis_csh'
```

Note: Appendix B contains information specific to the setup at the Martinos Center including instructions to access the Neuromag software.

CHAPTER 3 The Cookbook

3.1 Overview

This section describes the typical workflow needed to produce the minimum-norm estimate movies using the MNE software. The workflow is summarized in Figure 3.1.

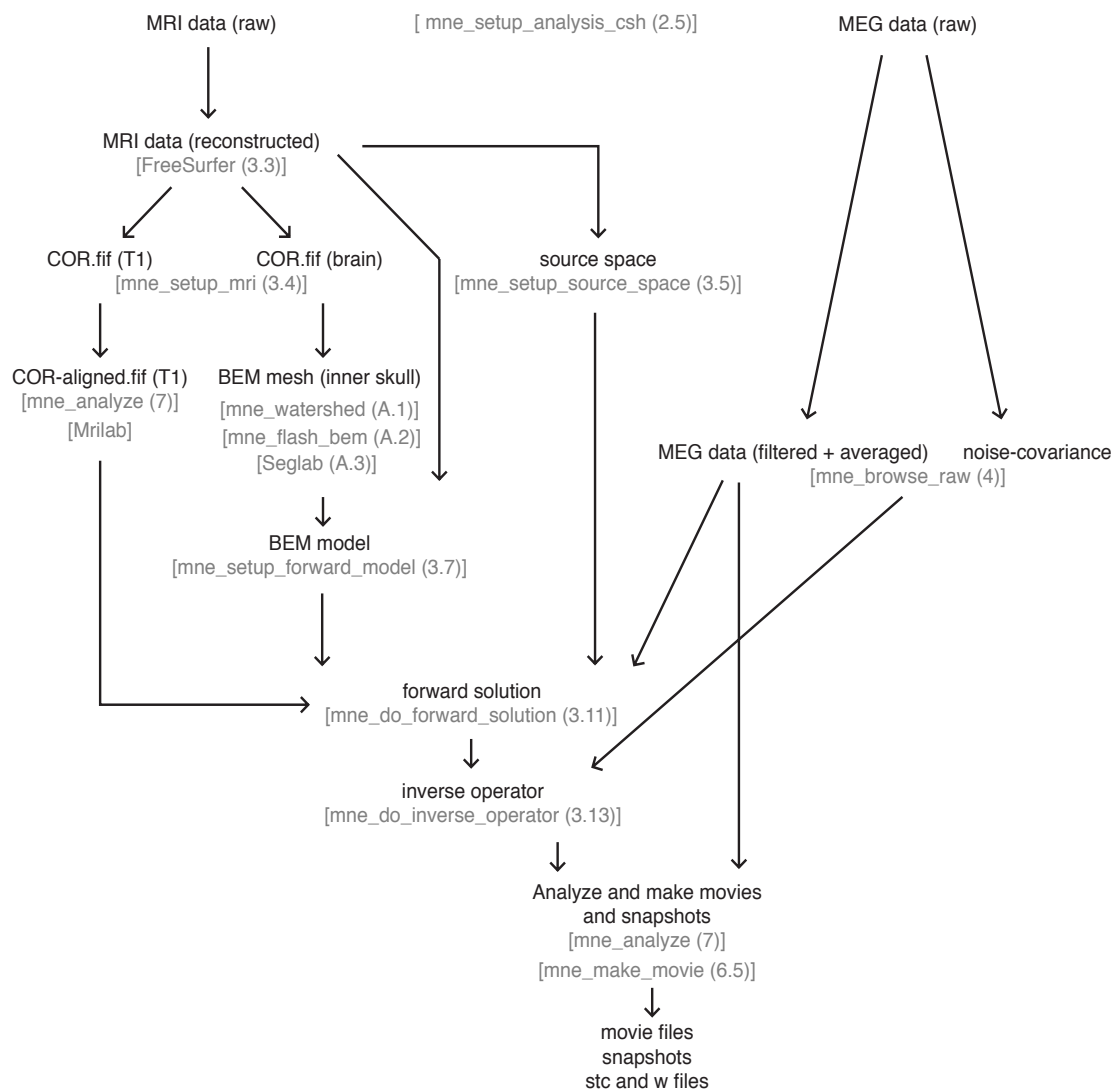


Figure 3.1 Workflow of the MNE software. References in parenthesis indicate sections and chapters of this manual.

3.2 Selecting the subject

Before starting the MNE analysis, setup the environment variable `SUBJECTS_DIR` to select the directory under which the anatomical MRI data are stored. Optionally, set `SUBJECT` as the name of the subject's MRI data directory under `SUBJECTS_DIR`. With this setting you can avoid entering the `--subject` option. In the following sections, files in the FreeSurfer directory hierarchy are usually referred to without specifying the leading directories. Thus, `bem/msh-7-src.fif` is used to refer to the file `$SUBJECTS_DIR/$SUBJECT/bem/msh-7-src.fif`.

3.3 Cortical surface reconstruction with FreeSurfer

The first processing stage is the creation of various surface reconstructions with *FreeSurfer*. Please refer to the *FreeSurfer* documentation for more information.

3.4 Setting up the anatomical MR images for MRILab

The Neuromag software accesses the MRIs with help of a fif-format description file. During various stages of the MNE calculations it is useful to have the MRIs accessible to the Neuromag MRI viewer, MRILab. This is accomplished by running the script `mne_setup_mri`, which requires that the subject is set with the `--subject` option or by the `SUBJECT` environment variable. The script processes one or more MRI data sets from `$SUBJECTS_DIR/$SUBJECT/mri`, by default they are `T1` and `brain`. This default can be changed by specifying the sets by one or more `--mri` options.

The script creates the directories `mri/<name>/slices` and `mri/<name>/sets`, makes symbolic links from the COR files in the directory `mri/<name>` into `mri/<name>-neuromag/slices`, and creates the corresponding fif file `COR.fif` in `mri/<name>-neuromag/sets`. For example:

```
mne_setup_mri --subject duck_donald --mri T1
```

This command processes the MRI data set `T1` for subject `duck_donald`.

Tip: If the `SUBJECT` environment variable is set it is usually sufficient to run `mne_setup_mri` without any options.

Important: If the surface reconstructions have been made with a recent version of *FreeSurfer*, the MRI data may be stored in *mgz* format. `mne_setup_mri` automatically converts these files into COR files using `mri_convert` and stores them in the `<name>-neuromag/slices`

directories. This step requires that the *FreeSurfer* user environment has been set up.

3.5 Setting up the source space

This stage consists of the following:

1. Creating a suitable decimated dipole grid on the white matter surface.
2. Creating the source space file in fif format.
3. Creating ascii versions of the source space file for viewing with MRILab.

All of the above is accomplished with the script convenience `mne_setup_source_space`. This script assumes that:

1. The anatomical MRI processing has been completed as described in Section 3.3.
2. The environment variable `SUBJECTS_DIR` is set correctly.

The script accepts the following options:

--subject <subject>

Defines the name of the subject. If the environment variable `SUBJECT` is set correctly, this option is not required.

--spacing <spacing/mm>

Specifies the grid spacing for the source space. If not set, a default spacing of 7 mm is used. Either the default or a 5-mm spacing is recommended. **Note:** This option is not available on the Mac OSX platform.

--ico <number>

Instead of using the traditional method for cortical surface decimation it is possible to create the source space using the topology of a recursively subdivided icosahedron (`<number> > 0`) or an octahedron (`<number> < 0`). This method uses the cortical surface inflated to a sphere as a tool to find the appropriate vertices for the source space. The benefit of the `--ico` option is that the source space will have triangulation information for the decimated vertices included, which future versions of MNE software may be able to utilize. The number of triangles increases by a factor of four in each subdivision, starting from 20 triangles in an icosahedron and 8 triangles in an octahedron. Since the number of vertices on a closed surface is $n_{\text{vert}} = (n_{\text{tri}} + 4)/2$, the number of vertices in the k th subdivision of an icosahedron and an octahedron are $10 \cdot 4^k + 2$ and $4^{k+1} + 2$, respectively. The recommended values for `<number>` and the corresponding number of source space locations are listed in Table 3.1.

--surface <name>

Name of the surface under the surf directory to be used. Defaults to 'white'. `mne_setup_source_space` looks for files `rh.<name>` and `lh.<name>` under the surf directory.

--overwrite

An existing source space file with the same name is overwritten only if this option is specified.

--cps

Compute the cortical patch statistics. This is need if current-density estimates are computed, see Section 6.2.8. If the patch information is available in the source space file the surface normal is considered to be the average normal calculated over the patch instead of the normal at each source space location. The calculation of this information takes a considerable amount of time because of the large number of Dijkstra searches involved.

<number>	Sources per hemisphere	Source spacing / mm	Surface area per source / mm²
-5	1026	9.9	97
4	2562	6.2	39
-6	4098	4.9	24
5	10242	3.1	9.8

Table 3.1 Recommended subdivisions of an icosahedron and an octahedron for the creation of source spaces. The approximate source spacing and corresponding surface area have been calculated assuming a 1000-cm² surface area per hemisphere.

For example, to create the reconstruction geometry for Donald Duck with a 5-mm spacing between the grid points, say

```
mne_setup_source_space --subject duck_donald \
--spacing 5
```

As a result, the following files are created into the `bem` directory:

1. `<subject>-<spacing>-src.fif` containing the source space description in fif format.
2. `<subject>-<spacing>-lh.pnt` and `<subject>-<spacing>-rh.pnt` containing the source space points in MNIlab compatible ascii format.
3. `<subject>-<spacing>-lh.dip` and `<subject>-<spacing>-rh.dip` containing the source space points in MNIlab compatible ascii format.

These files contain ‘dipoles’, *i.e.*, both source space points and cortex normal directions.

4. If cortical patch statistics is requested, another source space file called `<subject>-<spacing>p-src.fif` will be created.

Note: `<spacing>` will be the suggested source spacing in millimeters if the `--spacing` option is used. For source spaces based on *k*th subdivision of an icosahedron, `<spacing>` will be replaced by `ico-k` or `oct-k`, respectively.

After the geometry is set up it is advisable to check that the source space points are located on the cortical surface. This can be easily done with by loading the `COR.fif` file from `mri/T1/neuromag/sets` into MRILab and by subsequently overlaying the corresponding `pnt` or `dip` files using *Import/Strings* or *Import/Dipoles* from the *File* menu, respectively.

Tip: If the `SUBJECT` environment variable is set correctly it is usually sufficient to run `mne_setup_source_space` without any options.

3.6 Creating the BEM model meshes

Calculation of the forward solution using the boundary-element model (BEM) requires that the surfaces separating regions of different electrical conductivities are tessellated with suitable surface elements. Our BEM software employs triangular tessellations. Therefore, prerequisites for BEM calculations are the segmentation of the MRI data and the triangulation of the relevant surfaces.

For MEG computations, reasonably accurate solution can be obtained by using a single-compartment BEM assuming the shape of the intracranial volume. For EEG, the standard model contains the intracranial space, the skull, and the scalp.

At present, no bulletproof method exists for creating the triangulations. Feasible approaches are described in Appendix A.

3.6.1 Setting up the triangulation files

The segmentation algorithms described in Appendix A produce either FreeSurfer surfaces or text triangulation data. Before proceeding to the creation of the boundary element model, standard files (or symbolic links) have to be present in the subject’s `bem` directory. If you are employing ASCII triangle files the standard file names are:

inner_skull.tri

Contains the inner skull triangulation.

outer_skull.tri

Contains the outer skull triangulation.

outer_skin.tri

Contains the head surface triangulation.

The corresponding names for FreeSurfer surfaces are:

inner_skull.surf

Contains the inner skull triangulation.

outer_skull.surf

Contains the outer skull triangulation.

outer_skin.surf

Contains the head surface triangulation.

Tip: As long as all BEM surfaces are in the same format, different methods can be employed for the creation of the individual surfaces. For example, it may turn out that the watershed algorithm produces a better quality skin surface than the segmentation approach based on the FLASH images. If this is the case, `outer_skin.surf` can be set to point to the corresponding watershed output file while the other surfaces can be picked from the FLASH segmentation data.

3.7 Setting up the boundary-element model

This stage accomplishes the following operations:

1. Creating the boundary element model geometry file in `fif` format. This step also checks that the input surfaces are complete and that they are topologically correct, *i.e.*, that the surfaces don't intersect. Furthermore, the range of triangle sizes on each surface is reported. For the three-layer model, the minimum distance between the surfaces is also computed.
2. Creating `ascii` versions of the boundary surface vertex coordinates for viewing with `MRILab`.
3. Calculating the geometry-dependent BEM solution data. This step can be optionally omitted. This step takes several minutes to complete.

This processing stage is automated with the script `mne_setup_forward_model`. This script assumes that:

1. The anatomical MRI processing has been completed as described in Section 3.3.
2. The BEM model meshes have been created as outlined in Section 3.6.
3. The environment variable `SUBJECTS_DIR` is set correctly.

`mne_setup_forward_model` accepts the following options:

--subject <subject>

Defines the name of the subject. This can be also accomplished by setting the `SUBJECT` environment variable.

--surf

Use the FreeSurfer surface files instead of the default ASCII triangulation files. Please consult Section 3.6.1 for the standard file naming scheme.

--noswap

Traditionally, the vertices of the triangles in ‘tri’ files have been ordered so that, seen from the outside of the triangulation, the vertices are ordered in clockwise fashion. The `fif` files, however, employ the more standard convention with the vertices ordered counterclockwise. Therefore, `mne_setup_forward_model` by default reverses the vertex ordering before writing the `fif` file. If, for some reason, you have counterclockwise-ordered `tri` files available this behavior can be turned off by defining `--noswap`. When the `fif` file is created, the vertex ordering is checked and the process is aborted if it is incorrect after taking into account the state of the swapping. Should this happen, try to run `mne_setup_forward_model` again including the `--noswap` flag. In particular, if you employ the `seglab` software to create the triangulations (see Appendix A), the `--noswap` flag is required. This option is ignored if `--surf` is specified

--ico <number>

This option is relevant (and required) only with the `--surf` option and if the surface files have been produced by the watershed algorithm. The watershed triangulations are isomorphic with an icosahedron, which has been recursively subdivided six times to yield 20480 triangles. However, this number of triangles is too large for present computers. Therefore, the triangulations have to be decimated. Specifying `--ico 4` yields 5120 triangles per surface while `--ico 3` results in 1280 triangles. The recommended choice is `--ico 4`.

--homog

Use a single compartment model (brain only) instead a three layer one (scalp, skull, and brain). Only the `inner_skull.tri` triangulation is required. This model is usually sufficient for MEG but invalid for EEG. If you are employing MEG data only, this option is recommended because of faster computation times. If this flag is specified, the options `--brainc`, `--skullc`, and `--scalpc` are irrelevant.

--brainc <conductivity/ S/m>

Defines the brain compartment conductivity. The default value is 0.3 S/m.

--skullc <conductivity/ S/m>

Defines the skull compartment conductivity. The default value is 0.06 S/m corresponding to a conductivity ratio 1/50 between the brain and skull compartments.

--scalpc <conductivity/ S/m>

Defines the brain compartment conductivity. The default value is 0.3 S/m.

--innershift <value/mm>

Shift the inner skull surface outwards along the vertex normal directions by this amount.

--nosol

Omit the BEM model geometry dependent data preparation step. This can be done later by running `mne_setup_forward_model` without the `--nosol` option.

--model <name>

Name for the BEM model geometry file. The model will be created into the directory `bem` as `<name>-bem.fif`. If this option is missing, standard model names will be used (see below).

For example, to create the BEM model for Donald Duck with standard vertex ordering and conductivities, say

```
mne_setup_forward_model --subject duck_donald
```

As a result, the following files are created into the `bem` directory:

1. BEM model geometry specifications `<subject>-<ntri-scalp>-<ntri-outer_skull>-<ntri-inner_skull>-bem.fif` or `<subject>-<ntri-inner_skull>-bem.fif` containing the BEM geometry in `fif` format. The latter file is created if `-homog` option is specified. Here, `<ntri-xxx>` indicates the number of triangles on the corresponding surface.
2. `<subject>-<surface name>-bem.pnt` files are created for each of the surfaces present in the BEM model. These can be loaded to MRILab to check the location of the surfaces.
3. The BEM 'solution' file containing the geometry dependent solution data will be produced with the same name as the BEM geometry specifications with the ending `-bem-sol.fif`. These files also contain all the information in the `-bem.fif` files.

After the BEM is set up it is advisable to check that the BEM model meshes are correctly positioned. This can be easily done with by loading the `COR.fif` file from `mri/T1-neuromag/sets` into MRILab and by subse-

quently overlaying the corresponding pnt files using *Import/Strings* from the *File* menu.

Tip: The *FreeSurfer* format BEM surfaces can be also viewed with the *tkmedit* program which is part of the *FreeSurfer* distribution.

Tip: If the `SUBJECT` environment variable is set, it is usually sufficient to run `mne_setup_forward_model` without any options for the three-layer model and with the `--homog` option for the single-layer model. If the input files are *FreeSurfer* surfaces, `--surf` and `--ico` 4 are required as well.

Tip: With help of the `--nosol` option it is possible to create candidate BEM geometry data files quickly and do the checking with respect to the anatomical MRI data. When the result is satisfactory, `mne_setup_forward_model` can be run without `--nosol` to invoke the time-consuming calculation of the solution file as well.

Note: The triangle meshes created by the *seglab* program have counter-clockwise vertex ordering and thus require the `--noswap` option.

Note: Up to this point all processing stages depend on the anatomical (geometrical) information only and thus remain identical across different MEG studies.

3.8 Setting up the MEG analysis directory

The remaining steps require that the actual MEG data are available. It is recommended that a new directory is created for the MEG data processing. The raw data files collected shouldn't be copied there but rather referred to with symbolic links. Averages calculated on-line can be either copied or referred to with links.

Tip: If you don't know how to create a directory, how to make symbolic links, or how to copy files from the shell command line, this is a perfect time to learn about this basic skills from other users or from a suitable elementary book before proceeding.

3.9 Preprocessing the raw data

The following MEG and EEG data preprocessing steps are recommended:

1. The coding problems on the trigger channel STI 014 may have to fixed, see Section 3.9.1.
2. EEG electrode location information and MEG coil types may need to be fixed, see Section 3.9.2.

3. The data may be optionally downsampled to facilitate subsequent processing, see Section 3.9.4.
4. Bad channels in the MEG and EEG data must be identified, see Section 3.9.3
5. The data has to be filtered to the desired passband, see Section 3.9.5. If continuous data are processed with the Neuromag signal processor, graph, this step is usually accomplished on the fly as data are being processed. If *mne_browse_raw* or *mne_process_raw* is employed to calculate the offline averages and covariance matrices, this step is unnecessary. For information on these programs, please consult Chapter 4.
6. For evoked-response analysis, the data has to be reaveraged off line, see Section 3.9.6.

3.9.1 Cleaning the digital trigger channel

The calibration factor of the digital trigger channel used to be set to a value much smaller than one by the Neuromag data acquisition software. Especially to facilitate viewing of raw data in graph it is advisable to change the calibration factor to one. Furthermore, the eighth bit of the trigger word is coded incorrectly in the original raw files. Both problems can be corrected by saying:

```
mne_fix_stim14 <raw file>
```

More information about *mne_fix_stim14* is available in Section 11.4.2. It is recommended that this fix is included as the first raw data processing step. Note, however, the *mne_browse_raw* and *mne_process_raw* always sets the calibration factor to one internally.

Note: If your data file was acquired on or after November 10, 2005 on the Martinos center Vectorview system, it is not necessary to use *mne_fix_stim14*.

3.9.2 Fixing channel information

There are two potential discrepancies in the channel information which need to be fixed before proceeding:

1. EEG electrode locations may be incorrect if more than 60 EEG channels are acquired.
2. The magnetometer coil identifiers are not always correct.

These potential problems can be fixed with the utilities *mne_check_eeg_locations* and *mne_fix_mag_coil_types*, see Sections 11.4.3 and 11.4.4.

3.9.3 Designating bad channels

Sometimes some MEG or EEG channels are not functioning properly for various reasons. These channels should be excluded from the analysis by marking them bad using the *mne_mark_bad_channels* utility, see Section 11.4.1. Especially if a channel is not show a signal at all (flat) it is most important to exclude it from the analysis, since its noise estimate will be unrealistically low and thus the current estimate calculations will give a strong weight to the zero signal on the flat channels and will essentially vanish. It is also important to exclude noisy channels because they can possibly affect others when signal-space projections or EEG average electrode reference is employed. Noisy bad channels can also adversely affect off-line averaging and noise-covariance matrix estimation by causing unnecessary rejections of epochs.

Recommended ways to identify bad channels are:

1. Observe the quality of data during data acquisition and make notes of observed malfunctioning channels to your measurement protocol sheet.
2. View the on-line averages and check the condition of the channels.
3. Compute preliminary off-line averages with artefact rejection, signal-space projection, and EEG average electrode reference computation off and check the condition of the channels.
4. View raw data in *mne_process_raw* or the Neuromag signal processor *graph* without signal-space projection or EEG average electrode reference computation and identify bad channels.

Important: It is strongly recommended that bad channels are identified and marked in the original raw data files. If present in the raw data files, the bad channel selections will be automatically transferred to averaged files, noise-covariance matrices, forward solution files, and inverse operator decompositions.

3.9.4 Downsampling the MEG/EEG data

The minimum practical sampling frequency of the Vectorview system is 600 Hz. Lower sampling frequencies are allowed but result in elevated noise level in the data. Especially in cognitive studies and for long raw data files with epileptic data it may be advisable to lowpass filter down-sample the data off line to speed up subsequent processing. This can be accomplished with the *mne_process_raw* and *mne_browse_raw* software modules. For details, see Sections 4.2.2 and 4.4.1.3.

Tip: It is recommended that the original raw file is called *<name>_raw.fif* and the downsampled version *<name>_ds_raw.fif*, respectively.

Note: The Neuromag downsampling utility *fiff_downsample* is not included with the MNE software any more.

3.9.5 Filtering the MEG/EEG data

Before averaging and computation of the noise-covariance matrix the raw data should be filtered to the final passband. Flexible filtering options are provided by the continuous data processor, `graph`, part of the Neuromag software suite. This program is presently available on the HP-UX workstations only and only for owners of the Neuromag analysis software. Filtering is also included in `mne_browse_raw` and `mne_process_raw`, see Chapter 4.

3.9.6 Off-line averaging

The recommended tools for off-line averaging are `mne_browse_raw` and `mne_process_raw`. `mne_browse_raw` is an interactive program for averaging and noise-covariance matrix computation. It also includes routines for filtering so that the downsampling and filtering steps can be skipped. Therefore, with `mne_browse_raw` you can produce the off-line average and noise-covariance matrix estimates directly. The batch-mode version of `mne_browse_raw` is called `mne_process_raw`. Detailed information on `mne_browse_raw` and `mne_process_raw` can be found in Chapter 4.

Note: Some MNE software users may have access to the averaging software `meg_average` written by Thomas Witzel. Please consult the author of `meg_average` for instructions.

3.10 Aligning the coordinate frames

The calculation of the forward solution requires knowledge of the relative location and orientation of the MEG and MRI coordinate systems. The MEG head coordinate system is defined in Section 5.2. The conversion tools included in the MNE software take care of the idiosyncracies of the coordinate frame definitions in different MEG and EEG systems so that the `fif` files always employ the same definition of the head coordinate system.

Ideally, the head coordinate frame has a fixed orientation and origin with respect to the head anatomy. Therefore, a single MRI-head coordinate transformation for each subject should be sufficient. However, as explained in Section 5.2, the head coordinate frame is defined by identifying the fiducial landmark locations, making the origin and orientation of the head coordinate system slightly user dependent. As a result, the most conservative choice for the definition of the coordinate transformation computation is to re-establish it for each experimental session, i.e., each time when new head digitization data are employed.

The interactive source analysis software *mne_analyze* provides tools for coordinate frame alignment, see Chapter 7. Section 12.11 also contains tips for using *mne_analyze* for this purpose.

Another useful tool for the coordinate system alignment is *MRIlab*, the Neuromag MEG-MRI integration tool. Section 3.3.1 of the *MRIlab* User's Guide, Neuromag P/N NM20419A-A contains a detailed description of this task. Employ the images in the set `.../mri/T1-neuromag/sets/COR.fif` for the alignment. Check the alignment carefully using the digitization data included in the measurement file as described in Section 5.3.1 of the above manual. Save the aligned description file in the same directory as the original description file without the alignment information but under a different name.

Warning: This step is extremely important. If the alignment of the coordinate frames is inaccurate all subsequent processing steps suffer from the error. Therefore, this step should be performed by the person in charge of the study or by a trained technician. Written or photographic documentation of the alignment points employed during the MEG/EEG acquisition can also be helpful.

3.11 Computing the forward solution

After the MRI-MEG alignment has been set, the forward solution, *i.e.*, the magnetic fields and electric potentials at the measurement sensors and electrodes due to dipole sources located on the cortex, can be calculated with help of the convenience script *mne_do_forward_solution*. This utility accepts the following options:

--subject <subject>

Defines the name of the subject. This can be also accomplished by setting the `SUBJECT` environment variable.

--src <name>

Source space name to use. This option overrides the `--spacing` option, if specified. The source space is searched first from the current working directory and then from `$SUBJECTS_DIR/<subject>/bem`. The source space file must be specified exactly, including the `fif` extension.

--spacing <spacing/mm> or `ico-<number>` or `oct-<number>`

This is an alternate way to specify the name of the source space file. For example, if `--spacing 6` is given on the command line, the source space files searched for are `./<subject>-6-src.fif` and `$SUBJECTS_DIR/$SUBJECT/bem/<subject>-6-src.fif`. The first file found is used. Spacing defaults to 7 mm.

--bem <name>

Specifies the BEM to be used. The name of the file can be any of *<name>*, *<name>-bem.fif*, *<name>-bem-sol.fif*. The file is searched for from the current working directory and from *bem*. If this option is omitted, the newest BEM in the *bem* directory is used.

--mri <name>

The name of the MRI description file containing the MEG/MRI coordinate transformation. This file was saved as part of the alignment procedure outlined in Section 3.10. The file is searched for from the current working directory and from *mri/T1-neuromag/sets*. If this option is omitted, the newest file of the form *COR-<user>-<date>.fif* in *mri/T1-neuromag/sets* is used.

--trans <name>

The name of a text file containing the 4 x 4 matrix for the coordinate transformation from head to mri coordinates, see below. If the option **--trans** is present, the **--mri** option is not required.

--meas <name>

This file is the measurement fif file or an off-line average file produced thereof. It is recommended that the average file is employed for evoked-response data and the original raw data file otherwise. This file provides the MEG sensor locations and orientations as well as EEG electrode locations as well as the coordinate transformation between the MEG device coordinates and MEG head-based coordinates.

--fwd <name>

This file will contain the forward solution as well as the coordinate transformations, sensor and electrode location information, and the source space data. A name of the form *<name>-fwd.fif* is recommended. If this option is omitted the forward solution file name is automatically created from the measurement file name and the source space name.

--mindist <dist/mm>

Omit source space points closer than this value to the inner skull surface. Any source space points outside the inner skull surface are automatically omitted. The use of this option ensures that numerical inaccuracies for very superficial sources do not cause unexpected effects in the final current estimates. Suitable value for this parameter is of the order of the size of the triangles on the inner skull surface. If you employ the seglab software to create the triangulations, this value should be about equal to the wish for the side length of the triangles.

--megonly

Omit EEG forward calculations.

--megonly

Omit MEG forward calculations.

--overwrite

Overwrite the possibly existing forward model file.

Tip: If the standard MRI description file and BEM file selections are appropriate and the 7-mm source space grid spacing is appropriate, only the `--meas` option is necessary. If EEG data is not used `--megonly` option should be included.

Tip: If it is conceivable that the current-density transformation will be incorporated into the inverse operator, specify a source space with patch information for the forward computation. This is not mandatory but saves a lot of time when the inverse operator is created, since the patch information does not need to be created at that stage.

Tip: The MEG head to MRI transformation matrix specified with the `--trans` option should be a text file containing a 4-by-4 matrix:

$$\mathbf{T} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & x_0 \\ R_{13} & R_{13} & R_{13} & y_0 \\ R_{13} & R_{13} & R_{13} & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

defined so that if the augmented location vectors in MRI head and MRI coordinate systems are denoted by $\mathbf{r}_{\text{head}} = \begin{bmatrix} x_{\text{head}} & y_{\text{head}} & z_{\text{head}} & 1 \end{bmatrix}$ and $\mathbf{r}_{\text{MRI}} = \begin{bmatrix} x_{\text{MRI}} & y_{\text{MRI}} & z_{\text{MRI}} & 1 \end{bmatrix}$, respectively,

$$\mathbf{r}_{\text{MRI}} = \mathbf{T} \mathbf{r}_{\text{head}}$$

Note: It is not possible to calculate the EEG forward solution with a single-layer BEM.

3.12 Setting up the noise-covariance matrix

The MNE software employs an estimate of the noise-covariance matrix to weight the channels correctly in the calculations. The noise-covariance matrix provides information about field and potential patterns representing uninteresting noise sources of either human or environmental origin.

The noise covariance matrix can be calculated in several ways:

1. Employ the individual epochs during off-line averaging to calculate the full noise covariance matrix. This is the recommended approach for evoked responses.
2. Employ empty room data (collected without the subject) to calculate the full noise covariance matrix. This is recommended for analyzing ongoing spontaneous activity.
3. Employ a section of continuous raw data collected in the presence of the subject to calculate the full noise covariance matrix. This is the recommended approach for analyzing epileptic activity. The data used for this purpose should be free of technical artifacts and epileptic activity of interest. The length of the data segment employed should be at least 20 seconds. One can also use a long (> 200 s) segment of data with epileptic spikes present provided that the spikes occur infrequently and that the segment is apparently stationary with respect to background brain activity.

The new raw data processing tools, *mne_browse_raw* or *mne_process_raw* include computation of noise-covariance matrices both from raw data and from individual epochs. For details, see Chapter 4.

3.13 Calculating the inverse operator decomposition

The MNE software doesn't calculate the inverse operator explicitly but rather computes an SVD of a matrix composed of the noise-covariance matrix, the result of the forward calculation, and the source covariance matrix. This approach has the benefit that the regularization parameter ('SNR') can be adjusted easily when the final source estimates or dSPMs are computed. For mathematical details of this approach, please consult Section 6.2.

This computation stage is facilitated by the convenience script *mne_do_inverse_operator*. It invokes the program *mne_inverse_operator* with appropriate options, derived from the command line of *mne_do_inverse_operator*.

mne_do_inverse_operator assumes the following options:

--fwd <name of the forward solution file>

This is the forward solution file produced in the computations step described in Section 3.11.

--meg

Employ MEG data in the inverse calculation. If neither *--meg* nor *--eeg* is set only MEG channels are included. If either *--meg* or *--eeg* is set the the channels specified by these options are included.

--eeg

Employ EEG data in the inverse calculation. If neither `--meg` nor `--eeg` is set only MEG channels are included. If either `--meg` or `--eeg` is set the the channels specified by these options are included.

--fixed

Use fixed source orientations normal to the cortical mantle. By default, the source orientations are not constrained. If `--fixed` is specified, the `--loose` flag is ignored.

--loose <amount>

Use a 'loose' orientation constraint. This means that the source covariance matrix entries corresponding to the current component normal to the cortex are set equal to one and the transverse components are set to `<amount>`. Recommended value of amount is 0.2...0.6. The value 0.4 works well in most instances.

--depth

Employ depth weighting with the standard settings. For details, see Sections 6.2.10 and 6.4.

--bad <name>

A text file to designate bad channels, listed one channel name (like MEG 1933) on each line of the file. Be sure to include both noisy and flat (non-functioning) channels in the list. If bad channels were designated using *mne_mark_bad_channels* in the measurement file which was specified with the `--meas` option when the forward solution was computed, the bad channel information will be automatically included. Also, any bad channel information in the noise-covariance matrix file will be included.

--senscov <name>

Name of the noise-covariance matrix file computed with one of the methods described in Section 3.12. By default, the script looks for a file whose name is derived from the forward solution file by replacing its ending `-<anything>-fwd.fif` by `-cov.fif`. If this file contains a projection operator, which will automatically attached to the noise-covariance matrix by *mne_browse_raw* and *mne_process_raw*, no `--proj` option is necessary because *mne_inverse_operator* will automatically include the projectors from the noise-covariance matrix file.

--megreg <value>

Regularize the MEG part of the noise-covariance matrix by this amount. Suitable values are in the range 0.05...0.2. For details, see Section 6.2.4.

--eegreg <value>

Like `--megreg` but applies to the EEG channels.

--diagnoise

Omit the off-diagonal terms of the noise covariance matrix. This option is irrelevant to most users.

--fmri <name>

With help of this w file, the source covariance matrix can be biased, *i.e.*, an *a priori* weighting can be applied. The source of the bias function is usually fMRI but may be also some other data, provided that the bias function can be expressed as a scalar value on the cortical surface, stored in a w file. It is recommended that this w file is appropriately smoothed in tksurfer or with `mne_smooth_w` to contain nonzero values at all vertices of the triangular tessellation of the cortical surface. The name of the file given is used as a stem of the w files. The actual files should be called `<name>-lh.pri` and `<name>-rh.pri` for the left and right hemisphere bias files, respectively. The application of the bias is discussed in Section 6.2.11.

--fmrithresh <value>

This option is mandatory and has an effect only if a weighting function has been specified with the `--fmri` option. If the value in the *a priori* files falls below this value at a particular source space point, the source covariance matrix values are multiplied by the value specified with the `--fmrioff` option (default 0.1). Otherwise it is left unchanged.

--fmrioff <value>

The value by which the source covariance elements are multiplied if the *a priori* weight falls below the threshold set with `--fmrithresh`, see above.

--srccov <name>

Use this diagonal source covariance matrix. By default the source covariance matrix is a multiple of the identity matrix. This option is irrelevant to most users.

--proj <name>

Include signal-space projection information from this file.

--inv <name>

Save the inverse operator decomposition here. By default, the script looks for a file whose name is derived from the forward solution file by replacing its ending `-fwd.fif` by `<options>-inv.fif`, where `<options>` includes options `--meg`, `--eeg`, and `--fixed` with the double dashes replaced by single ones.

Tip: Thanks to the automatic naming conventions it is usually enough to invoke the inverse operator calculation script as

```
mne_do_inverse_operator --fwd <forward solution name>
```


Important: If bad channels are included in the calculation, strange results may ensue. Therefore, it is recommended that the data to be analyzed is carefully inspected with to assign the bad channels correctly.

Tip: For convenience, we have provided sample bad-channel designation files which can be used in order to ignore all magnetometer or all gradiometer channels in Vectorview measurements. These files are called *vv_grad_only.bad* and *vv_mag_only.bad*, respectively. Both files are located in *\$MNE_ROOT/setup/mne/templates*.

3.14 Analyzing the data

Once all the preprocessing steps described above have been completed, the inverse operator computed can be applied to the MEG and EEG data and the results can be viewed and stored in several ways:

1. The interactive analysis tool *mne_analyze* can be used to explore the data and to produce quantitative analysis results, screen snapshots, and QuickTime™ movie files. For comprehensive information on *mne_analyze*, please consult Chapter 7.
2. The command-line tool *mne_make_movie* can be invoked to produce QuickTime movies and snapshots. *mne_make_movie* can also output the data in the stc (movies) and w (snapshots) formats for subsequent processing. Furthermore, subject-to-subject morphing is included in *mne_make_movie* to facilitate cross-subject averaging and comparison of data among subjects. *mne_make_movie* is described in Section 6.5,
3. The command-line tool *mne_make_movie* can be employed to interrogate source source estimate waveforms from labels (ROIs).
4. The *mne_make_movie* tool can be also used to create movies from stc files and to resample stc files in time.
5. The *mne_compute_raw_inverse* tool can be used to produce fif files containing source estimates at selected ROIs. The input data file can be either a raw data or evoked response MEG/EEG file, see Section 6.6.
6. It is also possible to average the source estimates across subjects as described in Chapter 8.

CHAPTER 4 Processing raw data

4.1 Overview

The purpose of the raw data processor *mne_browse_raw* included in the MNE software is to provide an alternative to the Neuromag signal processor *graph* for simple raw data viewing and processing operations. This include off-line averaging and computation of covariance matrices. If the program is invoked as *mne_process_raw*, no user interface appears. In *mne_process_raw*, command line options are used to specify the filtering parameters as well as averaging and covariance-matrix computation command files for batch processing. In addition, *mne_browse_raw* can be used to view averaged data in the topographical layout with similar functionality to that found in the Neuromag data plotter (*xplotter*).

4.2 Command-line options

This section first describes the options common to *mne_browse_raw* and *mne_process_raw*. Thereafter, options unique to the batch mode (*mne_process_raw*) are listed.

4.2.1 Common options

--version

Show the program version and compilation date.

--help

List the command-line options.

--raw <name>

Specifies the raw data file to be opened. This option is required for batch version, *mne_process_raw*. If a raw data file is not specified for the interactive version, *mne_browse_raw*, and empty interactive browser will open.

--grad <number>

Apply software gradient compensation of the given order (0, 1, 2, or 3) to the data loaded with the **--raw** option. This option is effective

only for data acquired with the CTF and 4D Magnes MEG systems. If orders different from zero are requested for Neuromag data, an error message appears and data are not loaded. Any compensation already existing in the file can be undone or changed to another order by using an appropriate `--grad` options. Possible orders are 0 (No compensation), 1 - 3 (CTF data), and 101 (Magnes data).

--sel <name>

Specifies the channel selection file to be used. This overrides the use of the standard channel selection files, see Section 4.4.2.3.

--filtersize <size>

Adjust the length of the FFT to be applied in filtering. The number will be rounded up to the next power of two. If the size is N , the corresponding length of time is N/f_s , where f_s is the sampling frequency of your data. The filtering procedure includes overlapping tapers of length $N/2$ so that the total FFT length will actually be $2N$. This value cannot be changed after the program has been started.

--highpass <value/Hz>

Highpass filter frequency limit. If this is too low with respect to the selected FFT length and data file sampling frequency, the data will not be highpass filtered. You can experiment with the interactive version to find the lowest applicable filter for your data. This value can be adjusted in the interactive version of the program. The default is 0, i.e., no highpass filter in effect.

--highpassw <value/Hz>

The width of the transition band of the highpass filter. The default is 6 frequency bins, where one bin is $f_s/(2N)$. This value cannot be adjusted in the interactive version of the program.

--lowpass <value/Hz>

Lowpass filter frequency limit. This value can be adjusted in the interactive version of the program. The default is 40 Hz.

--lowpassw <value/Hz>

The width of the transition band of the lowpass filter. This value can be adjusted in the interactive version of the program. The default is 5 Hz.

--filteroff

Do not filter the data. This initial value can be changed in the interactive version of the program.

--digtrig <name>

Name of the digital trigger channel. The default value is 'STI 014'. Underscores in the channel name will be replaced by spaces.

Note: Multiple raw data files can be specified for *mne_process_raw*.

4.2.2 Batch-mode options

These options apply to the batch-mode version, *mne_process_raw* only.

--proj <name>

Specify the name of the file of the file containing a signal-space projection (SSP) operator. If `--proj` options are present the data file is not consulted for an SSP operator. The operator corresponding to average EEG reference is always added if EEG data are present.

--projon

Activate the projections loaded. One of the options `--projon` or `--projoff` must be present on the *mne_process_raw* command line.

--projoff

Deactivate the projections loaded. One of the options `--projon` or `--projoff` must be present on the *mne_process_raw* command line.

--eventsout <name>

List the STI 014 trigger line events to the specified file. Only transitions from zero to a non-zero value are listed. If multiple raw data files are specified, an equal number of `--eventsout` options should be present.

--ave <name>

Specifies the name of an off-line averaging description file. For details of the format of this file, please consult Section 4.8. If multiple raw data files are specified, the number of `--ave` options can be smaller or equal to the number of raw data files. If it is equal, the averaging description file names will be associated with the raw data files in the order given. If it is smaller, the last description file will be used for the remaining raw data files.

--saveavetag <name>

If this option is present and averaging is evoked with the `--ave` option, the outfile and logfile options in the averaging description file are ignored. Instead, trailing `.fif` or `_raw.fif` is removed from the raw data file name and `<name>.fif` or `<name>.log` is appended to create the output and log file names, respectively.

--gave <name>

If multiple raw data files are specified as input and averaging is requested, the grand average over all data files will be saved to `<name>.`

--cov <name>

Specify the name of a description file for covariance matrix computation. For details of the format of this file, please see Section 4.9. If multiple raw data files are specified, the number of **--cov** options can be smaller or equal to the number of raw data files. If it is equal, the averaging description file names will be associated with the raw data files in the order given. If it is smaller, the last description file will be used for the remaining raw data files.

--savecovtag <name>

If this option is present and covariance matrix computation is evoked with the **--cov** option, the outfile and logfile options in the covariance computation description file are ignored. Instead, trailing **.fif** or **_raw.fif** is removed from the raw data file name and **<name>.fif** or **<name>.log** is appended to create the output and log file names, respectively. For compatibility with other MNE software scripts, **--savecovtag -cov** is recommended.

--gcov <name>

If multiple raw data files are specified as input and covariance matrix computation is requested, the grand average over all data files will be saved to **<name>**. The details of the covariance matrix computation are given in Section 4.12.

--save <name>

Save a filtered and optionally downsampled version of the data file to **<name>**. If multiple raw data files are specified, an equal number of **--save** options should be present.

--anon

Do not include any subject information in the output files created with the **--save** option.

--decim <number>

The data are decimated by this factor before saving to the file specified with the **--save** option. For decimation to succeed, the data must be lowpass filtered to less than third of the sampling frequency effective after decimation.

4.3 The user interface



Figure 4.1 The user interface of *mne_browse_raw*

The *mne_browse_raw* user interface contains the following areas:

1. The menu bar.
2. The data display area.
3. Viewing and averaging tools.
4. Message line.

The viewing and averaging tools allow quick browsing of the raw data with triggers, adding new triggers, and averaging on a single trigger.

4.4 Description of menus

4.4.1 The *File* menu

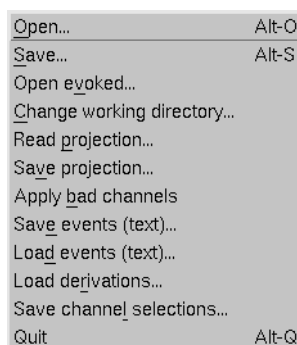


Figure 4.2 The contents of the *File* menu.

The File menu pane is shown in Figure 4.2.

4.4.1.1 Open

Selecting *Open* from file menu pops up the dialog shown in Figure 4.3.

The *Keep the initial skip* button controls how the initial segment of data not stored in the raw data file is handled. During the MEG acquisition data are collected continuously but saving to the raw data file is controlled by the *Record raw* button. Initial skip refers to the segment of data between the start of the recording and the first activation of *Record raw*. If *Keep initial skip* is set, this empty segment is taken into account in timing, otherwise time zero is set to the beginning of the data stored to disk.

The options under *Software gradient compensation* allow selection of the compensation grade for the data. These selections apply to the CTF data only. The standard choices are *No compensation* and *Third-order gradient*. If other than *No compensation* is attempted for non-CTF data, an error will be issued. The compensation selection affects the averages and noise-covariance matrices subsequently computed. The desired compensation takes effect independent of the compensation state of the data in the file, *i.e.*, already compensated data can be uncompensated and vice versa. For more information on software gradient compensation please consult Section 9.2.4.

When a raw data file is opened, trigger channel STI 014 is scanned for events. For large files this may take a while.

Note: After scanning the trigger channel for events, *mne_browse_raw* and *mne_process_raw* produce a fif file containing the event information. This file will be called *<raw data file name without .fif>-eve.fif*. If the

same raw data file is opened again, this file will be consulted for event information thus making it unnecessary to scan through the file for trigger line events.

Tip: You can produce the fif event file by running `mne_process_raw` as follows: `mne_process_raw --raw <raw data file>`.

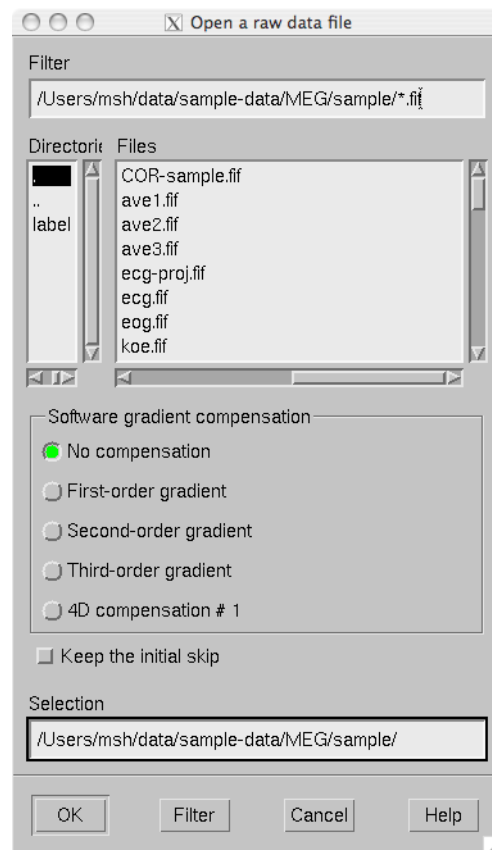


Figure 4.3 The *Open* dialog.

4.4.1.2 Open evoked

This menu item brings up a standard file selection dialog to load evoked-response data from files. All data sets from a file are loaded automatically and display in the full view window, see Section 4.7. The data loaded are affected by the scale settings, see, Section 4.4.2.2, the filter, see Section 4.4.2.1, and the options selected in the *Manage averages* dialog, see Section 4.4.2.9.

4.4.1.3 Save

It is possible to save filtered and projected data into a new raw data file. When you invoke the save option from the file menu, you will be prompted for the output file name and a downsampling factor. The sam-

pling frequency after downsampling must be at least three times the low-pass filter corner frequency. For downsampling and saving options in *mne_process_raw*, see Section 4.2.2.

4.4.1.4 Change working directory

Brings up a file selection dialog which allows changing of the working directory.

4.4.1.5 Read projection

Selecting *Read projection...* from the *File* menu, pops up a dialog to enter a name of a file containing a signal-space projection operator to be applied to the data. There is an option to keep existing projection items.

Note: Whenever EEG channels are present in the data, a projection item corresponding to the average EEG reference is automatically added.

4.4.1.6 Save projection

The *Save projection...* item in the *File* menu pops up a dialog to save the present projection operator into a file.

4.4.1.7 Apply bad channels

Applies the current selection of bad channels to the currently open raw file.

4.4.1.8 Save events (text)

Brings up a a dialog to save all or selected types of events into a text file. This file can be edited and used in the averaging and covariance matrix computations as an input file to specify the time points of events, see Section 4.10.

4.4.1.9 Save channel selections

This menu choice brings up a dialog to save the current channel selections. This is particularly useful if the standard set of selections has been modified as explained in Section 4.4.2.3.

4.4.1.10 Load derivations

This menu choice allows loading of channel derivation data files created with the *mne_make_derivations* utility, see Section 11.5. Most common use of derivations is to calculate differences between EEG channels, *i.e.*, bipolar EEG data. Since any number of channels can be included in a derivation with arbitrary use, other applications are possible as well. Before a derivation is accepted to use, the following criteria have to be met:

1. All channels to be combined into a single derivation must have identical units of measure.
2. All channels in a single derivation have to be of the same kind, *e.g.*, MEG channels or EEG channels.
3. All channels specified in a derivation have to be present in the currently loaded data set.

Multiple derivation data files can be loaded by specifying the *Keep previous derivations option* in the dialog that specifies the derivation file to be loaded. After a derivation file has been successfully loaded, a list of available derivations will be shown in a message dialog.

Each of the derived channels has a name specified when the derivation file was created. The derived channels can be included in channel selections, see Section 4.4.2.3. At present, derived channels cannot be displayed in topographical data displays. Derived channels are not included in averages or noise-covariance matrix computations.

Note: If the file `$HOME/.mne/mne_browse_raw-deriv.fif` exists and contains derivation data, it is loaded automatically when *mne_browse_raw* starts.

4.4.1.11 Quit

Exits the program without questions asked.

4.4.2 The *Adjust* menu

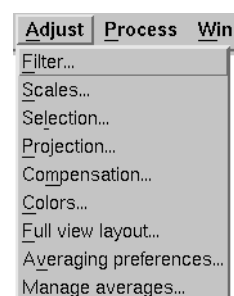


Figure 4.4 The contents of the *Adjust* menu.

The contents of the *Adjust* menu is shown in Figure 4.4. This menu allows the manipulation of various settings of the program,

4.4.2.1 Filter

Selecting *Filter...* from the *Adjust* menu pops up the dialog shown in Figure 4.5.

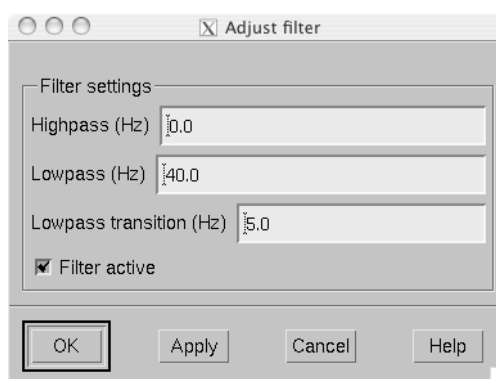


Figure 4.5 The filter adjustment dialog.

The items in the dialog have the following functions:

Highpass (Hz)

The half-amplitude point of the highpass filter. The width of the transition from zero to one can be specified with the `--high-passw` command-line option, see Section 4.2. Lowest feasible highpass value is constrained by the length of the filter and sampling frequency. You will be informed when you press *OK* or *Apply* if the selected highpass could not be realized. The default value zero means no highpass filter is applied in addition to the analog highpass present in the data.

Lowpass (Hz)

The half-amplitude point of the lowpass filter.

Lowpass transition (Hz)

The width of the \cos^2 -shaped transition from one to zero, centered at the *Lowpass* value.

Filter active

Selects whether or not the filter is applied to the data.

The filter is realized in the frequency domain and has a zero phase shift. At the end and the beginning there might be a short filter artefact if the dc level of the data is significantly different from zero.

Note: The filter affects both the raw data and evoked-response data loaded from files. However, the averages computed in *mne_browse_raw* and shown in the topographical display are not refiltered if the filter is changed after the average was computed.

4.4.2.2 Scales

Selecting *Scales...* from the *Adjust* menu pops up the dialog shown in Figure 4.6.

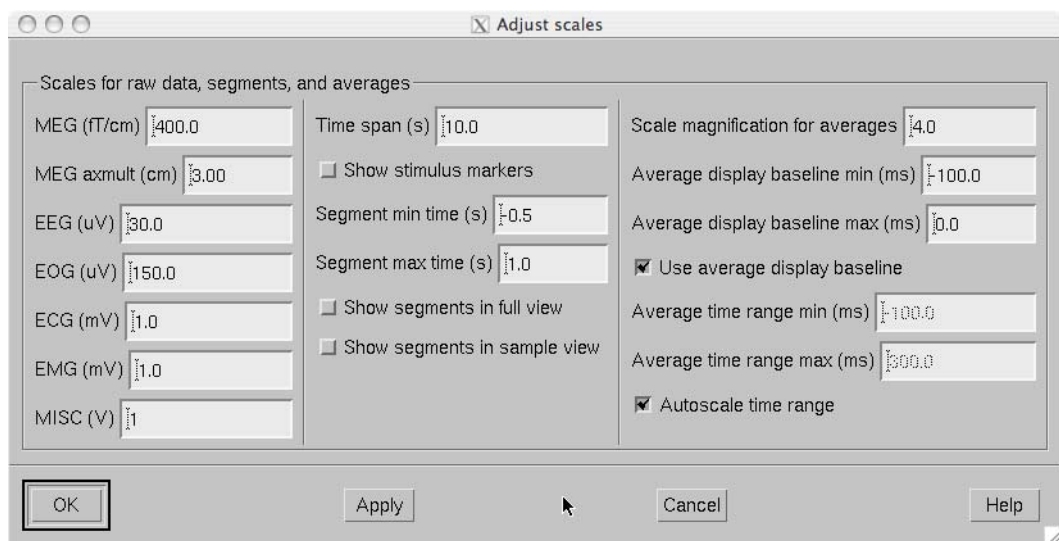


Figure 4.6 The *Scales* dialog.

The items in the dialog have the following functions:

MEG (fT/cm)

Sets the scale for MEG planar gradiometer channels in fT/cm. All scale values are defined from zero to maximum, *i.e.*, the viewport where signals are plotted in have the limits $\pm <scale\ value>$.

MEG axmult (cm)

The scale for MEG magnetometers and axial gradiometers is defined by multiplying the gradiometer scale by this number, yielding units of fT.

EEG (μV)

The scale for EEG channels in μV .

EOG (μV)

The scale for EOG channels in μV .

ECG (mV)

The scale for ECG channels in mV.

EMG (mV)

The scale for EMG channels in mV.

MISC (V)

The scale for MISC channels in V.

Time span (s)

The length of raw data displayed in the main window at a time.

Show stimuli markers

Draw vertical lines at time points where the digital trigger channel has a transition from zero to a nonzero value.

Segment min. time (s)

It is possible to show data segments in the topographical (full view) layout, see below. This parameter sets the starting time point, relative to the selected time, to be displayed.

Segment max. time (s)

This parameter sets the ending time point, relative to the current time, to be displayed in the topographical layout.

Show segments in full view

Switches on the display of data segments in the topographical layout.

Show segments in sample view

Switches on the display of data segments in a “sidebar” to the right of the main display.

Scale magnification for averages

For average displays, the scales are made more sensitive by this factor.

Average display baseline min (ms)

Sets the lower time limit for the average display baseline. This setting does not affect the averages stored.

Average display baseline max (ms)

Sets the upper time limit for the average display baseline. This setting does not affect the averages stored.

Use average display baseline

Switches the application of a baseline to the displayed averages on and off.

Average time range min (ms)

Sets the minimum time for the average display. This setting is inactive if *Autoscale time range* is on.

Average time range max (ms)

Sets the maximum time for the average data display. This setting is inactive if *Autoscale time range* is on.

Autoscale time range

Set the average display time range automatically to be long enough to accommodate all data.

4.4.2.3 Selection

Brings up a dialog to select channels to be shown in the main raw data display. This dialog also allows modification of the set of channel selections as described below.

By default, the available selections are defined by the file `$MNE_ROOT/setup/mne/mne_browse_raw/mne_browse_raw.sel`. This default channel selection file can be modified by copying the file into `$HOME/.mne/mne_browse_raw.sel`. The format of this text file should be self explanatory.

The channel selection dialog is shown in Figure 4.7. The number of buttons shown and their labels depend on the content of your channel selection file.

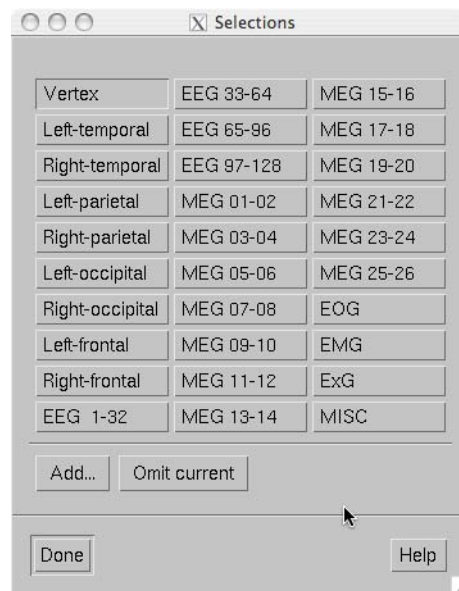


Figure 4.7 The channel selection dialog.

The two buttons below the channel selection buttons facilitate the modification of the selections:

Add...

Brings up the selection dialog shown in Figure 4.8 to create new channel selections.

Omit current

Delete the current channel selection. Deletion only affects the selections in the memory of the program. To save the changes permanently into a file, use *Save channel selections...* in the *File* menu, see Section 4.4.1.9.



Figure 4.8 Dialog to create a new channel selection.

The components of the selection creation dialog shown in Figure 4.8 have the following functions:

List of channel names

The channels selected from this list will be included in the new channel selection. A selection can be extended with the control key. A range of channels can be selected with the shift key. The list contains both the original channels actually present in the file and the

names of the channels in currently loaded derivation data, see Section 4.4.1.10.

Regexp:

This provides another way to select channels. By entering here a regular expression as defined in IEEE Standard 1003.2 (POSIX.2), all channels matching it will be selected and added to the present selection. An empty expression unselects all items in the channel list. Some useful regular expressions are listed in Table 4.1. In the present version, regular matching does not look at the derived channels.

Name:

This text field specifies the name of the new selection.

Select

Select the channels specified by the regular expression. The same effect can be achieved by entering return in the *Regexp*:

Add

Add a new channel selection which contains the channels selected from the channel name list. The name of the selection is specified with the *Name*: text field.

Regular expression	Meaning
MEG	Selects all MEG channels.
EEG	Selects all EEG channels.
MEG.*1\$	Selects all MEG channels whose names end with the number 1, <i>i.e.</i> , all magnetometer channels.
MEG.*[2,3]\$	Selects all MEG gradiometer channels.
EEG STI 014	Selects all EEG channels and stimulus channel STI 014.
^M	Selects all channels whose names begin with the letter M.

Table 4.1 Examples of regular expressions for channel selections

4.4.2.4 Compensation

Brings up a dialog to select software gradient compensation. This overrides the choice made at the open time. For details, see Section 4.4.1.1, above.

4.4.2.5 Projection

Lists the currently available signal-space projection (SSP) vectors and allows the activation and deactivation of items. For more information on SSP, see Section 4.11.

4.4.2.6 Colors

Shows a dialog which allows changes to the default colors of various display items.

4.4.2.7 Full view layout

Shows a selection of available layouts for the topographical views (full view and average display). The default layouts reside in `$MNE_ROOT/setup/mne_analyze/lout`. In addition any layout files residing in `$HOME/.mne/lout` are listed. The format of the layout files is the same as for the Neuromag programs *xplotter* and *xfit*.

4.4.2.8 Averaging preferences

Selecting *Averaging preferences...* from the *Adjust* menu pops up the dialog shown in Figure 4.9. These settings apply only to the simple averages calculated with help of tools residing just below the main raw data display, see Section 4.6.

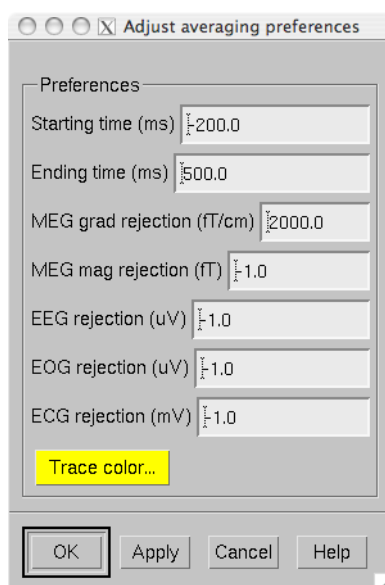


Figure 4.9 Averaging preferences.

The items in the dialog have the following functions:

Starting time (ms)

Beginning time of the epoch to be averaged (relative to the trigger).

Ending time (ms)

Ending time of the epoch to be averaged.

MEG grad rejection (fT/cm)

Rejection criterion for MEG planar gradiometers. If the peak-to-peak value of any planar gradiometer epoch exceed this value, it will be omitted. A negative value turns off rejection for a particular channel type.

MEG mag rejection (fT)

Rejection criterion for MEG magnetometers.

EEG rejection (μ V)

Rejection criterion for EEG channels.

EOG rejection (μ V)

Rejection criterion for EOG channels.

ECG rejection (mV)

Rejection criterion for ECG channels.

Trace color

The color assigned for the averaged traces in the display can be adjusted by pressing this button.

4.4.2.9 Managing averages

This selection pops up a dialog which allows the management of computed averages. The controls in the dialog, shown in Figure 4.10, allow the following:

1. Select which categories (conditions) are displayed in the average view.
2. Select the colors of the traces.
3. Viewing the log information accumulated in the averaging process.
4. Saving of averaged data.
5. Setting the active vectors for signal-space projection if the data were loaded from a file.
6. Setting the current software gradient compensation for data loaded from a file.

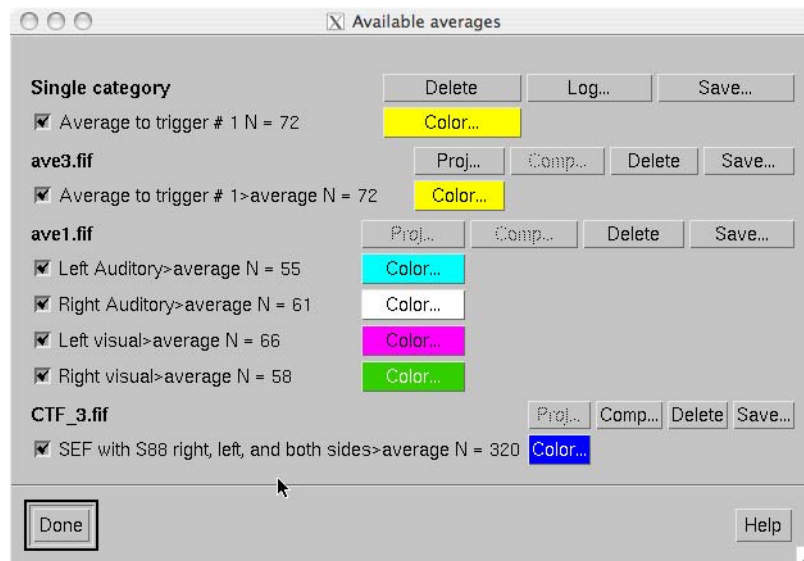


Figure 4.10 The dialog for managing available averages.

In the example of Figure 4.10, the first item is an average computed within *mne_browse_raw*, the second one contains data loaded from a file with signal-space projection data available, the third one demonstrates multiple data sets loaded from a file with neither projection nor software gradient compensation available, and the last one is a data set loaded from file with software gradient compensation data present.

4.4.3 The *Process* menu

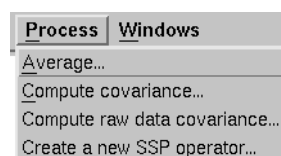


Figure 4.11 The contents of the *Process* menu.

The contents of the *Process* menu is shown in Figure 4.11. This menu accesses the following operations:

1. Averaging according to a description file,
2. Computation of a covariance matrix according to a description file,
3. Computation of a covariance matrix from continuous raw data, and
4. Estimation of the noise subspace for SSP.

4.4.3.1 Averaging

The *Average...* menu item pops up a file selection dialog to access a description file for batch-mode averaging. The structure of these files is described in Section 4.8. All parameters for the averaging are taken from the description file, *i.e.*, the parameters set in the averaging preferences dialog (Section 4.4.2.8) do not effect the result.

4.4.3.2 Computation of a covariance matrix

The *Compute covariance...* menu item pops up a file selection dialog to access a description file which specifies the options for the computation of a covariance matrix. The structure of these files is described in Section 4.9.

4.4.3.3 Computation of a covariance matrix from raw data

The *Compute raw data covariance...* menu item pops up a dialog which specifies a time range for raw data covariance matrix computation and the file to hold the result. If a covariance matrix is computed in this way, the rejection parameters specified in averaging preferences are in effect. For description of the rejection parameters, see Section 4.4.2.8.

4.4.3.4 Estimation of noise subspace for SSP

The *Create a new SSP operator...* pops up a window which allows the specification of a time range to be employed in the calculation of a raw data covariance matrix. Normally, you should use empty room data for this computation. For the estimation of the covariance matrix any existing projection will be temporarily switched off. Remember to inspect your data for bad channels and select an appropriate filter setting before creating a new SSP operator. It is usually desirable to employ a generous pass-band.

Once the covariance matrix is ready, the parts corresponding to magnetometer or axial gradiometer and planar gradiometer channels are separated and the corresponding eigenvectors and eigenvalues are computed. Once complete, a projection selector with eight magnetometer eigenvectors and five planar gradiometer eigenvectors is displayed.

Using the projection selector, you can experiment which vectors have a significant effect on the noise level of the data. You should strive for using a minimal number of vectors. When the selection is complete, you can click *Accept* to introduce this selection of vectors as the new projection operator. *Discard* abandons the set of calculated vectors. Whenever EEG channels are present in the data, a projection item corresponding to the

average EEG reference is automatically added when a new projection operator is introduced.

Note: The new projection data created in *mne_browse_raw* is not automatically copied to the data file. You need to create a standalone projection file from *File/Save projection...* to save the new projection data and load it manually after the data file has been loaded if you want to include in any subsequent analysis.

Tip: For the Martinos Center Vectorview data it is sufficient to employ a three-dimensional noise subspace with projection applied to magnetometers only. This means that you can select the first three *Axial* components when creating a new projection operator.

4.4.4 The *Windows* menu

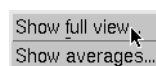


Figure 4.12 The *Windows* menu.

The contents of the *Windows* menu is shown in Figure 4.12. This menu pops up one of the topographical displays: the *Full view* or the *Average* display, see Section 4.7.

4.4.5 The *Help* menu

The contents of the *Help* menu is shown in Figure 4.13:

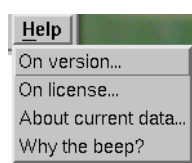


Figure 4.13 The *Help* menu.

On version...

Displays the version and compilation date of the program.

On license...

Displays the license information.

About current data...

Displays essential information about the currently loaded data set.

Why the beep?

In some simple error situations, *mne_browse_raw* does not pop up an error dialog but refuses the action and rings the bell. The reason for this can be displayed through this help menu item.

4.5 The raw data display

The main data displays shows a section of the raw data in a strip-chart recorder format. The names of the channels displayed are shown on the left. The selection of channels is controlled from the selection dialog, see Section 4.4.2.3. The length of the data section displayed is controlled from the scales dialog (Section 4.4.2.2) and the filtering from the filter dialog (Section 4.4.2.1). A signal-space projection can be applied to the data by loading a projection operator (Section 4.4.1.5). The selection of the projection operator items is controlled from the projection dialog described in Section 4.4.2.5.

The control and browsing functions integrated with the main data display are:

1. Selection of bad channels. If you click on a channel name the corresponding channel is marked bad or reinstated as an acceptable one. A channel marked bad is not considered in the artefact rejection procedures in averaging and it is omitted from the signal-space projection operations.
2. Browsing through the data. The section displayed can be selected from the scroll bar at the bottom of the display,
3. Selection of time points. When you click on the data with the left button, a vertical marker appears. If *Show segments in full view* and/or *Show segments in sample view* is active in the scales dialog (see Section 4.4.2.2), a display of an epoch of data specified in the scales dialog will appear. For more information on full view, see Section 4.7. Multiple time points can be selected by holding the control key down when clicking. If multiple time points are selected several samples will be shown in the sample and/or full view, aligned at the picked time point. The toolbar offers functions to operate on the selected time points, see Section 4.6.
4. Range selection. If you drag on the signals with the left mouse button and the shift key down, a range of times will be selected and displayed in the sample and/or full view. Note: All previous selections are cleared by this operation.
5. Making hardcopies. The right mouse button invokes a popup menu which allows saving of the display in various formats.

Note: When selecting bad channels, switch the signal-space projection off from the projection dialog. Otherwise bad channels may not be easily recognizable.

4.6 The toolbar

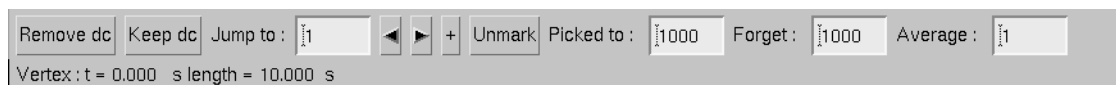


Figure 4.14 The toolbar controls.

The toolbar controls are shown in Figure 4.14. They perform the following functions:

Remove dc

Remove the dc offset from the signals for display. This does not affect the data used for averaging and noise-covariance matrix computation.

Keep dc

Return to the original true dc levels.

Jump to

Enter a value of a trigger to be searched for. The arrow buttons jump to the next event of this kind. A selection is also automatically created and displayed as requested in the scales dialog, see Section 4.4.2.2. If the '+' button is active, previous selections are kept, otherwise they are cleared.

Unmark

Unmark the stimulus marked when the 'Jump to' option was invoked.

Picked to

Make user events with this event number at all picked time points.

Forget

Forget desired user events.

Average

Compute an average to this event.

4.7 Topographical data displays

Segments of data can be shown in a topographical layout in the *Full view* window, which can be requested from the *Scale* dialog or from the *Windows* menu. Another similar display is available to show the averaged data. The topographical layout to use is selected from *Adjust/Full view layout...*, which brings up a window with a list of available layouts. The default layouts reside in `$MNE_ROOT/setup/mne_analyze/lout`.

In addition any layout files residing in `$HOME/.mne/layout` are listed. The format of the layout files is the same as for the Neuromag programs *xplotter* and *xfit*. A custom EEG layout can be easily created with the *mne_make_eeg_layout* utility, see Section 11.6.

Several actions can be performed with the mouse in the topographical data display:

Left button

Shows the time and the the channel name at the cursor at the bottom of the window.

Left button drag with shift key

Enlarge the view to contain only channels in the selected area.

Right button

Brings up a popup menu which gives a choice of graphics output formats for the current topographical display.

4.8 Description files for off-line averaging

For averaging tasks more complex than those involving only one trigger, the averaging parameters are specified with help of a text file. This section describes the format of this file. A sample averaging file can be found in `$MNE_ROOT/setup/mne/mne_browse_raw/templates`.

4.8.1 Overall format

Any line beginning with the pound sign (#) in this description file is a comment. Each parameter in the description file is defined by a keyword usually followed by a value. Text values consisting of multiple words, separated by spaces, must be included in quotation marks. The case of the keywords in the file does not matter. The ending `.ave` is suggested for the average description files.

The general format of the description file is:

```
average {
    <common parameters>
    category {
        <category definition parameters>
    }
    ....
}
```

The file may contain arbitrarily many categories. The word `category` interchangeable with `condition`.

Warning: Due to a bug that existed in some versions of the Neuromag acquisition software, the trigger line 8 is incorrectly decoded on trigger channel STI 014. This can be fixed by running `mne_fix_stim14` on the raw data file before using `mne_browse_raw` or `mne_process_raw`. The bug has been fixed on Nov. 10, 2005.

4.8.2 Common parameters

The average definition starts with the common parameters. They include:

outfile <name>

The name of the file where the averages are to be stored. In interactive mode, this can be omitted. The resulting average structure can be viewed and stored from the *Manage averages* window.

eventfile <name>

Optional file to contain event specifications. If this file is present, the trigger events in the raw data file are ignored and this file is consulted instead. The event file format is described in Section 4.10.

logfile <name>

This optional file will contain detailed information about the averaging process. In the interactive mode, the log information can be viewed from the *Manage averages* window.

gradReject <value / T/m>

Rejection limit for MEG gradiometer channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the gradiometer channels, the epoch will be omitted from the average.

magReject <value / T>

Rejection limit for MEG magnetometer channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the magnetometer channels, the epoch will be omitted from the average.

eegReject <value / V>

Rejection limit for EEG channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the EEG channels, the epoch will be omitted from the average.

eogReject *<value / V>*

Rejection limit for EOG channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the EOG channels, the epoch will be omitted from the average.

ecgReject *<value / V>*

Rejection limit for ECG channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the ECG channels, the epoch will be omitted from the average.

name *<text>*

A descriptive name for this set of averages. If the name contains multiple words, enclose it in quotation marks “like this”. The name will appear in the average manager window listing in the interactive version of the program and as a comment in the processed data section in the output file.

4.8.3 Category definition

A category (condition) is defined by the parameters listed in this section.

event *<number>*

The zero time point of an epoch to be averaged is defined by a transition from zero to this number on the trigger channel STI 014. The interpretation of the values on the trigger channel can be further modified by the ignore keyword.

ignore *<number>*

If this parameter is specified the selected bits on trigger channel values can be mask (set to zero) out prior to checking for an existence of an event. For example, to ignore the values of trigger input lines three and eight, specify `ignore 132` ($2^2 + 2^7 = 132$).

tmin *<time / s>*

Beginning time point of the epoch.

tmax *<time / s>*

End time point of the epoch.

bmin *<time / s>*

Beginning time point of the baseline. If both `bmin` and `bmax` parameters are present, the baseline defined by this time range is subtracted from each epoch before they are added to the average. In addition, the standard error of mean is computed for this category and included in the output fif file.

basemin *<time / s>*

Synonym for `bmin`.

bmax *<time / s>*

End time point of the baseline.

basemax *<time / s>*

Synonym for bmax.

name *<text>*

A descriptive name for this category. If the name contains multiple words, enclose it in quotation marks “like this”. The name will appear in the average manager window listing in the interactive version of the program and as a comment averaging category section in the output file.

abs

Calculate the absolute values of the data in the epoch before adding it to the average.

4.9 Description files for covariance matrix computation

Covariance matrix computation is controlled by a another description file, very similar to the average definition. A example of a covariance description file can be found in the directory `$MNE_ROOT/setup/mne/mne_browse_raw/templates`.

4.9.1 Overall format

Any line beginning with the pound sign (#) in this description file is a comment. Each parameter in the description file is defined by a keyword usually followed by a value. Text values consisting of multiple words, separated by spaces, must be included in quotation marks. The case of the keywords in the file does not matter. The ending `.cov` is suggested for the covariance-matrix description files.

The general format of the description file is:

```
cov {  
    <common parameters>  
    def {  
        <covariance definition parameters>  
    }  
  
    ....  
}
```

The file may contain arbitrarily many covariance definitions, starting with `def`.

Warning: Due to a bug that existed in some versions of the Neuromag acquisition software, the trigger line 8 is incorrectly decoded on trigger channel STI 014. This can be fixed by running *mne_fix_stim14* on the raw data file before using *mne_browse_raw* or *mne_process_raw*. This bug has been fixed in the acquisition software at the Martinos Center on Nov. 10, 2005.

4.9.2 Common parameters

The average definition starts with the common parameters. They include:

outfile <name>

The name of the file where the covariance matrix is to be stores. This parameter is mandatory.

eventfile <name>

Optional file to contain event specifications. If this file is present, the trigger events in the raw data file are ignored and this file is consulted instead. The event file format is described in Section 4.10.

logfile <name>

This optional file will contain detailed information about the averaging process. In the interactive mode, the log information can be viewed from the *Manage averages* window.

gradReject <value / T/m>

Rejection limit for MEG gradiometer channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the gradiometer channels, the epoch will be omitted from the average.

magReject <value / T>

Rejection limit for MEG magnetometer channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the magnetometer channels, the epoch will be omitted from the average.

eegReject <value / V>

Rejection limit for EEG channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the EEG channels, the epoch will be omitted from the average.

eogReject <value / V>

Rejection limit for EOG channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the EOG channels, the epoch will be omitted from the average.

ecgReject *<value / V>*

Rejection limit for ECG channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the ECG channels, the epoch will be omitted from the average.

4.9.3 Covariance definitions

The covariance definitions starting with `def` specify the epochs to be included in the computation of the covariance matrix.

event *<number>*

The zero time point of an epoch to be averaged is defined by a transition from zero to this number on the trigger channel STI 014. The interpretation of the values on the trigger channel can be further modified by the `ignore` keyword. If the event parameter is missing or set to zero, the covariance matrix is computed over a section of the raw data, defined by the `tmin` and `tmax` parameters. If a definition with `event 0` or missing event parameter exists in the description file, all others are ignored.

ignore *<number>*

If this parameter is specified the selected bits on trigger channel values can be mask (set to zero) out prior to checking for an existence of an event. For example, to ignore the values of trigger input lines three and eight, specify `ignore 132` ($2^2 + 2^7 = 132$).

tmin *<time / s>*

Beginning time point of the epoch. If the `event` parameter is zero or missing, this defines the beginning point of the raw data range to be included.

tmax *<time / s>*

End time point of the epoch. If the `event` parameter is zero or missing, this defines the end point of the raw data range to be included.

bmin *<time / s>*

It is possible to remove a baseline from the epochs before they are included in the covariance computation. This parameter defines the starting point of the baseline. This feature can be employed to avoid overestimation of noise in the presence of low-frequency drifts. Setting of `bmin` and `bmax` is always recommended for epoch-based computation.

basemin *<time / s>*

Synonym for `bmin`.

bmax *<time / s>*

End time point of the baseline, see above.

basemax *<time / s>*
 Synonym for bmax.

4.10 Event files

An event file contains information about transitions on the digital trigger line in a raw data file. Any lines beginning with the pound sign (#) are considered as comments. The format of the event file data is:

<sample> *<time>* *<from>* *<to>*

where

<sample> is the sample number. Note that the setting of the *Keep initial skip* in the open dialog, see Section 4.4.1.1, affects the sample number saved here. If the initial skip is kept, the length of time elapsed before the *Record raw* button was depressed during the acquisition is taken into account when computing sample numbers. If the skip is omitted, the first sample in the raw data file is sample number zero.

<time> is the time from the beginning of the file to this sample in seconds. The setting of *Keep initial skip* affects the times stored and their interpretation as well.

<from> is the value of the digital trigger channel at *<sample>*-1.

<to> is the value of the digital trigger channel at *<sample>*.

When an event file is read back, the *<sample>* value will be primarily used to specify the time. If you want the *<time>* to be converted to the sample number instead, specify a negative value for *<sample>*.

4.11 The Signal-Space Projection (SSP) method

The Signal-Space Projection (SSP) is one approach to rejection of external disturbances in software. The section presents some relevant details of this method.

4.11.1 General concepts

Unlike many other noise-cancellation approaches, SSP does not require additional reference sensors to record the disturbance fields. Instead, SSP relies on the fact that the magnetic field distributions generated by the sources in the brain have spatial distributions sufficiently different from those generated by external noise sources. Furthermore, it is implicitly assumed that the linear space spanned by the significant external noise patterns has a low dimension.

Without loss of generality we can always decompose any n -channel measurement $\mathbf{b}(t)$ into its signal and noise components as

$$\mathbf{b}(t) = \mathbf{b}_s(t) + \mathbf{b}_n(t)$$

Further, if we know that $\mathbf{b}_n(t)$ is well characterized by a few field patterns $\mathbf{b}_1 \dots \mathbf{b}_m$, we can express the disturbance as

$$\mathbf{b}_n(t) = \mathbf{U}\mathbf{c}_n(t) + \mathbf{e}(t) ,$$

where the columns of \mathbf{U} constitute an orthonormal basis for $\mathbf{b}_1 \dots \mathbf{b}_m$, $\mathbf{c}_n(t)$ is an m -component column vector, and the error term $\mathbf{e}(t)$ is small and does not exhibit any consistent spatial distributions over time, *i.e.*, $\mathbf{C}_e = E\{\mathbf{e}\mathbf{e}^T\} = \mathbf{I}$. Subsequently, we will call the column space of \mathbf{U} the noise subspace. The basic idea of SSP is that we can actually find a small basis set $\mathbf{b}_1 \dots \mathbf{b}_m$ such that the conditions described above are satisfied. We can now construct the orthogonal complement operator

$$\mathbf{P}_\perp = \mathbf{I} - \mathbf{U}\mathbf{U}^T$$

and apply it to $\mathbf{b}(t)$ yielding

$$\mathbf{b}(t) \approx \mathbf{P}_\perp \mathbf{b}_s(t) ,$$

since $\mathbf{P}_\perp \mathbf{b}_n(t) = \mathbf{P}_\perp \mathbf{U}\mathbf{c}_n(t) \approx 0$. The projection operator \mathbf{P}_\perp is called the signal-space projection operator and generally provides considerable rejection of noise, suppressing external disturbances by a factor of 10 or more. The effectiveness of SSP depends on two factors:

1. The basis set $\mathbf{b}_1 \dots \mathbf{b}_m$ should be able to characterize the disturbance field patterns completely and
2. The angles between the noise subspace space spanned by $\mathbf{b}_1 \dots \mathbf{b}_m$ and the signal vectors $\mathbf{b}_s(t)$ should be as close to $\pi/2$ as possible.

Covariance matrix of magnetometer data

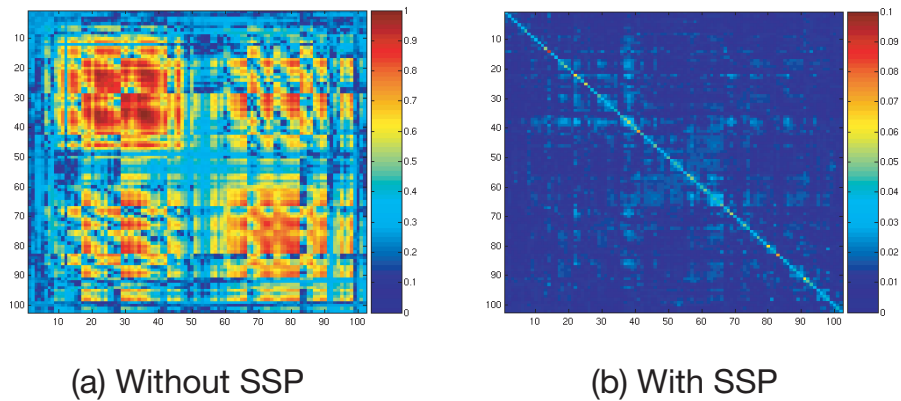


Figure 4.15 An example of the effect of SSP. The covariance matrix \mathbf{C}_n of noise data on the 102 Vectorview magnetometers was computed (a) before and (b) after the application of SSP with three-dimensional noise subspace. The plotted quantity is $\sqrt{|(\mathbf{C}_n)_{jk}|}$. Note that the vertical scale in (b) is ten times smaller than in (a).

If the first requirement is not satisfied, some noise will leak through because $\mathbf{P}_\perp \mathbf{b}_n(t) \neq 0$. If any of the brain signal vectors $\mathbf{b}_s(t)$ is close to the noise subspace not only the noise but also the signal will be attenuated by the application of \mathbf{P}_\perp and, consequently, there might be little gain in signal-to-noise ratio. Figure 4.15 demonstrates the effect of SSP on the Vectorview magnetometer data. After the elimination of a three-dimensional noise subspace, the absolute value of the noise is dampened approximately by a factor of 10 and the covariance matrix becomes diagonally dominant.

Since the signal-space projection modifies the signal vectors originating in the brain, it is necessary to apply the projection to the forward solution in the course of inverse computations. This is accomplished by *mne_inverse_operator* as described in Section 6.4. For more information on SSP, please consult the references listed in Section 13.4.

4.11.2 Estimation of the noise subspace

As described above, application of SSP requires the estimation of the signal vectors $\mathbf{b}_1 \dots \mathbf{b}_m$ constituting the noise subspace. The most common approach, implemented in *mne_browse_raw* is to compute a covariance matrix of empty room data, compute its eigenvalue decomposition, and employ the eigenvectors corresponding to the highest eigenvalues as the constituents of the noise subspace. It is also customary to use a separate set of vectors for magnetometers and gradiometers in the Vectorview system. As

4.11.3 EEG average electrode reference

In the computation of EEG-based source estimates, the MNE software employs the average-electrode reference, which means that the average over all electrode signals $\mathbf{v}_1 \dots \mathbf{v}_p$ is subtracted from each \mathbf{v}_j :

$$\mathbf{v}'_j = \mathbf{v}_j - \frac{1}{p} \sum_k \mathbf{v}_k.$$

It is easy to see that the above equation actually corresponds to a projection operator

$$\mathbf{v}' = (\mathbf{I} - \mathbf{u}\mathbf{u}^T)\mathbf{v}$$

with

$$\mathbf{u} = \frac{1}{\sqrt{p}} \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T.$$

4.12 Covariance matrix computation

This section describes how the covariance matrices are computed for raw data and epochs.

4.12.1 Continuous raw data

If a covariance matrix of a raw data is computed the data are checked for artefacts in 200-sample pieces. Let us denote the accepted samples for all channel by the vectors $\mathbf{s}_j, j = 1 \dots M$. The estimate of the covariance matrix is then computed as:

$$\hat{\mathbf{C}} = \frac{1}{M-1} \sum_{j=1}^M (\mathbf{s}_j - \bar{\mathbf{s}})(\mathbf{s}_j - \bar{\mathbf{s}})^T$$

where

$$\bar{\mathbf{s}} = \frac{1}{M} \sum_{j=1}^M \mathbf{s}_j$$

is the average of the signals over all times. Note that no attempt is made to correct for low frequency drifts in the data. If the contribution of any frequency band is not desired in the covariance matrix estimate, suitable bandpass filter should be applied.

For actual computations, it is convenient to rewrite the expression for the covariance matrix as

$$\hat{\mathbf{C}} = \frac{1}{M-1} \sum_{j=1}^M \mathbf{s}_j \mathbf{s}_j^T - \frac{M}{M-1} \bar{\mathbf{s}} \bar{\mathbf{s}}^T$$

4.12.2 Epochs

The calculation of the covariance matrix is slightly more complicated in the epoch mode. In this case, baseline correction is first applied to each epoch. Instead of calculating the mean over all time points, the mean at each time point in the epoch is subtracted from the individual epochs. Let,

$$\mathbf{s}_{pj}^{(r)}, p = 1 \dots P_r, j = 1 \dots N_r, r = 1 \dots R$$

be the (baseline corrected) epochs used to calculate the covariance matrix. In the above, P_r is the number of accepted epochs in category r , N_r is the number of samples in the epochs of category r , and R is the number of categories. For this case we estimate the covariance matrix as

$$\hat{\mathbf{C}} = \frac{1}{N_c} \sum_{r=1}^R \sum_{j=1}^{N_r} \sum_{p=1}^{P_r} \left(\mathbf{s}_{pj}^{(r)} - \bar{\mathbf{s}}_j^{(r)} \right) \left(\mathbf{s}_{pj}^{(r)} - \bar{\mathbf{s}}_j^{(r)} \right)^T,$$

where

$$\bar{\mathbf{s}}_j^{(r)} = \frac{1}{P_r} \sum_{p=1}^{P_r} \mathbf{s}_{pj}^{(r)}$$

and

$$N_c = \sum_{r=1}^R N_r (P_r - 1),$$

which reflects the fact that N_r means are computed for category r . Again, the expression for the covariance matrix estimate can be cast into a more convenient form

$$\hat{\mathbf{C}} = \frac{1}{N_c} \sum_{r,j,p} \mathbf{s}_{pj}^{(r)} \mathbf{s}_{pj}^{(r)} - \frac{1}{N_c} \sum_r P_r \sum_j \bar{\mathbf{s}}_j^{(r)} \mathbf{s}_j^{(r)}.$$

4.12.3 Combination of covariance matrix estimates

Let us assume that we have computed multiple covariance matrix estimates $\hat{\mathbf{C}}_1 \dots \hat{\mathbf{C}}_Q$ with corresponding degrees of freedom $N_1 \dots N_Q$. We can combine these matrices together as

$$\mathbf{C} = \sum_q \alpha_q \hat{\mathbf{C}}_q,$$

where

$$\alpha_q = \frac{N_q}{\sum_q N_q}.$$

4.12.4 Projections included with the covariance matrix

If a signal space projection was on when a covariance matrix was calculated, information about the projections applied is included with the covariance matrix when it is saved. These projection data are read by *mne_inverse_operator* and applied to the forward solution as well as appropriate. Inclusion of the projections into the covariance matrix limits the possibilities to use the `--bad` and `--proj` options in *mne_inverse_operator*, see Section 6.4.

CHAPTER 5 The forward solution

5.1 Overview

This Chapter covers the details of the computation of the forward solutions and the associated low-level utilities.

5.2 The head and device coordinate systems

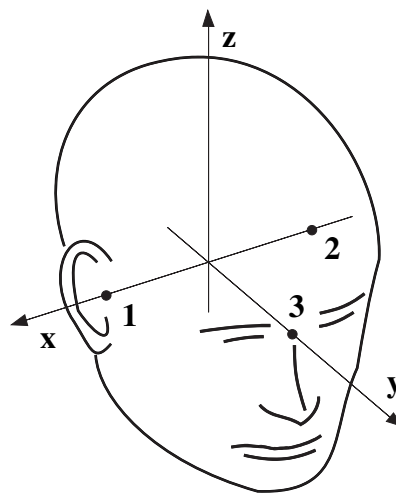


Figure 5.1 The head coordinate system

The MEG/EEG head coordinate system employed in the MNE software is a right-handed Cartesian coordinate system. The direction of x axis is from left to right, that of y axis to the front, and the z axis thus points up.

The x axis of the head coordinate system passes through the two periauricular or preauricular points digitized before acquiring the data with positive direction to the right. The y axis passes through the nasion and is normal to the x axis. The z axis points up according to the right-hand rule and is normal to the xy plane.

The origin of the MEG device coordinate system is device dependent. Its origin is located approximately at the center of a sphere which fits the occipital section of the MEG helmet best with x axis going from left to

right and y axis pointing front. The z axis is, again normal to the xy plane with positive direction up.

Important: The above definition is identical to that of the Neuromag head coordinate system. However, in 4-D Neuroimaging and CTF MEG systems the head coordinate frame definition is different. The origin of the coordinate system is at the midpoint of the left and right auricular points. The x axis passes through the nasion and the origin with positive direction to the front. The y axis is perpendicular to the x axis on the and lies in the plane defined by the three fiducial landmarks, positive direction from right to left. The z axis is normal to the plane of the landmarks, pointing up. Note that the auricular points are not necessarily exactly on any of the coordinate axes. The file conversion utilities (see Section 9.2) take care of these idiosyncracies and convert all coordinate information to the MNE software head coordinate frame.

5.3 Creating a source space

The *fif* format source space files containing the dipole locations and orientations are created with the utility *mne_make_source_space*. This utility is usually invoked by the convenience script *mne_setup_source_space*, see Section 3.5.

The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--dip <name1>:<name2>:...

Specifies the dip file names separated by colons. The file can be either in text or binary format. This file specifies the locations and orientations of the dipoles in the source space. This options is provided for compatibility with the older file formats.

--tri <name1>:<name2>:...

tri file names, separated by colons to be used instead of dip files. The format of this file is the same as that of the BEM meshes. However, the triangle vertex ordering must follow the ‘swapped’ convention, i.e., the vector cross product of the vectors from vertex 1 to 2 and 1 to 3 should give the direction of the inward surface normal. No topology check is performed on this surface.

--surf <name1>:<name2>:...

FreeSurfer surface file names, separated by colons.

--dec <name1>:<name2>:...

Specifies the dec file names separated by colons in the same order as the dip files. The file can be either in text or binary format. This file specifies which surface nodes are actually employed as dipole locations in the source space.

--ico <number>

Instead of using the traditional method for cortical surface decimation it is possible to create the source space using the topology of a recursively subdivided icosahedron ($\text{<number>} > 0$) or an octahedron ($\text{<number>} < 0$). This method uses the cortical surface inflated to a sphere as a tool to find the appropriate vertices for the source space. The benefit of the --ico option is that the source space will have triangulation information between the decimated vertices included, which some future versions of MNE software may be able to utilize. The number of triangles increases by a factor of four in each subdivision, starting from 20 triangles in an icosahedron and 8 triangles in an octahedron. Since the number of vertices on a closed surface is $n_{\text{vert}} = (n_{\text{tri}} + 4)/2$, the number of vertices in the k th subdivision of an icosahedron and an octahedron are $10 \cdot 4^k + 2$ and $4^{k+1} + 2$, respectively. The recommended values for <number> and the corresponding number of source space locations are listed in Table 3.1.

--all

Include all nodes to the output. The active dipole nodes are identified in the fif file by a separate tag. If tri files were used as input the output file will also contain information about the surface triangulation. This option is always recommended to include complete information.

--src <name>

Output file name. Use a name $\text{<dir>/<name>-src.fif}$

Important: Only one of --dip, --tri, or --surf options should be specified.

Example:

```
mne_make_source_space \
-dip lh.dip:rh.dip \
-dec ld-10.dec:rh-10.dec \
-all \
-src <dir>/<name>-src.fif
```

5.4 Creating the BEM meshes

The *mne_surf2bem* utility converts surface triangle meshes from ASCII and FreeSurfer binary file formats to the fiff format. The resulting fiff file also contains conductivity information so that it can be employed in the BEM calculations.

Note: The utility *mne_tri2fiff* previously used for this task has been replaced by *mne_surf2bem*.

Tip: The convenience script *mne_setup_forward_model* described in Section 3.7 calls *mne_surf2bem* with the appropriate options.

Note: The vertices of all surfaces should be given in the MRI coordinate system.

5.4.1 Command-line options

This program has the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--surf <name>

Specifies a FreeSurfer binary format surface file. Before specifying the next surface (**--surf** or **--tri** options) details of the surface specification can be given with the options listed in Section 5.4.2.

--tri <name>

Specifies a text format surface file. Before specifying the next surface (**--surf** or **--tri** options) details of the surface specification can be given with the options listed in Section 5.4.2. The format of these files is described in Section 5.4.3.

--check

Check that the surfaces are complete and that they do not intersect. This is a recommended option. For more information, see Section 5.4.4.

--checkmore

In addition to the checks implied by the **--check** option, check skull and skull thicknesses. For more information, see Section 5.4.4.

--fif <name>

The output fif file containing the BEM. These files normally reside in the `bem` subdirectory under the subject's mri data. A name ending with `-bem.fif` is recommended.

5.4.2 Surface options

These options can be specified after each `--surf` or `--tri` option to define details for the corresponding surface.

--swap

Swap the ordering of the triangle vertices. The standard convention in the MNE software is to have the vertices ordered so that the vector cross product of the vectors from vertex 1 to 2 and 1 to 3 gives the direction of the outward surface normal. Text format triangle files produced by the FreeSurfer software have an opposite order. For these files, the `--swap` option is required.

--sigma <value>

The conductivity of the compartment inside this surface in S/m.

--shift <value/mm>

Shift the vertices of this surface by this amount, given in mm, in the outward direction, *i.e.*, in the positive vertex normal direction.

--meters

The vertex coordinates of this surface are given in meters instead of millimeters. This option applies to text format files only. This definition does not affect the units of the shift option.

--id <number>

Identification number to assign to this surface. (1 = inner skull, 2 = outer skull, 4 = scalp).

--ico <number>

Downsample the surface to the designated subdivision of an icosahedron. This option is relevant (and required) only if the triangulation is isomorphic with a recursively subdivided icosahedron. For example, the surfaces produced by `mri_watershed` are isomorphic with the 5th subdivision of an icosahedron thus containing 20480 triangles. However, this number of triangles is too large for present computers. Therefore, the triangulations have to be decimated. Specifying `--ico 4` yields 5120 triangles per surface while `--ico 3` results in 1280 triangles. The recommended choice is `--ico 4`.

5.4.3 Tessellation file format

The format of the text format surface files is the following:

```
<nvert>
<vertex 1>
<vertex 2>
...
<vertex nvert>
<ntri>
<triangle 1>
<triangle 2>
...
<triangle ntri>,
```

where *<nvert>* and *<ntri>* are the number of vertices and number of triangles in the tessellation, respectively.

The format of a vertex entry is one of the following:

x y z

The x,y, and z coordinates of the vertex location are given in mm.

number x y z

A running number and the x, y, and z coordinates are given. The running number is not considered by mne_tri2fiff. The nodes must be thus listed in the correct consecutive order.

x y z nx ny nz

The x, y, and z coordinates as well as the approximate vertex normal direction cosines are given.

number x y z nx ny nz

A running number is given in addition to the vertex location and vertex normal.

Each triangle entry consists of the numbers of the vertices belonging to a triangle. The vertex numbering starts from one. The triangle list may also contain running numbers on each line describing a triangle.

5.4.4 Topology checks

If the `--check` option is specified, the following topology checks are performed:

1. The completeness of each surface is confirmed by calculating the total solid angle subtended by all triangles from a point inside the triangulation. The result should be very close to 4π . If the result is -4π instead,

it is conceivable that the ordering of the triangle vertices is incorrect and the `--swap` option should be specified.

2. The correct ordering of the surfaces is verified by checking that the surfaces are inside each other as expected. This is accomplished by checking that the sum solid angles subtended by triangles of a surface S_k at all vertices of another surface S_p which is supposed to be inside it equals 4π . Naturally, this check is applied only if the model has more than one surface. Since the surface relations are transitive, it is enough to check that the outer skull surface is inside the skin surface and that the inner skull surface is inside the outer skull one.
3. The extent of each of the triangulated volumes is checked. If the extent is smaller than 50 mm, an error is reported. This may indicate that the vertex coordinates have been specified in meters instead of millimeters.

5.4.5 Examples

Example: three layer BEM model

```
mne_surf2bem \
-head outer_skin642.tri \
-skull outer_skull1642.tri \
-brain inner_skull1642.tri \
-scalpc 0.3 \
-skullc 0.00375 \
-brainc 0.3 \
-coord mri \
-out <dir>/<name>-bem.fif
```

Example: homogeneous model (isolating skull):

```
mne_surf2bem \
-brain inner_skull1642.tri \
-brainc 0.3 \
-coord mri \
-out <dir>/<name>-bem.fif
```

5.4.6 Importing tessellations from BrainSuite

The BrainSuite software stores the tessellation data in a format called *dfs*. These files can be converted to *FreeSurfer* and *fif* formats with the utility *mne_convert_dfs*, which accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--dfs <name>

Specifies the name of a dfs file to be converted.

--trans <name>

Specifies the name of a text file which contains the coordinate transformation to be applied to the data read from the dfs file to bring it to the *FreeSurfer* MRI coordinates, see below.

--surf <name>

Specifies the name of an output file which will contain the surface in FreeSurfer format.

--fif <name>

Specifies the name of an output file which will contain the surface in fif format.

The coordinate transformation file specified with the **--trans** should contain a 4 x 5 coordinate transformation matrix:

$$\mathbf{T} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & x_0 \\ R_{13} & R_{13} & R_{13} & y_0 \\ R_{13} & R_{13} & R_{13} & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

defined so that if the augmented location vectors in the dfs file and MRI coordinate systems are denoted by $\mathbf{r}_{\text{dfs}} = \begin{bmatrix} x_{\text{dfs}} & y_{\text{dfs}} & z_{\text{dfs}} & 1 \end{bmatrix}$ and $\mathbf{r}_{\text{MRI}} = \begin{bmatrix} x_{\text{MRI}} & y_{\text{MRI}} & z_{\text{MRI}} & 1 \end{bmatrix}$, respectively,

$$\mathbf{r}_{\text{MRI}} = \mathbf{T} \mathbf{r}_{\text{dfs}}$$

5.5 Computing the BEM geometry data

The utility *mne_prepare_bem_model* computes the geometry information for BEM. This utility is usually invoked by the convenience script *mne_setup_forward_model*, see Section 3.7. The command-line options are:

--bem <name>

Specify the name of the file containing the triangulations of the BEM surfaces and the conductivities of the compartments. The standard ending for this file is **-bem.fif** and it is produced either with the utility *mne_surf2bem* (Section 5.4) or the convenience script *mne_setup_forward_model*, see Section 3.7.

--sol <name>

Specify the name of the file containing the triangulation and conductivity information together with the BEM geometry matrix computed by *mne_prepare_bem_model*. The standard ending for this file is `-bem-sol.fif`.

--constant

Use the constant collocation BEM approximation method. In this approach, the BEM basis functions are constant functions on each triangle and the collocation points are the midpoints of the triangles.

--linear

Use the linear collocation BEM approximation method. In this approach, the BEM basis functions are linear functions on each triangle and the collocation points are the vertices of the triangulation. This is the preferred method to use. The accuracy will be the same or better than in the constant collocation approach with about half the number of unknowns in the BEM equations.

5.6 Coil geometry information

This Section explains the presentation of MEG detection coil geometry information the approximations used for different detection coils in MNE software. Two pieces of information are needed to characterize the detectors:

1. The location and orientation a local coordinate system for each detector.
2. A unique identifier, which has an one-to-one correspondence to the geometrical description of the coil.

5.6.1 The coil coordinate system

The coil coordinate system is completely characterized by the location of its origin and the direction cosines of three orthogonal unit vectors pointing to the directions of the x , y , and z axis. In fact, the unit vectors contain redundant information because the orientation can be uniquely defined with three angles. The measurement fif files list these data in MEG device coordinates. Transformation to the MEG head coordinate frame can be easily accomplished by applying the device-to-head coordinate transformation matrix available in the data files provided that the head-position indicator was used. Optionally, the MNE software forward calculation applies another coordinate transformation to the head-coordinate data to bring the coil locations and orientations to the MRI coordinate system.

If \mathbf{r}_0 is a row vector for the origin of the local coil coordinate system and \mathbf{e}_x , \mathbf{e}_y , and \mathbf{e}_z are the row vectors for the three orthogonal unit vectors,

all given in device coordinates, a location of a point \mathbf{r}_C in coil coordinates is transformed to device coordinates (\mathbf{r}_D) by

$$\begin{bmatrix} \mathbf{r}_D & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_C & 1 \end{bmatrix} \mathbf{T}_{CD},$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{e}_x & 0 \\ \mathbf{e}_y & 0 \\ \mathbf{e}_z & 0 \\ \mathbf{r}_{0D} & 1 \end{bmatrix}.$$

5.6.2 Calculation of the magnetic field

The forward calculation in the MNE software computes the signals detected by each MEG sensor for three orthogonal dipoles at each source space location. This requires specification of the conductor model, the location and orientation of the dipoles, and the location and orientation of each MEG sensor as well as its coil geometry.

The output of each SQUID sensor is a weighted sum of the magnetic fluxes threading the loops comprising the detection coil. Since the flux threading a coil loop is an integral of the magnetic field component normal to the coil plane, the output of the k th MEG channel, b_k , can be approximated by:

$$b_k = \sum_{p=1}^{N_k} w_{kp} \mathbf{B}(\mathbf{r}_{kp}) \cdot \mathbf{n}_{kp}$$

where \mathbf{r}_{kp} are a set of N_k integration points covering the pickup coil loops of the sensor, $\mathbf{B}(\mathbf{r}_{kp})$ is the magnetic field due to the current sources calculated at \mathbf{r}_{kp} , \mathbf{n}_{kp} are the coil normal directions at these points, and w_{kp} are the weights associated to the integration points. This formula essentially presents numerical integration of the magnetic field over the pickup loops of sensor k .

There are three accuracy levels for the numerical integration expressed above. The *simple* accuracy means the simplest description of the coil. This accuracy is not used in the MNE forward calculations. The *normal* or *recommended* accuracy typically uses two integration points for planar gradiometers, one in each half of the pickup coil and four evenly distrib-

uted integration points for magnetometers. This is the default accuracy used by MNE. If the `--accurate` option is specified, the forward calculation typically employs a total of eight integration points for planar gradiometers and sixteen for magnetometers. Detailed information about the integration points is given in the next section.

5.6.3 Implemented coil geometries

This section describes the coil geometries currently implemented in Neuromag software. The coil types fall in two general categories:

1. Axial gradiometers and planar gradiometers and
2. Planar gradiometers.

For axial sensors, the z axis of the local coordinate system is parallel to the field component detected, *i.e.*, normal to the coil plane. For circular coils, the orientation of the x and y axes on the plane normal to the z axis is irrelevant. In the square coils employed in the Vectorview™ system the x axis is chosen to be parallel to one of the sides of the magnetometer coil. For planar sensors, the z axis is likewise normal to the coil plane and the x axis passes through the centerpoints of the two coil loops so that the detector gives a positive signal when the normal field component increases along the x axis.

Table 5.1 lists the parameters of the *normal* coil geometry descriptions Table 5.2 lists the *accurate* descriptions. For *simple* accuracy, please consult the coil definition file, see Section 5.6.4. The columns of the tables contain the following data:

1. The number identifying the coil id. This number is used in the coil descriptions found in the FIF files.
2. Description of the coil.
3. Number of integration points used
4. The locations of the integration points in coil coordinates.
5. Weights assigned to the field values at the integration points. Some formulas are listed instead of the numerical values to demonstrate the principle of the calculation. For example, in the normal coil descriptions of the planar gradiometers the weights are inverses of the baseline of the gradiometer to show that the output is in T/m.

Note: The coil geometry information is stored in the file `$MNE_ROOT/setup/mne/coil_def.dat`, which is automatically created by the utility

mne_list_coil_def when the MNE software is initially configured, see Section 5.6.5.

Id	Description	n	r/mm	w
2	Neuromag-122 planar gradiometer	2	$\pm(8.1, 0, 0)$ mm	$\pm 1/16.2$ mm
2000	A point magnetometer	1	$(0, 0, 0)$ mm	1
3012	Vectorview type 1 planar gradiometer	2	$(\pm 8.4, 0, 0.3)$ mm	$\pm 1/16.8$ mm
3013	Vectorview type 2 planar gradiometer	2	$(\pm 8.4, 0, 0.3)$ mm	$\pm 1/16.8$ mm
3022	Vectorview type 1 magnetometer	4	$(\pm 6.45, \pm 6.45, 0.3)$ mm	1/4
3023	Vectorview type 2 magnetometer	4	$(\pm 6.45, \pm 6.45, 0.3)$ mm	1/4
3024	Vectorview type 3 magnetometer	4	$(\pm 5.25, \pm 5.25, 0.3)$ mm	1/4
2000	An ideal point magnetometer	1	$(0,0,0)$	1
4001	Magnes WH magnetometer	4	$(\pm 5.75, \pm 5.75, 0.0)$ mm	1/4
4002	Magnes WH 3600 axial gradiometer	8	$(\pm 4.5, \pm 4.5, 0.0)$ mm $(\pm 4.5, \pm 4.5, 50.0)$ mm	1/4 -1/4
4003	Magnes reference magnetometer	4	$(\pm 7.5, \pm 7.5, 0)$ mm	1/4
4004	Magnes reference gradiometer measuring diagonal gradients	8	$(\pm 20, \pm 20, 0.0)$ mm $(\pm 20, \pm 20, 135)$ mm	1/4 -1/4
4005	Magnes reference gradiometer measuring off-diagonal gradients	8	$(87.5, \pm 20, 0.0)$ mm $(47.5, \pm 20, 0.0)$ mm $(-87.5, \pm 20, 0.0)$ mm $(-47.5, \pm 20, 0.0)$ mm	1/4 -1/4 1/4 -1/4
5001	CTF 275 axial gradiometer	8	$(\pm 4.5, \pm 4.5, 0.0)$ mm $(\pm 4.5, \pm 4.5, 50.0)$ mm	1/4 -1/4

Table 5.1 Normal coil descriptions. Note: If a plus-minus sign occurs in several coordinates, all possible combinations have to be included.

Id	Description	n	r/mm	w
5002	CTF reference magnetometer	4	$(\pm 4, \pm 4, 0)$ mm	1/4
5003	CTF reference gradiometer measuring diagonal gradients	8	$(\pm 8.6, \pm 8.6, 0.0)$ mm $(\pm 8.6, \pm 8.6, 78.6)$ mm	1/4 -1/4
5004	CTF reference gradiometer measuring off-diagonal gradients	8	$(47.8, \pm 8.5, 0.0)$ mm $(30.8, \pm 8.5, 0.0)$ mm $(-47.8, \pm 8.5, 0.0)$ mm $(-30.8, \pm 8.5, 0.0)$ mm	1/4 -1/4 1/4 -1/4
6001	MIT KIT system axial gradiometer	8	$(\pm 3.875, \pm 3.875, 0.0)$ mm $(\pm 3.875, \pm 3.875, 50.0)$ mm	1/4 -1/4

Table 5.1 Normal coil descriptions. Note: If a plus-minus sign occurs in several coordinates, all possible combinations have to be included.

Id	Description	n	r/mm	w
2	Neuromag-122 planar gradiometer	8	$(\pm 5.44, \pm 7.68, 0)$ mm , $(\pm 11.1, \pm 7.68, 0)$ mm	$\pm 1/(4 \cdot 16.54\text{mm})$
2000	A point magnetometer	1	$(0, 0, 0)$ mm	1
3012	Vectorview type 1 planar gradiometer	8	$(\pm 5.89, \pm 6.71, 0.3)$ mm , $(\pm 10.8, \pm 6.71, 0.3)$ mm	$\pm 1/(4 \cdot 16.69\text{mm})$
3013	Vectorview type 2 planar gradiometer	8	$(\pm 5.89, \pm 6.71, 0.3)$ mm , $(\pm 10.8, \pm 6.71, 0.3)$ mm	$\pm 1/(4 \cdot 16.69\text{mm})$
3022	Vectorview type 1 magnetometer	16	$(\pm 9.68, \pm 9.68, 0.3)$ mm , $(\pm 9.68, \pm 3.23, 0.3)$ mm $(\pm 3.23, \pm 9.68, 0.3)$ mm $(\pm 3.23, \pm 3.23, 0.3)$ mm	1/16
3023	Vectorview type 2 magnetometer	16	$(\pm 9.68, \pm 9.68, 0.3)$ mm , $(\pm 9.68, \pm 3.23, 0.3)$ mm $(\pm 3.23, \pm 9.68, 0.3)$ mm $(\pm 3.23, \pm 3.23, 0.3)$ mm	1/16

Table 5.2 Accurate coil descriptions

Id	Description	n	r/mm	w
3023	Vectorview type 3 magnetometer	16	($\pm 7.88, \pm 7.88, 0.3$)mm , ($\pm 7.88, \pm 2.63, 0.3$)mm ($\pm 2.63, \pm 7.88, 0.3$)mm ($\pm 2.63, \pm 2.63, 0.3$)mm	1/16
4001	Magnes WH magnetometer	7	(0.0, 0.0, 0.0)mm ($\pm 9.390, 0.0, 0.0$)mm ($\pm 4.695, \pm 8.132, 0.0$)mm	1/4 1/8 1/8
4002	Magnes WH 3600 axial gradiometer	14	(0.0, 0.0, 0.0)mm ($\pm 7.348, 0.0, 0.0$)mm ($\pm 3.674, \pm 6.364, 0.0$)mm (0.0, 0.0, 50.0)mm ($\pm 7.348, 0.0, 50.0$)mm ($\pm 3.674, \pm 6.364, 50.0$)mm	1/4 1/8 1/8 -1/4 -1/8 -1/8
4004	Magnes reference gradiometer measuring diagonal gradients	8	($\pm 20, \pm 20, 0.0$)mm ($\pm 20, \pm 20, 135$)mm	1/4 -1/4
4005	Magnes reference gradiometer measuring off-diagonal gradients	8	($87.5, \pm 20, 0.0$)mm ($47.5, \pm 20, 0.0$)mm ($-87.5, \pm 20, 0.0$)mm ($-47.5, \pm 20, 0.0$)mm	1/4 -1/4 1/4 -1/4
4004	Magnes reference gradiometer measuring diagonal gradients	8	($\pm 20, \pm 20, 0.0$)mm ($\pm 20, \pm 20, 135$)mm	1/4 -1/4
5001	CTF 275 axial gradiometer	14	(0.0, 0.0, 0.0)mm ($\pm 7.348, 0.0, 0.0$)mm ($\pm 3.674, \pm 6.364, 0.0$)mm (0.0, 0.0, 50.0)mm ($\pm 7.348, 0.0, 50.0$)mm ($\pm 3.674, \pm 6.364, 50.0$)mm	1/4 1/8 1/8 -1/4 -1/8 -1/8
5002	CTF reference magnetometer	4	($\pm 4, \pm 4, 0$)mm	1/4
5003	CTF reference gradiometer measuring diagonal gradients	8	($\pm 8.6, \pm 8.6, 0.0$)mm ($\pm 8.6, \pm 8.6, 78.6$)mm	1/4 -1/4

Table 5.2 Accurate coil descriptions

Id	Description	n	r/mm	w
5004	CTF reference gradiometer measuring off-diagonal gradients	8	(47.8, ± 8.5 , 0.0)mm (30.8, ± 8.5 , 0.0)mm (-47.8, ± 8.5 , 0.0)mm (-30.8, ± 8.5 , 0.0)mm	1/4 -1/4 1/4 -1/4
6001	MIT KIT system axial gradiometer	14	(0.0, 0.0, 0.0)mm (± 6.328 , 0.0, 0.0)mm (± 3.164 , ± 5.48 , 0.0)mm (0.0, 0.0, 50.0)mm (± 6.328 , 0.0, 50.0)mm (± 3.164 , ± 5.48 , 50.0)mm	1/4 1/8 1/8 -1/4 -1/8 -1/8

Table 5.2 Accurate coil descriptions

5.6.4 The coil definition file

The coil geometry information is stored in the text file \$MNE_ROOT/setup/mne/coil_def.dat. In this file, any lines starting with the pound sign (#) are comments. A coil definition starts with a description line containing the following fields:

<class>

This is a number indicating class of this coil. Possible values are listed in Table 5.3.

<id>

Coil id value. This value is listed in the first column of Tables 5.1 and 5.2.

<accuracy>

The coil representation accuracy. Possible values and their meanings are listed in Table 5.4.

<np>

Number of integration points in this representation.

<size/m>

The size of the coil. For circular coils this is the diameter of the coil and for square ones the side length of the square. This information is mainly included to facilitate drawing of the coil geometry. It should not be employed to infer a coil approximation for the forward calculations.

<baseline/m>

The baseline of a this kind of a coil. This will be zero for magnetometer coils. This information is mainly included to facilitate drawing of the coil geometry. It should not be employed to infer a coil approximation for the forward calculations.

<description>

Short description of this kind of a coil. If the description contains several words, it is enclosed in quotes.

Value	Meaning
1	magnetometer
2	first-order axial gradiometer
3	planar gradiometer
4	second-order axial gradiometer
1000	an EEG electrode (used internally in software only).

Table 5.3 Coil class values

Value	Meaning
1	The simplest representation available
2	The standard or <i>normal</i> representation (see Table 5.1)
3	The most <i>accurate</i> representation available (see Table 5.2)

Table 5.4 Coil representation accuracies.

Each coil description line is followed by one or more integration point lines, consisting of seven numbers:

<weight>

Gives the weight for this integration point (last column in Tables 5.1 and 5.2).

<x/m> <y/m> <z/m>

Indicates the location of the integration point (fourth column in Tables 5.1 and 5.2).

<nx> <ny> <nz>

Components of a unit vector indicating the field component to be selected. Note that listing a separate unit vector for each integration

points allows the implementation of curved coils and coils with the gradiometer loops tilted with respect to each other.

5.6.5 Creating the coil definition file

The standard coil definition file `$MNE_ROOT/setup/mne/coil_def.dat` is automatically created during MNE software configuration (see Section C.2.2). The configuration script uses the utility `mne_list_coil_def`, which produces a the text of the coil definition file to the standard output. Because this utility performs a software administration function, it is located in `$MNE_ROOT/bin/admin` and it is normally not on the search path.

5.7 Computing the forward solution

5.7.1 Purpose

Instead of using the convenience script `mne_do_forward_solution` it is also possible to invoke the forward solution computation program `mne_forward_solution` directly. In this approach, the convenience of the automatic file naming conventions present in `mne_do_forward_solution` are lost. However, there are some special-purpose options available in `mne_forward_solution` only. Please refer to Section 3.11 for information on `mne_do_forward_solution`.

5.7.2 Command line options

`mne_forward_solution` accepts the following command-line options:

--src <name>

Source space name to use. The name of the file must be specified exactly, including the directory. Typically, the source space files reside in `$SUBJECTS_DIR/$SUBJECT/bem`.

--bem <name>

Specifies the BEM to be used. These files end with `bem.fif` or `bem-sol.fif` and reside in `$SUBJECTS_DIR/$SUBJECT/bem`. The former file contains only the BEM surface information while the latter files contain the geometry information precomputed with `mne_prepare_bem_model`, see Section 5.5. If precomputed geometry is not available, the linear collocation solution will be computed by `mne_forward_solution`.

--origin <x/mm>:<y/mm>:<z/mm>

Indicates that the sphere model should be used in the forward calculations. The origin is specified in MEG head coordinates unless the

`--mricoord` option is present. The MEG sphere model solution computed using the analytical Sarvas formula. For EEG, an approximative solution described in

`--eegmodels <name>`

This option is significant only if the sphere model is used and EEG channels are present. The specified file contains specifications of the EEG sphere model layer structures as detailed in Section 5.7.4.

`--eegmodel <model name>`

Specify the name of the sphere model to be used for EEG. If this option is missing, the model *Default* will be employed, see Section 5.7.4.

`--eegscalp`

Scale the EEG electrode locations to the surface of the outermost sphere when using the sphere model.

`--accurate`

Use accurate MEG sensor coil descriptions. This is the recommended choice. More information

`--fixed`

Compute the solution for sources normal to the cortical mantle only.

`--all`

Compute the forward solution for all vertices on the source space surface.

`--label <name>`

Compute the solution only for points within the specified label. Multiple labels can be present. The label files should end with `-lh.label` or `-rh.label` for left and right hemisphere label files, respectively. If `--all` flag is present, all surface points falling within the labels are included. Otherwise, only decimated points within the label are selected.

`--mindist <dist/mm>`

Omit source space points closer than this value to the inner skull surface. Any source space points outside the inner skull surface are automatically omitted. The use of this option ensures that numerical inaccuracies for very superficial sources do not cause unexpected effects in the final current estimates. Suitable value for this parameter is of the order of the size of the triangles on the inner skull surface. If you employ the seglab software to create the triangulations, this value should be about equal to the wish for the side length of the triangles.

--mindistout <name>

Specifies a file name to contain the coordinates of source space points omitted due to the `--mindist` option.

--mri <name>

The name of the MRI description file containing the MEG/MRI coordinate transformation. This file was saved as part of the alignment procedure outlined in Section 3.10. These files typically reside in `$SUBJECTS_DIR/$SUBJECT/mri/T1-neuromag/sets`.

--trans <name>

The name of a text file containing the 4 x 4 matrix for the coordinate transformation from head to mri coordinates. This matrix was used in the traditional MNE processing stream. With `--trans`, `--mri` option is not required.

--mricoord

Do all computations in the MRI coordinate system.

--meas <name>

This file is the measurement fif file or an off-line average file produced thereof. It is recommended that the average file is employed for evoked-response data and the original raw data file otherwise. This file provides the MEG sensor locations and orientations as well as EEG electrode locations as well as the coordinate transformation between the MEG device coordinates and MEG head-based coordinates.

--fwd <name>

This file will contain the forward solution as well as the coordinate transformations, sensor and electrode location information, and the source space data. A name of the form `<name>-fwd.fif` is recommended.

--meg

Compute the MEG forward solution.

--eeg

Compute the EEG forward solution.

5.7.3 Implementation of software gradient compensation

As described in Section 9.2.4 the CTF and 4D Neuroimaging data may have been subjected to noise cancellation employing the data from the reference sensor array. Even though these sensor are rather far away from the brain sources, `mne_forward_solution` takes them into account in the computations. If the data file specified with the `--meas` option has software gradient compensation activated, `mne_forward_solution` computes the field of at the reference sensors in addition to the main MEG sensor array

and computes a compensated forward solution using the methods described in Section 9.2.4.

Warning: If a data file specified with the `--meas` option and that used in the actual inverse computations with `mne_analyze` and `mne_make_movie` have different software gradient compensation states., the forward solution will be in mismatch with the data to be analyzed and the current estimates will be slightly erroneous.

5.7.4 The EEG sphere model definition file

For the computation of the electric potential distribution on the surface of the head (EEG) it is necessary to define the conductivities (σ) and radii of the spherically symmetric layers. Different sphere models can be specified with the `--eegmodels` option.

The EEG sphere model definition files may contain comment lines starting with a `#` and model definition lines in the following format:

```
<name>:<radius1>:<conductivity1>:<radius2>:<conductivity2>:...
```

When the file is loaded the layers are sorted so that the radii will be in ascending order and the radius of the outermost layer is scaled to 1.0. The scalp radius specified with the `--eegrad` option is then consulted to scale the model to the correct dimensions. Even if the model setup files are not present, a model called *Default* is always provided. This model has the structure given in Table 5.5

Layer	Relative outer radius	σ (S/m)
Head	1.0	0.33
Skull	0.97	0.04
CSF	0.92	1.0
Brain	0.90	0.33

Table 5.5 Structure of the default EEG model

5.7.5 EEG forward solution in the sphere model

When the sphere model is employed, the computation of the EEG solution can be substantially accelerated by using approximation methods described by Mosher, Zhang, and Berg, see Section 13.3 (Mosher *et al.* and references therein). `mne_forward_solution` approximates the solution

with three dipoles in a homogeneous sphere whose locations and amplitudes are determined by minimizing the cost function:

$$S(r_1, \dots, r_m, \mu_1, \dots, \mu_m) = \int_{\text{scalp}} (V_{\text{true}} - V_{\text{approx}}) dS$$

where r_1, \dots, r_m and μ_1, \dots, μ_m are the locations and amplitudes of the approximating dipoles and V_{true} and V_{approx} are the potential distributions given by the true and approximative formulas, respectively. It can be shown that this integral can be expressed in closed form using an expansion of the potentials in spherical harmonics. The formula is evaluated for the most superficial dipoles, i.e., those lying just inside the inner skull surface.

CHAPTER 6 The current estimates

6.1 Overview

This Chapter describes the computation of the minimum-norm estimates. This is accomplished with two programs: *mne_inverse_operator* and *mne_make_movie*. The chapter starts with a mathematical description of the method, followed by description of the two software modules. The interactive program for inspecting data and inverse solutions, *mne_analyze*, is covered in Chapter 7.

6.2 Minimum-norm estimates

This section describes the mathematical details of the calculation of minimum-norm estimates. In Bayesian sense, the ensuing current distribution is the maximum a posteriori (MAP) estimate under the following assumptions:

1. The viable locations of the currents are constrained to the cortex. Optionally, the current orientations can be fixed to be normal to the cortical mantle.
2. The amplitudes of the currents have a Gaussian a priori distribution with a known source covariance matrix.
3. The measured data contain additive noise with a Gaussian distribution with a known covariance matrix. The noise is not correlated over time.

6.2.1 The linear inverse operator

The measured data in the source estimation procedure consists of MEG and EEG data, recorded on a total of N channels. The task is to estimate a total of M strengths of sources located on the cortical mantle. If the number of source locations is P , $M = P$ for fixed-orientation sources and $M = 3P$ if the source orientations are unconstrained. The regularized linear inverse operator following from the Bayesian approach is given by the $M \times N$ matrix

$$\mathbf{M} = \mathbf{R}'\mathbf{G}^T(\mathbf{G}\mathbf{R}'\mathbf{G}^T + \mathbf{C})^{-1},$$

where \mathbf{G} is the gain matrix relating the source strenghts to the measured MEG/EEG data, \mathbf{C} is the data noise-covariance matrix and \mathbf{R}' is the source covariance matrix. The dimensions of these matrices are $N \times M$, $N \times N$, and $M \times M$, respectively. The $M \times 1$ source-strength vector is obtained by multiplying the $N \times 1$ data vector by \mathbf{M} .

The expected value of the current amplitudes at time t is then given by $\hat{\mathbf{j}}(t) = \mathbf{M}\mathbf{x}(t)$, where $\mathbf{x}(t)$ is a vector containing the measured MEG and EEG data values at time t .

6.2.2 Regularization

The a priori variance of the currents is, in practise, unknown. We can express this by writing $\mathbf{R}' = \mathbf{R}/\lambda^2$, which yields the inverse operator

$$\mathbf{M} = \mathbf{R}\mathbf{G}^T(\mathbf{G}\mathbf{R}\mathbf{G}^T + \lambda^2\mathbf{C})^{-1},$$

where the unknown current amplitude is now interpreted in terms of the regularization parameter λ^2 . Small λ^2 corresponds to large current amplitudes and complex estimate current patterns while a large λ^2 means the amplitude of the current is limited and a simpler, smooth, current estimate is obtained.

We can arrive in the regularized linear inverse operator also by minimizing the cost function

$$S = \tilde{\mathbf{e}}^T \tilde{\mathbf{e}} + \lambda^2 \hat{\mathbf{j}}^T \mathbf{R}^{-1} \hat{\mathbf{j}},$$

where the first term consists of the difference between the whitened measured data (see Section 6.2.3) and those predicted by the model while the second term is a weighted-norm of the current estimate. It is seen that, with increasing λ^2 , the source term receive more weight and larger discrepancy between the measured and predicted data is tolerable.

6.2.3 Whitening and scaling

The MNE software employs data whitening so that a ‘whitened’ inverse operator assumes the form

$$\tilde{\mathbf{M}} = \mathbf{R}\tilde{\mathbf{G}}^T(\tilde{\mathbf{G}}\mathbf{R}\tilde{\mathbf{G}}^T + \mathbf{I})^{-1},$$

where $\tilde{\mathbf{G}} = \mathbf{C}^{-1/2}\mathbf{G}$ is the spatially whitened gain matrix. The expected current values are $\hat{\mathbf{j}}(t) = \tilde{\mathbf{M}}\mathbf{x}(t)$, where $\mathbf{x}(t) = \mathbf{C}^{-1/2}\mathbf{x}(t)$ is a the whitened measurement vector at t . The spatial whitening operator is obtained

with the help of the eigenvalue decomposition $\mathbf{C} = \mathbf{U}_C \Lambda_C^2 \mathbf{U}_C^T$ as $\mathbf{C}^{-1/2} = \Lambda_C^{-1} \mathbf{U}_C^T$. In the MNE software the noise-covariance matrix is stored as the one applying to raw data. To reflect the decrease of noise due to averaging, this matrix, \mathbf{C}_0 , is scaled by the number of averages, L , *i.e.*, $\mathbf{C} = \mathbf{C}_0/L$.

As shown above, regularization of the inverse solution is equivalent to a change in the variance of the current amplitudes in the Bayesian *a priori* distribution.

Convenient choice for the source-covariance matrix \mathbf{R} is such that $\text{trace}(\mathbf{G}\mathbf{R}\mathbf{G}^T)/\text{trace}(\mathbf{I}) = 1$. With this choice we can approximate $\lambda^2 \sim 1/\text{SNR}$, where SNR is the (power) signal-to-noise ratio of the whitened data.

Note: The definition of the signal to noise-ratio/ λ^2 relationship given above works nicely for the whitened forward solution. In the unwhitened case scaling with the trace ratio $\text{trace}(\mathbf{G}\mathbf{R}\mathbf{G}^T)/\text{trace}(\mathbf{C})$ does not make sense, since the diagonal elements summed have, in general, different units of measure. For example, the MEG data are expressed in T or T/m whereas the unit of EEG is Volts.

6.2.4 Regularization of the noise-covariance matrix

Since finite amount of data is usually available to compute an estimate of the noise-covariance matrix \mathbf{C} , the smallest eigenvalues of its estimate are usually inaccurate and smaller than the true eigenvalues. Depending on the seriousness of this problem, the following quantities can be affected:

1. The model data predicted by the current estimate,
2. Estimates of signal-to-noise ratios, which lead to estimates of the required regularization, see Section 6.2.2,
3. The estimated current values, and
4. The noise-normalized estimates, see Section 6.2.6.

Fortunately, the latter two are least likely to be affected due to regularization of the estimates. However, in some cases especially the EEG part of the noise-covariance matrix estimate can be deficient, *i.e.*, it may possess very small eigenvalues and thus regularization of the noise-covariance matrix is advisable.

The MNE software accomplishes the regularization by replacing a noise-covariance matrix estimate \mathbf{C} with

$$\mathbf{C}' = \mathbf{C} + \sum_k \varepsilon_k \bar{\sigma}_k^2 \mathbf{I}^{(k)},$$

where the index k goes across the different channel groups (MEG planar gradiometers, MEG axial gradiometers and magnetometers, and EEG), ϵ_k are the corresponding regularization factors, $\bar{\sigma}_k^2$ are the average variances across the channel groups, and $\mathbf{I}^{(k)}$ are diagonal matrices containing ones at the positions corresponding to the channels contained in each channel group. The values ϵ_k can be adjusted with the regularization options `--magreg`, `--gradreg`, and `--eegreg` specified at the time of the inverse operator decomposition, see Section 6.4. The convenience script *mne_do_inverse_solution* has the `--magreg` and `--gradreg` combined to a single option, `--megreg`, see Section 3.13. Suggested range of values for ϵ_k is 0.05...0.2.

6.2.5 Computation of the solution

The most straightforward approach to calculate the MNE is to employ expression for the original or whitened inverse operator directly. However, for computational convenience we prefer to take another route, which employs the singular-value decomposition (SVD) of the matrix

$$\mathbf{A} = \tilde{\mathbf{G}}\mathbf{R}^C = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$$

where the superscript C indicates the Cholesky factorization.

With the above SVD it is easy to show that

$$\tilde{\mathbf{M}} = \mathbf{R}^C \mathbf{V} \mathbf{\Gamma} \mathbf{U}^T$$

where the elements of the diagonal matrix $\mathbf{\Gamma}$ are

$$\gamma_k = \frac{1}{\lambda_k} \frac{\lambda_k^2}{\lambda_k^2 + \lambda^2}.$$

With $\mathbf{w}(t) = \mathbf{U}^T \mathbf{C}^{-1/2} \mathbf{x}(t)$ the expression for the expected current is

$$\hat{\mathbf{j}}(t) = \mathbf{R}^C \mathbf{V} \mathbf{\Gamma} \mathbf{w}(t) = \sum_k \bar{\mathbf{v}}_k \gamma_k w_k(t),$$

where $\bar{\mathbf{v}}_k = \mathbf{R}^C \mathbf{v}_k$, \mathbf{v}_k being the k th column of \mathbf{V} . It is thus seen that the current estimate is a weighted sum of the ‘modified’ eigenleads \mathbf{v}_k .

It is easy to see that $\mathbf{w}(t) \propto \sqrt{L}$. To maintain the relation $\text{trace}(\mathbf{G}\mathbf{R}\mathbf{G})/\text{trace}(\mathbf{I}) = 1$ when L changes we must have $\mathbf{R} \propto 1/L$. With this approach, λ_k is independent of L and, for fixed λ , we see directly that $\hat{\mathbf{j}}(t)$ is independent of L .

6.2.6 Noise normalization

The noise-normalized linear estimates introduced by Dale et al. require division of the expected current amplitude by its variance. Noise normalization serves three purposes:

1. It converts the expected current value into a dimensionless statistical test variable. Thus the resulting time and location dependent values are often referred to as dynamic statistical parameter maps (dSPM).
2. It reduces the location bias of the estimates. In particular, the tendency of the MNE to prefer superficial currents is eliminated.
3. The width of the point-spread function becomes less dependent on the source location on the cortical mantle. The point-spread is defined as the MNE resulting from the signals coming from a point current source (a current dipole) located at a certain point on the cortex.

In practice, noise normalization requires the computation of the diagonal elements of the matrix

$$\mathbf{MCM}^T = \tilde{\mathbf{M}}\tilde{\mathbf{M}}^T.$$

With help of the singular-value decomposition approach we see directly that

$$\tilde{\mathbf{M}}\tilde{\mathbf{M}}^T = \bar{\mathbf{V}}\Gamma^2\bar{\mathbf{V}}^T.$$

Under the conditions expressed at the end of Section 6.2.5, it follows that the t -statistic values associated with fixed-orientation sources) are thus proportional to \sqrt{L} while the F -statistic employed with free-orientation sources is proportional to L , correspondingly.

Note: A section discussing statistical considerations related to the noise normalization procedure will be added to this manual in one of the subsequent releases.

Note: The MNE software usually computes the square roots of the F -statistic to be displayed on the inflated cortical surfaces. These are also proportional to \sqrt{L} .

6.2.7 Predicted data

Under noiseless conditions the SNR is infinite and thus leads to $\lambda^2 = 0$ and the minimum-norm estimate explains the measured data perfectly. Under realistic conditions, however, $\lambda^2 > 0$ and there is a misfit between measured data and those predicted by the MNE. Comparison of the predicted data, here denoted by $\mathbf{x}(t)$, and measured one can give valuable insight on the correctness of the regularization applied.

In the SVD approach we easily find

$$\hat{\mathbf{x}}(t) = \mathbf{G}\hat{\mathbf{j}}(t) = \mathbf{C}^{1/2}\mathbf{U}\mathbf{\Pi}\mathbf{w}(t),$$

where the diagonal matrix $\mathbf{\Pi}$ has elements $\pi_k = \lambda_k \gamma_k$. The predicted data is thus expressed as the weighted sum of the ‘recolored eigenfields’ in $\mathbf{C}^{1/2}\mathbf{U}$.

6.2.8 Cortical patch statistics

If the `--cps` option was used in source space creation (see Section 3.5) or if `mne_add_patch_info` described in Section 11.7 was run manually the source space file will contain for each vertex of the cortical surface the information about the source space point closest to it as well as the distance from the vertex to this source space point. The vertices for which a given source space point is the nearest one define the cortical patch associated with the source space point. Once these data are available, it is straightforward to compute the following cortical patch statistics (CPS) for each source location d :

1. The average over the normals of at the vertices in a patch, $\bar{\mathbf{n}}_d$,
2. The areas of the patches, A_d , and
3. The average deviation of the vertex normals in a patch from their average, σ_d , given in degrees.

6.2.9 The orientation constraints

The principal sources of MEG and EEG signals are generally believed to be postsynaptic currents in the cortical pyramidal neurons. Since the net primary current associated with these microscopic events is oriented normal to the cortical mantle, it is reasonable to use the cortical normal orientation as a constraint in source estimation. In addition to allowing completely free source orientations, the MNE software implements three orientation constraints based of the surface normal data:

1. Source orientation can be rigidly fixed to the surface normal direction (the `--fixed` option). If cortical patch statistics are available the average normal over each patch, $\bar{\mathbf{n}}_d$, are used to define the source orientation. Otherwise, the vertex normal at the source space location is employed.
2. A *location independent or fixed loose orientation constraint* (fLOC) can be employed (the `--loose` option). In this approach, a source coordinate system based on the local surface orientation at the source location is employed. By default, the three columns of the gain matrix \mathbf{G} , associated with a given source location, are the fields of unit dipoles pointing to the directions of the x, y, and z axis of the coordinate system employed in the forward calculation (usually the MEG head coor-

dinate frame). For LOC the orientation is changed so that the first two source components lie in the plane normal to the surface normal at the source location and the third component is aligned with it. Thereafter, the variance of the source components tangential to the cortical surface are reduced by a factor defined by the `--loose` option.

3. A *variable loose orientation constraint* (vLOC) can be employed (the `--loosevar` option). This is similar to fLOC except that the value given with the `--loosevar` option will be multiplied by σ_d , defined above.

6.2.10 Depth weighting

The minimum-norm estimates have a bias towards superficial currents. This tendency can be alleviated by adjusting the source covariance matrix \mathbf{R} to favor deeper source locations. In the depth weighting scheme employed in MNE analyze, the elements of \mathbf{R} corresponding to the p^{th} source location are scaled by a factor

$$f_p = (\mathbf{g}_{1p}^T \mathbf{g}_{1p} + \mathbf{g}_{2p}^T \mathbf{g}_{2p} + \mathbf{g}_{3p}^T \mathbf{g}_{3p})^{-\gamma},$$

where \mathbf{g}_{1p} , \mathbf{g}_{2p} , and \mathbf{g}_{3p} are the three columns of \mathbf{G} corresponding to source location p and γ is the order of the depth weighting, specified with the `--weightexp` option to `mne_inverse_operator`. The maximal amount of depth weighting can be adjusted `--weightlimit` option.

6.2.11 fMRI-biased estimates

The fMRI weighting in MNE software means that the source-covariance matrix is modified to favor areas of significant fMRI activation. For this purpose, the fMRI activation map is thresholded first at the value defined by the `--fmrithresh` option to `mne_do_inverse_operator` or `mne_inverse_operator`. Thereafter, the source-covariance matrix values corresponding to the sites under the threshold are multiplied by f_{off} , set by the `--fmriooff` option.

It turns out that the fMRI bias has a strong influence on the MNE but the noise-normalized estimates are much less affected by it.

6.3 Effective number of averages

It is often the case that the epoch to be analyzed is a linear combination over conditions rather than one of the original averages computed. As stated above, the noise-covariance matrix computed is originally one corresponding to raw data. Therefore, it has to be scaled correctly to corre-

spend to the actual or effective number of epochs in the condition to be analyzed. In general, we have

$$\mathbf{C} = \mathbf{C}_0/L_{\text{eff}}$$

where L_{eff} is the effective number of averages. To calculate L_{eff} for an arbitrary linear combination of conditions

$$y(t) = \sum_{i=1}^n w_i x_i(t)$$

we make use of the the fact that the noise-covariance matrix

$$\mathbf{C}_y = \sum_{i=1}^n w_i^2 \mathbf{C}_{x_i} = \mathbf{C}_0 \sum_{i=1}^n w_i^2 / L_i$$

which leads to

$$1/L_{\text{eff}} = \sum_{i=1}^n w_i^2 / L_i$$

An important special case of the above is a weighted average, where

$$w_i = L_i / \sum_{i=1}^n L_i$$

and, therefore

$$L_{\text{eff}} = \sum_{i=1}^n L_i$$

Instead of a weighted average, one often computes a weighted sum, a simplest case being a difference or sum of two categories. For a difference $w_1 = 1$ and $w_2 = -1$ and thus

$$1/L_{\text{eff}} = 1/L_1 + 1/L_2$$

or

$$L_{\text{eff}} = \frac{L_1 L_2}{L_1 + L_2}$$

Interestingly, the same holds for a sum, where $w_1 = w_2 = 1$. Generalizing, for any combination of sums and differences, where $w_i = 1$ or $w_i = -1$, $i = 1 \dots n$, we have

$$1/L_{\text{eff}} = \sum_{i=1}^n 1/L_i$$

6.4 Inverse-operator decomposition

The program `mne_inverse_operator` calculates the decomposition $\mathbf{A} = \mathbf{G}\mathbf{R}^C = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$, described in Section 6.2.5. It is normally invoked from the convenience script `mne_do_inverse_operator`. This section describes the options to `mne_inverse_operator` should a user need to invoke it directly for special-purpose processing.

The command-line options of `mne_inverse_operator` are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--meg

Employ MEG data in the calculation of the estimates.

--eeg

Employ EEG data in the calculation of the estimates. Note: The EEG computations have not been thoroughly tested at this time.

--fixed

Use fixed source orientations normal to the cortical mantle. By default, the source orientations are not constrained.

--loose <amount>

Employ a loose orientation constraint (LOC). This means that the source covariance matrix entries corresponding to the current component normal to the cortex are set equal to one and the transverse components are set to <amount>. Recommended value of amount is 0.2...0.6.

--loosevar <amount>

Use an adaptive loose orientation constraint. This option can be only employed if the source spaces included in the forward solution have the patch information computed, see Section 3.5. Blaa blaa...

--fwd <name>

Specifies the name of the forward solution to use.

--senscov <name>

Specifies the name of the noise-covariance matrix to use. If this file contains a projection operator, attached by *mne_browse_raw* and *mne_process_raw*, no additional projection vectors can be added with the `--proj` option.

--gradreg <value>

Regularize the planar gradiometer section (channels for which the unit of measurement is T/m) of the noise-covariance matrix by the given amount. The value is restricted to the range 0...1. For details, see Section 6.2.4.

--magreg <value>

Regularize the magnetometer and axial gradiometer section (channels for which the unit of measurement is T) of the noise-covariance matrix by the given amount. The value is restricted to the range 0...1. For details, see Section 6.2.4.

--eegreg <value>

Regularize the EEG section of the noise-covariance matrix by the given amount. The value is restricted to the range 0...1. For details, see Section 6.2.4.

--diagnoise

Omit the off-diagonal terms from the noise-covariance matrix in the computations. This may be useful if the amount of signal-free data has been insufficient to calculate a reliable estimate of the full noise-covariance matrix.

--srccov <name>

Specifies the name of the diagonal source-covariance matrix to use. By default the source covariance matrix is a multiple of the identity matrix. This option can be employed to incorporate the fMRI constraint. The software to create a source-covariance matrix file from fMRI data will be provided in a future release of this software package.

--depth

Employ depth weighting. For details, see Section 6.2.10.

--weightexp <value>

This parameter determines the steepness of the depth weighting function (default = 0.8). For details, see Section 6.2.10.

--weightlimit <value>

Maximum relative strength of the depth weighting (default = 10). For details, see Section 6.2.10.

--fmri <name>

With help of this w file, the source covariance matrix can be biased, *i.e.*, an *a priori* weighting can be applied. The source of the bias function is usually fMRI but may be also some other data, provided that the bias function can be expressed as a scalar value on the cortical surface, stored in a w file. It is recommended that this w file is appropriately smoothed (see Section 8.3) in *mne_analyze*, *tksurfer* or with *mne_smooth_w* to contain nonzero values at all vertices of the triangular tessellation of the cortical surface. The name of the file given is used as a stem of the w files. The actual files should be called <name>-lh.pri and <name>-rh.pri for the left and right hemisphere bias files, respectively. The application of the bias is discussed in Section 6.2.11.

--fmrithresh <value>

This option is mandatory and has an effect only if a weighting function has been specified with the **--fmri** option. If the value in the *a priori* files falls below this value at a particular source space point, the source covariance matrix values are multiplied by the value specified with the **--fmrio** option (default 0.1). Otherwise it is left unchanged.

--fmrio <value>

The value by which the source covariance elements are multiplied if the *a priori* weight falls below the threshold set with **--fmrithresh**, see above.

--bad <name>

A text file to designate bad channels, listed one channel name on each line of the file. If the noise-covariance matrix specified with the **--sensorscov** option contains projections, bad channel lists can be included only if they specify all channels containing non-zero entries in a projection vector. For example, bad channels can usually specify all magnetometers or all gradiometers since the projection vectors for these channel types are completely separate. Similarly, it is possible to include MEG data only or EEG data only by using only one of **--meg** or **--eeg** options since the projection vectors for MEG and EEG are always separate.

--surfsrc

Use a source coordinate system based on the local surface orientation at the source location. By default, the three dipole components

are pointing to the directions of the x, y, and z axis of the coordinate system employed in the forward calculation (usually the MEG head coordinate frame). This option changes the orientation so that the first two source components lie in the plane normal to the surface normal at the source location and the third component is aligned with it. If patch information is available in the source space, the normal is the average patch normal, otherwise the vertex normal at the source location is used. If the `--loose` or `--loosevar` option is employed, `--surfsrc` is implied.

--exclude <name>

Exclude the source space points defined by the given FreeSurfer 'label' file from the source reconstruction. This is accomplished by setting the corresponding entries in the source-covariance matrix equal to zero. The name of the file should end with `-lh.label` if it refers to the left hemisphere and with `-rh.label` if it lists points in the right hemisphere, respectively.

--proj <name>

Include signal-space projection (SSP) information from this file. For information on SSP, see Section 4.11. If the projections are present in the noise-covariance matrix, the `--proj` option is not allowed.

--csd

Compute the inverse operator for surface current densities instead of the dipole source amplitudes. This requires the computation of patch statistics for the source space. Since this computation is time consuming, it is recommended that the patch statistics are precomputed and the source space file containing the patch information is employed already when the forward solution is computed, see Sections 3.5 and 3.11. For technical details of the patch information, please consult Section 6.2.8. This option is considered experimental at the moment.

--inv <name>

Save the inverse operator decomposition here.

6.5 Producing movies and snapshots

mne_make_movie is a program for producing movies and snapshot graphics frames without any graphics output to the screen. In addition, *mne_make_movie* can produce stc or w files which contain the numerical current estimate data in a simple binary format for postprocessing. These files can be displayed in *mne_analyze*, see Chapter 7, utilized in the cross-subject averaging process, see Chapter 8, and read into Matlab using the MNE Matlab toolbox, see Chapter 10.

The command-line options to *mne_make_movie* are explained in the following subsections.

6.5.1 General options

--version

Show the program version and compilation date.

--help

List the command-line options.

6.5.2 Input files

--inv <name>

Load the inverse operator decomposition from here.

--meas <name>

Load the MEG or EEG data from this file.

--set <number>

The data set (condition) number to load. This is the sequential number of the condition. You can easily see the association by looking at the condition list in *mne_analyze* when you load the file.

--stcin <name>

Specifies an stc file to read as input.

6.5.3 Times and baseline

--tmin <time/ms>

Specifies the starting time employed in the analysis. If **--tmin** option is missing the analysis starts from the beginning of the epoch.

--tmax <time/ms>

Specifies the finishing time employed in the analysis. If **--tmax** option is missing the analysis extends to the end of the epoch.

--tstep <step/ms>

Time step between consecutive movie frames, specified in milliseconds.

--integ < Δt /ms>

Integration time for each frame. Defaults to zero. The integration will be performed on sensor data. If the time specified for a frame is t_0 , the integration range will be $t_0 - \Delta t/2 \leq t \leq t_0 + \Delta t/2$.

--pick <time/ms>

Pick a time for the production of rgb, tif, or w files. Several pick options may be present. The time must be within the analysis interval, indicated by the `--tmin` and `--tmax` options. The `--rgb`, `--tif`, and `--w` options control which file types are actually produced.

--bmin <time/ms>

Specifies the starting time of the baseline. In order to activate baseline correction, both `--bmin` and `--bmax` options must be present.

--bmax <time/ms>

Specifies the finishing time of the baseline.

--baselines <file_name>

Specifies a file which contains the baseline settings. Each line of the file should contain a name of a channel, followed by the baseline value, separated from the channel name by a colon. The baseline values must be specified in basic units, i.e., Teslas/meter for gradiometers, Teslas for magnetometers, and Volts for EEG channels. If some channels are missing from the baseline file, warning messages are issued: for these channels, the `--bmin` and `--bmax` settings will be used.

6.5.4 Options controlling the estimates

--nave <value>

Specifies the effective number of averaged epochs in the input data, L_{eff} , as discussed in Section 6.3. If the input data file is one produced by `mne_browse_raw` or `mne_process_raw`, the number of averages is correct in the file. However, if subtractions or some more complicated combinations of simple averages are produced, e.g., by using the `xplotter` software, the number of averages should be manually adjusted along the guidelines given in Section 6.3. This is accomplished either by employing this flag or by adjusting the number of averages in the data file with help of the utility `mne_change_nave`.

--snr <value>

An estimate for the amplitude SNR. The regularization parameter will be set as $\lambda^2 = 1/\text{SNR}^2$. The default value is $\text{SNR} = 3$. Automatic selection of the regularization parameter is currently not supported.

--spm

Calculate the dSPM instead of the expected current value.

--sLORETA

Calculate the noise-normalized estimate using the sLORETA approach. sLORETA solutions have in general a smaller location bias than either the expected current (MNE) or the dSPM.

--signed

Indicate the current direction with respect to the cortex outer normal by sign. Currents flowing out of the cortex are thus considered positive (warm colors) and currents flowing into the cortex negative (cold colors).

--picknormalcomp

The components of the estimates corresponding to directions tangential with the cortical mantle are zeroed out.

6.5.5 Visualization options

--subject <subject>

Specifies the subject whose MRI data is employed in the visualization. This must be the same subject that was used for computing the current estimates. The environment variable `SUBJECTS_DIR` must be set to point to a locations where the subjects are to be found.

--morph <subject>

Morph the data to to the cortical surface of another subject. The Quicktime movie, stc-file, graphics snapshot, and w-file outputs are affected by this option, *i.e.*, they will take the morphing into account and will represent the data on the cortical surface of the subject defined with this option. The stc files morphed to a single subject's cortical surface are used by *mne_average_estimates* to combine data from different subjects, see Section 8.6. If morphing is selected appropriate smoothing must be specified with the **--smooth** option. The morphing process can be made faster by precomputing the necessary morphing maps with *mne_make_morph_maps*, see Section 8.4. More information about morphing and averaging can be found in Chapter 8.

--surface <surface name>

Name of the surface employed in the visualization. The default is *inflated*.

--curv <name>

Specify a nonstandard curvature file name. The default curvature files are *lh.curv* and *rh.curv*. With this option, the names become *lh.<name>* and *rh.<name>*.

--patch <name>[:<angle/deg>]

Specify the name of a surface patch to be used for visualization instead of the complete cortical surface. A complete name of a patch

file in the FreeSurface surf directory must be given. The name should begin with lh or rh to allow association of the patch with a hemisphere. Maximum of two --patch options can be in effect, one patch for each hemisphere. If the name refers to a flat patch, the name can be optionally followed by a colon and a rotation angle in degrees. The flat patch will be then rotated counterclockwise by this amount before display. You can check a suitable value for the rotation angle by loading the patch interactively in *mne_analyze*.

--width <value>

Width of the graphics output frames in pixels. The default width is 600 pixels.

--height <value>

Height of the graphics output frames in pixels. The default height is 400 pixels.

--mag <factor>

Magnify the the visualized scene by this factor.

--lh

Select the left hemisphere for graphics output. By default, both hemisphere are processed.

--rh

Select the right hemisphere for graphics output. By default, both hemisphere are processed.

--view <name>

Select the name of the view for mov, rgb, and tif graphics output files. The default viewnames, defined in `$MNE_ROOT/setup/mne/mne_analyze/eyes`, are *lat* (lateral), *med* (medial), *ven* (ventral), and *occ* (occipital). You can override these defaults by creating the directory `.mne` under your home directory and copying the eyes file there. Each line of the eyes file contains the name of the view, the viewpoint for the left hemisphere, the viewpoint for the right hemisphere, left hemisphere up vector, and right hemisphere up vector. The entities are separated by semicolons. Lines beginning with the pound sign (#) are considered to be comments.

--smooth <nstep>

Number of smoothsteps to take when producing the output frames. Depending on the source space decimation, an appropriate number is 4 – 7. Smoothing does not have any effect for the original brain if stc files are produced. However, if morphing is selected smoothing is mandatory even with stc output. For details of the smoothing procedure, see Section 8.3.

6.5.6 Thresholding

--fthresh <value>

Specifies the threshold for the displayed colormaps. At the threshold, the overlaid color will be equal to the background surface color. For currents, the value will be multiplied by 10^{-10} . The default value is 8.

--fmid <value>

Specifies the midpoint for the displayed colormaps. At this value, the overlaid color will be red (positive values) or blue (negative values). For currents, the value will be multiplied by 10^{-10} . The default value is 15.

--fmax <value>

Specifies the maximum point for the displayed colormaps. At this value, the overlaid color will be bright yellow (positive values) or light blue (negative values). For currents, the value will be multiplied by 10^{-10} . The default value is 20.

--fslope <value>

Included for background compatibility. If this option is specified and --fmax option is *not* specified, $F_{\max} = F_{\text{mid}} + 1/F_{\text{slope}}$.

6.5.7 Output files

--mov <name>

Produce QuickTime movie files. This is the 'stem' of the output file name. The actual name is derived by stripping anything upto and including the last period from the end of <name>. According to the hemisphere, -lh or -rh is then appended. The name of the view is indicated with -<viename>. Finally, .mov is added to indicate a QuickTime output file. The movie is produced for all times as dictated by the --tmin, --tmax, --tstep, and --integ options.

--qual <value>

Quality of the QuickTime movie output. The default quality is 80 and allowed range is 25 - 100. The size of the movie files is a monotonously increasing function of the movie quality.

--rgb <name>

Produce rgb snapshots. This is the 'stem' of the output file name. The actual name is derived by stripping anything upto and including the last period from the end of <name>. According to the hemisphere, -lh or -rh is then appended. The name of the view is indicated with -<viename>. Finally, .rgb is added to indicate an rgb output file. Files are produced for all picked times as dictated by the --pick and --integ options.

--tif <name>

Produce tif snapshots. This is the ‘stem’ of the output file name. The actual name is derived by stripping anything upto and including the last period from the end of <name>. According to the hemisphere, -lh or -rh is then appended. The name of the view is indicated with -<viewname>. Finally, .tif is added to indicate an rgb output file. Files are produced for all picked times as dictated by the --pick and --integ options. The tif output files are *not* compressed. Pass the files through an image processing program to compress them.

--w <name>

Produce w file snapshots. This is the ‘stem’ of the output file name. The actual name is derived by stripping anything upto and including the last period from the end of <name>. According to the hemisphere, -lh.w or -rh.w is then appended. Files are produced for all picked times as dictated by the --pick and --integ options.

--stc <name>

Produce stc files for either the original subject or the one selected with the --morph option. These files will contain data only for the decimated locations. If morphing is selected, appropriate smoothing is mandatory. The morphed maps will be decimated with help of a subdivided icosahedron so that the morphed stc files will always contain 10242 vertices. These morphed stc files can be easily averaged together, e.g., in Matlab since they always contain an identical set of vertices.

6.5.8 Label processing

--label <name>

Specifies a label file to process. For each label file, the values of the computed estimates are listed in text files. The label files are produced by *tksurfer* or *mne_analyze* and specify regions of interests (ROIs). A label file name should end with -lh.label for left-hemisphere ROIs and with -rh.label for right-hemisphere ones. The corresponding output files are tagged with -lh-<data type>.amp and -rh-<data type>.amp, respectively. <data type> equals ‘mne’ for expected current data and ‘spm’ for dSPM data. Each line of the output file contains the waveform of the output quantity at one of the source locations falling inside the ROI. For more information about the label output formats, see Section 7.13.3.1.

--labelcoords

Include coordinates of the vertices in the output. The coordinates will be listed in millimeters in the coordinate system which was specified for the forward model computations.

--labelverts

Include vertex numbers in the output. The numbers refer to the complete triangulation of the corresponding surface and are zero based. The vertex numbers are by default on the first row or first column of the output file depending on whether or not the `--labeltimebytime` option is present.

--labeltimebytime

Output the label data time by time instead of the default vertex-by-vertex output.

--labeltag <tag>

End the output files with the specified tag. By default, the output files will end with `-mne.amp` or `-spm.amp` depending on whether MNE or one of the noise-normalized estimates (dSPM or sLORETA) was selected.

--labeloutdir <directory>

Specifies the directory where the output files will be located. By default, they will be in the current working directory.

--labelcomments

Include comments in the output files. The comment lines begin with the percent sign to make the files compatible with Matlab.

--scaleby <factor>

By default, the current values output to the files will be in the actual physical units (A/m). This option allows scaling of the current values to other units. *mne_analyze* typically uses `1e10` to bring the numbers to a human-friendly scale.

6.5.9 Using stc file input

The `--stcin` option allows input of stc files. This feature has several uses:

1. QuickTime movies can be produced from existing stc files without having to resort to EasyMeg.
2. Graphics snapshot can be produced from existing stc files.
3. Existing stc files can be temporally resampled with help of the `--tmin`, `--tmax`, `--tstep`, and `--integ` options.
4. Existing stc files can be morphed to another cortical surface by specifying the `--morph` option.
5. Timecourses can be inquired and stored into text files with help of the `--label` options, see above.

6.6 Computing inverse from raw data

The purpose of the utility *mne_compute_raw_inverse* is to compute inverse solutions from either evoked-response or raw data at specified ROIs (labels) and to save the results in a fif file which can be viewed with *mne_browse_raw*, read to Matlab directly using the MNE Matlab Toolbox, see Chapter 10, or converted to Matlab format using either *mne_convert_mne_data*, *mne_raw2mat*, or *mne_epochs2mat*, see Chapter 9.

6.6.1 Command-line options

--version

Show the program version and compilation date.

--help

List the command-line options.

--in <filename>

Specifies the input data file. This can be either an evoked data file or a raw data file.

--bmin <time/ms>

Specifies the starting time of the baseline. In order to activate baseline correction, both **--bmin** and **--bmax** options must be present. This option applies to evoked data only.

--bmax <time/ms>

Specifies the finishing time of the baseline. This option applies to evoked data only.

--set <number>

The data set (condition) number to load. This is the sequential number of the condition. You can easily see the association by looking at the condition list in *mne_analyze* when you load the file.

--inv <name>

Load the inverse operator decomposition from here.

--nave <value>

Specifies the effective number of averaged epochs in the input data, L_{eff} , as discussed in Section 6.3. If the input data file is one produced by *mne_browse_raw* or *mne_process_raw*, the number of averages is correct in the file. However, if subtractions or some more complicated combinations of simple averages are produced, e.g., by using the *xplotter* software, the number of averages should be manually adjusted along the guidelines given in Section 6.3. This is accomplished either by employing this flag or by adjusting the num-

ber of averages in the data file with help of the utility *mne_change_nave*.

--snr <value>

An estimate for the amplitude SNR. The regularization parameter will be set as $\lambda^2 = 1/\text{SNR}^2$. The default value is $\text{SNR} = 3$. Automatic selection of the regularization parameter is currently not supported.

--spm

Calculate the dSPM instead of the expected current value.

--picknormalcomp

The components of the estimates corresponding to directions tangential with the cortical mantle are zeroed out.

--mricoord

Provide source locations and orientations in the MRI coordinate frame instead of the default head coordinate frame.

--label <name>

Specifies a label file to process. For each label file, the values of the computed estimates stored in a *fif* file. For more details, see Section 6.6.2. The label files are produced by *tksurfer* or *mne_analyze* and specify regions of interests (ROIs). A label file name should end with *-lh.label* for left-hemisphere ROIs and with *-rh.label* for right-hemisphere ones. The corresponding output files are tagged with *-lh-<data type>.fif* and *-rh-<data type>.fif*, respectively. *<data type>* equals 'mne' for expected current data and 'spm' for dSPM data. For raw data, *_raw.fif* is employed instead of *.fif*. The output files are stored in the same directory as the label files.

--labelselout

Produces additional label files for each label processed, containing only those vertices within the input label which correspond to available source space vertices in the inverse operator. These files have the same name as the original label except that *-lh* and *-rh* are replaced by *-sel-lh* and *-sel-rh*, respectively.

--align_z

Instructs the program to try to align the waveform signs within the label. For more information, see Section 6.6.2. This flag will not have any effect if the inverse operator has been computed with the strict orientation constraint active.

6.6.2 Implementation details

The *fif* files output from *mne_compute_raw_inverse* have various fields of the channel information set to facilitate interpretation by postprocessing software as follows:

channel name

Will be set to $J[xyz] \langle number \rangle$, where the source component is indicated by the coordinate axis name and number is the vertex number, starting from zero, in the complete triangulation of the hemisphere in question.

logical channel number

Will be set to is the vertex number, starting from zero, in the complete triangulation of the hemisphere in question.

sensor location

The location of the vertex in head coordinates or in MRI coordinates, determined by the `--mricoord` flag.

sensor orientation

The x -direction unit vector will point to the direction of the current. Other unit vectors are set to zero. Again, the coordinate system in which the orientation is expressed depends on the `--mricoord` flag.

The `--align_z` flag tries to align the signs of the signals at different vertices of the label. For this purpose, the surface normals within the label are collected into a $n_{\text{vert}} \times 3$ matrix. The preferred orientation will be taken as the first right singular vector of this matrix, corresponding to its largest singular value. If the dot product of the surface normal of a vertex is negative, the sign of the estimates at this vertex are inverted. The inversion is reflected in the current direction vector listed in the channel information, see above.

Tip: The raw data files output by *mne_compute_raw_inverse* can be converted to *mat* files with *mne_raw2mat*, see Section 9.9. Alternatively, the files can be read directly from Matlab using the routines in the MNE Matlab toolbox, see Chapter 10. The evoked data output can be easily read directly from Matlab using the *fiff_load_evoked* routine in the MNE Matlab toolbox. Both raw data and evoked output files can be loaded into *mne_browse_raw*, see Chapter 4.

CHAPTER 7 Interactive analysis

7.1 Overview

Interactive analysis of the MEG/EEG data and source estimates is facilitated by the *mne_analyze* tool. Its features include:

1. Viewing of evoked-response data or data segments in a topographical layout.
2. Alignment of MEG and head coordinate frames.
3. Display of maps of the magnetic field and electric potentials.
4. Computation and display of cortically-constrained minimum-norm current estimates and statistical maps derived from them. The solutions can be displayed on folded and inflated cortical surfaces as well as on curved and flattened surface patches.
5. Production of QuickTime™ movies and graphics snapshots in several image file formats.
6. Inquiry and saving of source waveforms at selected surface points or within ROIs defined by label files.
7. On-line morphing of the current distributions.
8. Output of snapshots in w file format.
9. Display of overlay data delivered in w and stc file formats.
10. Creation of ROI (label) files.
11. Viewing of continuous head-position data delivered by Neuromag software.

7.2 Command line options

Since *mne_analyze* is primarily an interactive analysis tool, there are only a few command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--cd <dir>

Change to this directory before starting.

--subject <name>

Specify the default subject name for surface loading.

Note: Before starting *mne_analyze* the `SUBJECTS_DIR` environment variable has to be set.

7.3 The main window

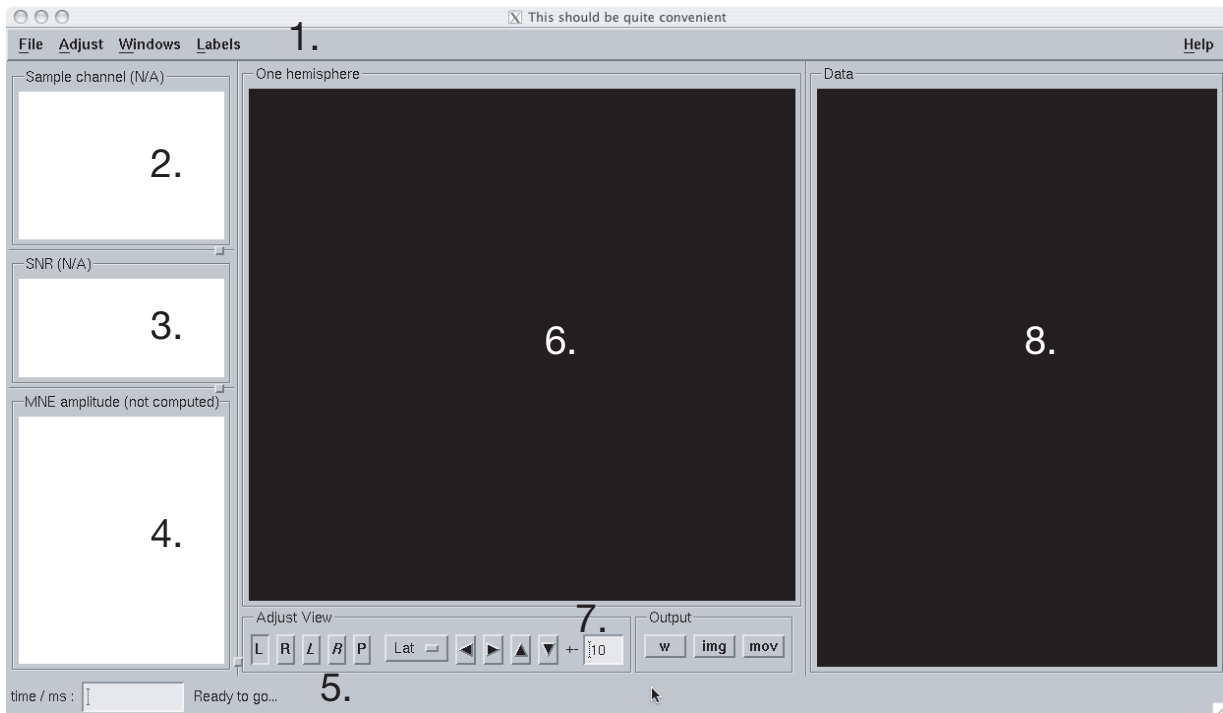


Figure 7.1 The main window of *mne_analyze*.

The main window of *mne_analyze* shown in Figure 7.1 has the following components:

1. The menu bar,
2. Display area for a sample response,
3. Display of the estimated SNR, see Section 7.12.2,
4. Display of a source waveform,
5. Message area and time-point selection text field,
6. Display area for the current estimates,
7. Controls for the current estimate display, and
8. Topographical display of data.

7.4 The menus

7.4.1 The File menu

The *File* shown in Figure 7.2 contains the following items:

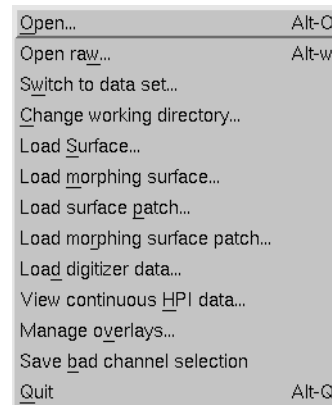


Figure 7.2 The *File* menu.

Open...

Load a new data set and an inverse operator. For details, see Section 7.5.

Open raw...

Load epoch data from a raw data file. For details, see Section 7.6.

Switch to data set...

If multiple data sets or epochs from a raw data file are loaded, this menu item brings up a list to switch between the data sets or epochs.

Change working directory...

Change the working directory of this program. This will usually be the directory where your MEG/EEG data and inverse operator are located.

Load surface...

Load surface reconstructions for the subject whose data you are analyzing, see Section 7.8.

Load morphing surface...

Load surface reconstructions of another subject for morphing, see Section 7.9.

Load surface patch...

Load a curved or flattened surface patch, see Section 7.8.

Load morphing surface patch...

Load a curved or flattened surface patch for morphing, see Section 7.9.

Load digitizer data...

Load digitizer data for coordinate frame alignment, see Section 7.16.

View continuous HPI data...

Load a data file containing continuous head position information, see Section 7.17.

Manage overlays...

Bring up the overlay manager to import data from stc and w files, see Section 7.14.

Save bad channel selection

Save the current bad channel selection created in the topographical data display, see Section 7.7.

Quit

Quit the program.

7.4.2 The Adjust menu

The contents of the *Adjust* menu is shown in Figure 7.3:

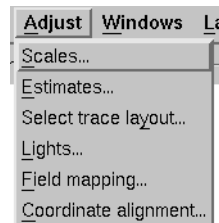


Figure 7.3 The *Adjust* menu.

Scales

Adjust the scales of the data display.

Estimates...

Adjust the properties of the displayed current estimates, see Section 7.12.

Select trace layout...

Select the layout for the topographical display, see Section 7.7.

Lights...

Adjust the lighting of the scenes in the main display and the viewer, see Sections 7.12 and 7.10.

Field mapping...

Adjust the field mapping preferences, see Section 7.11.

Coordinate alignment...

Establish a coordinate transformation between the MEG and MRI coordinate frames, see Section 7.16.

7.4.3 The Windows menu

The contents of the file menu is shown in Figure 7.4:

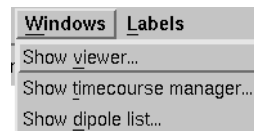


Figure 7.4 The *Windows* menu.

Show viewer...

Loads additional surfaces and pops up the viewer window. The functions available in the viewer are discussed in Section 7.10.

Show timecourse manager...

Brings up the timecourse manager if some timecourses are available. Timecourses are discussed in Section 7.13.

Show dipole list...

Display the dipole list window employed for importing current-dipole locations. For details, see Section 7.15.

7.4.4 The Labels menu

The contents of the *Labels* menu is shown in Figure 7.5. ROI analysis with help of labels is discussed in detail in Section 7.13.



Figure 7.5 The *Labels* menu.

The label menu contains the following items:

Load label...

Loads one label file for ROI analysis.

Load all labels...

Loads all label files available in a directory for ROI analysis.

Show label list...

Shows a list of all currently loaded labels for ROI analysis.

Discard all labels

Discard all labels loaded so far. The label list window will be hidden.

Clear marked vertices

Clear the label outline or a label created interactively.

7.4.5 The Help menu

The contents of the *Help* menu is shown in Figure 7.6:



Figure 7.6 The *Help* menu.

On version...

Displays the version and compilation date of the program.

On license...

Displays the license information.

On GLX...

Displays information about the OpenGL rendering context. If you experience poor graphics performance, check that the window that pops up from here says that you have a *Direct rendering context*. If not, either your graphics card or driver software needs an update.

Why the beep?

In some simple error situations, *mne_analyze* does not popup an error dialog but refuses the action and rings the bell. The reason for this can be displayed through this help menu item.

7.5 Loading data

When you select *Open...* from the *File* menu the data loading dialog shown in Figure 7.7 appears. It has four sections:

1. A standard file selection box.
2. List of available data sets. This part is automatically filled in when a proper data file is selected from the file list. You can select one or more data sets from this list. Multiple selection works with help of the shift and control keys. If multiple data sets are selected, the data set to be analyzed can be changed from the data set list accessible through *Switch to data set...* in the *File* menu.
3. List of available inverse operator decompositions in the current directory and its subdirectory called *inv*.
4. List of options:
 - a. *MRI/head transform source* specifies a file to read the MRI/MEG coordinate transformation information from. This is usually the inverse operator file. However, you can also load data with inverse operator set to *<none>* to view the data as well as field and potential maps derived thereof. In this case you need to specify the coordinate transformation file, usually located in *mr i/T1-neuromag/sets* under the subject's FreeSurfer directory.
 - b. *Use EEG average electrode ref.* selects whether the average electrode reference is applied to the data. This is only available if the inverse operator is set to *<none>*.
 - c. *nave* specifies the effective number of averages to compute the SNR correctly. Usually your measurement file contains this information.

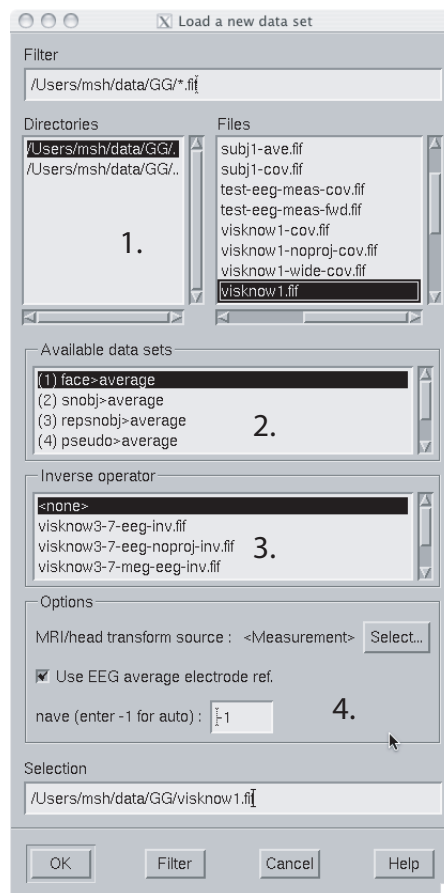


Figure 7.7 The open dialog.

After the data set has been selected, the following actions will take place:

1. The inverse operator will be loaded.
2. Baselines will be applied as specified in the scales dialog.
3. Projection will be applied to the data. If no inverse operator is specified, the source for the projection data will be the data file and the average EEG reference setting in the options. If an inverse operator is included, the projection will be read from the data file.
4. If an inverse operator is loaded, whitened data will be computed.
5. If an inverse operator is loaded, the SNR estimate as well as the effective SNR will be computed from the whitened data and displayed in the SNR window.
6. Waveforms will be shown in the topographical display as well as in the sample channel display.

If multiple data sets are loaded each data set has the following individual settings:

1. Amplitude and time scale settings,
2. Baseline,
3. Picked time point,

4. Sample channel to be displayed, and
5. MNE display preferences, see Section 7.12.1.

If a data set has not been previously displayed, the currently active settings are copied to the data set.

Tip: If you double click on an inverse operator file name displayed in the *Inverse operator* list, the command used to produce this file will be displayed in a message dialog.

7.6 Loading epochs from a raw data file

Instead of an evoked-response data file it is possible to load epochs of data (single trials) from a raw data file. This option is invoked from *File/Open raw...* The file selection box is identical to the one used for evoked responses (Figure 7.7) except that data set selector is replaced by the epoch selector shown in Figure 7.8.

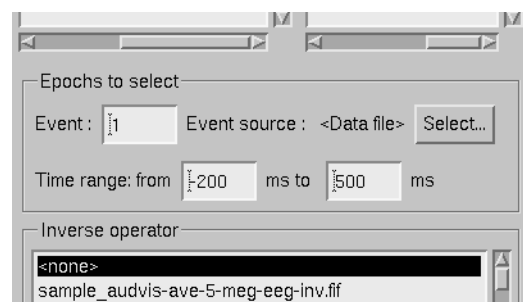


Figure 7.8 The raw data epoch selector.

The epoch selector contains the following controls:

1. The event specifier. Only events matching this number are going to be considered.
2. The event source specifier. The event source can be either the data file (the digital trigger channel STI 014) or an event data file produced with *mne_browse_raw* or *mne_process_raw*, see Section 4.4.1.8. Using an event data file is useful if, e.g., the epochs to be processed are epileptic spikes.
3. The time range specification. This determines the length of the epoch with respect to the selected event.

Once the settings have been accepted by clicking *OK*, the first matching epoch will be displayed. You can switch between epochs using the data set list accessible through *Switch to data set...* in the *File* menu.

7.7 Data displays

The MEG and EEG signals can be viewed in two ways:

1. A selection of MEG or EEG channel is shown in a topographical layout.
2. One representative channel can be selected to the *Sample channel* display by clicking on a channel in the topographical display.

In both the sample channel display and the topographical display, current time point can be selected with a left mouse click. In addition, time point of interest can be entered numerically in the text box at the bottom left corner of the main display.

7.7.1 The topographical display

A selection of channels is always shown in the rightmost part of the main display. The topographical layout to use is selected from *Adjust/Select trace layout...*, which brings up a window with a list of available layouts. The default layouts reside in `$MNE_ROOT/setup/mne_analyze/lout`. In addition any layout files residing in `$HOME/.mne/lout` are listed. The format of the layout files is the same as for the Neuromag programs *xplotter* and *xfit*. A custom EEG layout can be easily created with the *mne_make_eeg_layout* utility, see Section 11.6.

Several actions can be performed with the mouse in the topographical data display:

Left button click

Selects a time point of interest.

Left button click with control key

Selects a time point of interest and selects the channel under the pointer to the sample channel display.

Left button drag with shift key

Enlarges the view to contain only channels in the selected area.

Middle button click

Marks this channel as bad and clears all previously marked bad channel. This action is only available if an inverse operator is *not* loaded. An inverse operator dictates the selection of bad channels. The current bad channel selection can be applied to the data from *File/Save bad channel selection*.

Middle button click with control key

Extends the bad channel selection without clearing the previously active bad channels.

Right button

Brings up a popup menu which gives a choice of graphics output formats for the current topographical display.

7.7.2 The sample channel display

The sample channel display shows one of the measurement channels at the upper left corner of the *mne_analyze* user interface. A time point can be selected with a left mouse click. In addition, the following keyboard functions are associated with the sample channel display:

Down

Change the sample channel to the next channel in the scanning order.

Up

Change the sample channel to the previous channel in the scanning order.

Right

Move forward in time by 1 ms.

Control Right

Move forward in time by 5 ms.

Left

Move backward in time by 1 ms.

Control Left

Move backward in time by 5 ms.

7.7.3 Scale settings

The scales of the topographical and sample channel display can be adjusted from the *Scales* dialog which is invoked by selecting *Adjust/Scales...* from the menus. The *Scales* dialog shown in Figure 7.9 has the following entries:

Analyze range min [ms]

Specifies the lower limit of the time range of data to be shown.

Analyze range max [ms]

Specifies the upper limit of the time range of data to be shown.

Use full time range

If this box is checked, all data available in the data file will be shown.

Baseline min [ms]

Specifies the lower time limit of the baseline.

Baseline max [ms]

Specifies the upper time limit of the baseline.

Baseline in use

Baseline subtraction can be switched on and off from this button.

MEG amp min [fT/cm]

Lower limit of the vertical scale of planar gradiometer MEG channels.

MEG amp max [fT/cm]

Upper limit of the vertical scale of planar gradiometer MEG channels.

MEG axmult [cm]

The vertical scale of MEG magnetometers and axial gradiometers will be obtained by multiplying the planar gradiometer vertical scale limits by this value, given in centimeters.

EEG amp min [μ V]

Lower limit of the vertical scale of EEG channels.

EEG amp max [μ V]

Upper limit of the vertical scale of EEG channels.

Show stimulus channel

Show the trigger channel (STI 014) data in the sample view together with the sample channel.

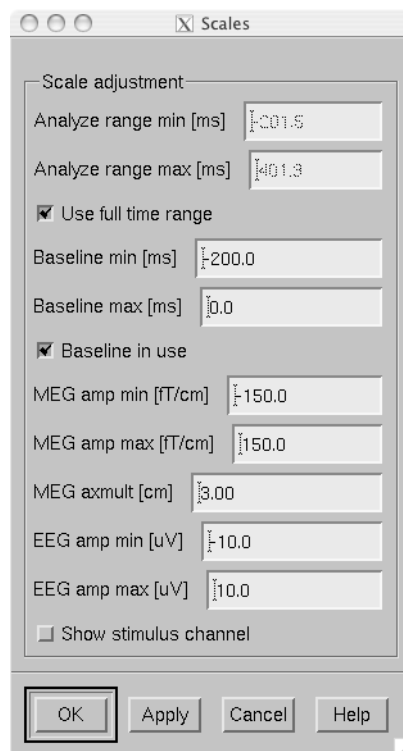


Figure 7.9 The *Scales* dialog.

7.8 The surface display

In *mne_analyze*, the current estimates are visualized on inflated or folded cortical surfaces. There are two visualization displays: the surface display, which is always visible, and the 3D viewer which is invoked from the *Windows/Show viewer...* menu selection, see Section 7.10.

A total of eight surfaces or patches can be assigned to the surface display:

1. The left and right hemisphere cortical surfaces for the subject whose data you are analyzing. These surfaces can be the inflated, white-matter, or pial surfaces. They are loaded through the *File/Load surface...* menu selection,
2. The left and right hemisphere cortical surfaces of another subject or an alternative representation of the cortical surface of the actual subject. For example, you can switch between the inflated and folded (pial or white matter) cortical surfaces very easily. These surfaces are loaded from the *File/Load morphing surface...* menu selection.
3. Left and right hemisphere curved or flat cortical patches for the subject you are analyzing. This patch is loaded from the *File/Load surface*

- patch...* menu selection. The full cortical surfaces must be loaded first before loading the patches.
4. Patches for an another subject or another pair of patches for the same subject through the *File/Load morphing surface patch...* menu selection. Again, the full cortical surfaces must have been loaded first.

7.8.1 The surface selection dialog

When *File/Load surface...* or *File/Load morphing surface...* is invoked, the surface selection dialog shown in Figure 7.10 appears.

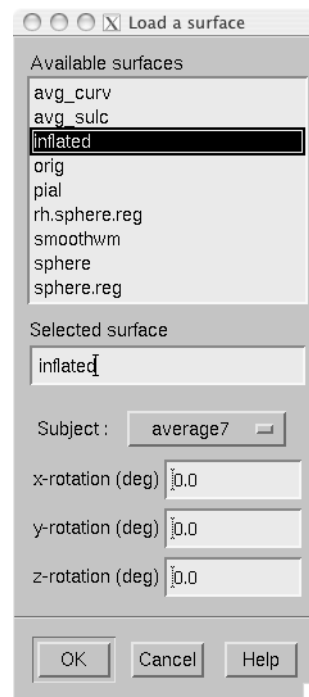


Figure 7.10 The surface selection dialog.

The dialog has the following components:

Subject: <menu>

This option menu lists the subjects available in the directory set with the SUBJECTS_DIR environment variable.

Available surfaces

Lists the surfaces available for the current subject. When you click on an item in this list, it appears in the *Selected surface* text field.

x-rotation (deg)

Specifies the initial rotation of the surface around the *x* (left to right) axis. Positive angle means a counterclockwise rotation when the surface is looked at from the direction of the positive *x* axis. Some-

times a more pleasing visualization is obtained when this rotations are specified when the surface is loaded.

y-rotation (deg)

Specifies the initial rotation of the surface around the y (back to front) axis.

z-rotation (deg)

Specifies the initial rotation of the surface around the z (bottom to up) axis.

7.8.2 The patch selection dialog

The surface patches are loaded with help of the patch selection dialog, which appears when *File/Load surface patch...* or *File/Load morphing surface patch...* is selected. This dialog, shown in Figure 7.11, contains a list of available patches and the possibility to rotate the a flat patch counterclockwise by the specified number of degrees from its original orientation. The patch is automatically associated with the correct hemisphere on the basis of the two first letters in the patch name (lh = left hemisphere, rh = right hemisphere).

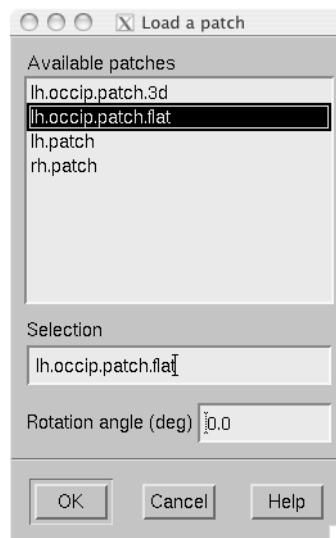


Figure 7.11 The patch selection dialog.

7.8.3 Controlling the surface display

The main surface display has a section called *Adjust view*, which has the controls shown in Figure 7.12:

L and R

Select the left or right hemisphere surface loaded through *File/Load surface...*

L and R

Select the left or right hemisphere surface loaded through *File/Load morphing surface...*

P

Select the patch associated with the currently selected surface.

Option menu

Select one of the predefined view orientations, see Section 7.8.4, below.

Arrow buttons

Rotate the surface by increments specified in degrees in the text box next to the arrows.



Figure 7.12 Surface controls.

The display can be also adjusted using keyboard shortcuts, which are available once you click in the main surface display with the left mouse button:

Arrow keys

Rotate the surface by increments specified in degrees in the *Adjust View* section.

+

Enlarge the image.

-

Reduce the image.

=

Return to the default size.

r

Rotate the image one full revolution around z axis using the currently specified rotation step. This is useful for producing a sequence of images when automatic image saving is on, see Section 7.8.6.

s

Produces a graphics file which contains a snapshot of the currently displayed image. For information on snapshot mode, see Section 7.8.6.

.

Stops the rotation invoked with the ‘r’ key, see above.

7.8.4 Defining viewing orientations

The list of viewing orientations available in the *Adjust View* section of the main surface display is controlled by a text file. The system-wide defaults reside in `$MNE_ROOT/setup/mne/mne_analyze/eyes`. If the file `$HOME/.mne/eyes` exists, it is used instead.

All lines in the eyes file starting with # are comments. The view orientation definition lines have the format:

```
<name>:<Left>:<Right>:<Left up>:<Right up> ,
```

where

<name>

is the name of this viewing orientation,

<Left>

specifies the coordinates of the viewing ‘eye’ location for the left hemisphere, separated by spaces,

<Right>

specifies the coordinates of the viewing location for the right hemisphere,

<Left up>

specifies the direction which is pointing up in the image for left hemisphere, and

<Right up>

is the corresponding up vector for the right hemisphere.

All values are given in a coordinate system where positive *x* points to the right, positive *y* to the front, and positive *z* up. The lengths of the vectors specified for each of the four items do not matter, since parallel projection is used and the up vectors will be automatically normalized. The up vectors are usually 0 0 1, *i.e.*, pointing to the positive *z* direction unless the view is directly from above or below or if some special effect is desired.

The names of viewing orientations should be less than 9 characters long. Otherwise, the middle pane of the main display will not be able to accom-

modate all the controls. The widths of the main window panes can be adjusted from the squares at the vertical sashes separating the panes.

7.8.5 Adjusting lighting

The scenes shown in the main surface display and the viewer, described in Section 7.10, are lit by fixed diffuse ambient lighting and a maximum of eight light sources. The states, locations, and colors of these light sources can be adjusted from the lighting adjustment dialog shown in Figure 7.13, which can be accessed through the *Adjust/Lights...* menu choice. The colors of the lights can be adjusted numerically or using a color adjustment dialog accessible through the *Color...* buttons.

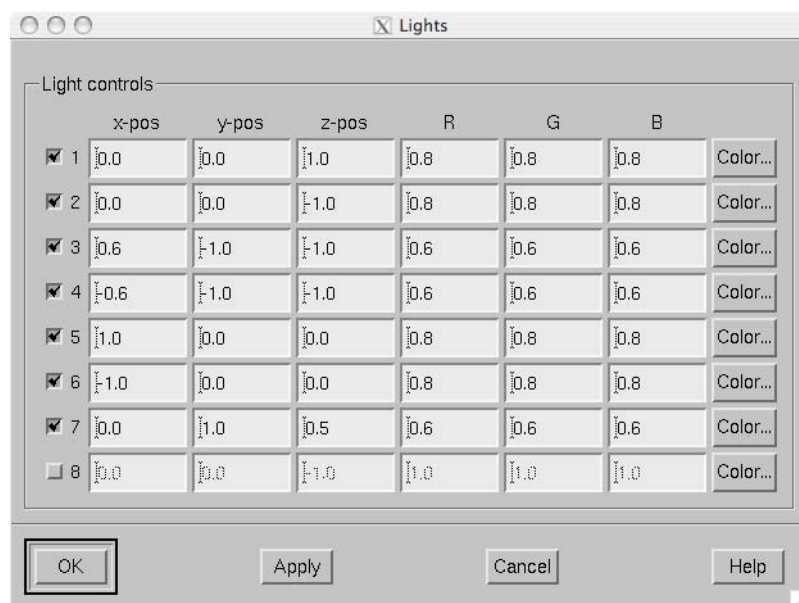


Figure 7.13 The lighting adjustment dialog.

7.8.6 Producing output files

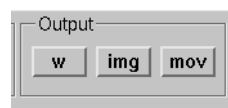


Figure 7.14 Graphics output controls.

Three types of output files can be produced from the main surface display using the graphics output buttons shown in Figure 7.14:

w files (w button)

These files are simple binary files, which contain a list of vertex numbers on the cortical surface and their current data values. The *w*

files will be automatically tagged with `-lh.w` and `-rh.w`. They will only contain vertices which currently have a nonzero value.

Graphics snapshots (*img* button)

These files will contain an exact copy of the image in tif or rgb formats. The output format and the output mode is selected from the image saving dialog shown in Figure 7.15. For more details, see Section 7.8.7. If snapshot or automatic image saving mode is in effect, the *img* button terminates this mode.

QuickTime™ movies (*mov* button)

These files will contain a sequence of images as a QuickTime™ movie file. The movie saving dialog shown in Figure 7.16 specifies the time range and the interval between the frames as well as the quality of the movies, which is restricted to the range 25...100. The size of the QuickTime file produced is approximately proportional to the quality.

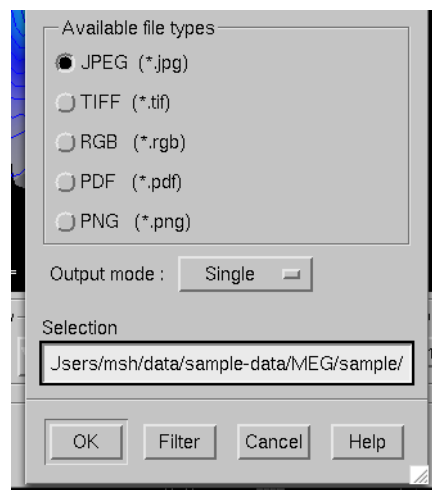


Figure 7.15 File type selection in the image saving dialog.

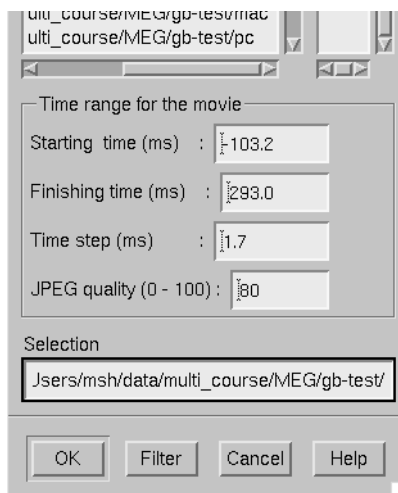


Figure 7.16 The controls in the movie saving dialog.

7.8.7 Image output modes

The image saving dialog shown in Figure 7.15 selects the format of the image files produced and the image output mode. The buttons associated with different image format change the file name filter in the dialog to display files of desired type. However, the final output format is defined by the ending of the file name in the *Selection* text field as follows:

jpg

JPEG (Joint Photographic Experts Group) format. Best quality jpeg is always produced.

tif or tiff

Uncompressed TIFF (Tagged Image File Format).

rgb

RGB format.

pdf

Portable Document File format.

png

Portable Network Graphics format.

Important: Only TIFF and RGB output routines are compiled into *mne_analyze* for other output format to work, the following programs must be present in your system: *tifftopdf*, *tifftopnm*, *pnmtjpeg*, and *pnm-topng*.

There are three image saving modes which can be selected from the option menu labelled *Output mode*:

Single

When *OK* is clicked one file containing the present image is output.

Snapshot

A new image file is produced every time **s** is pressed in the image window, see Sections 7.8.3 and 7.10.1. The image file name is used as the stem of the output files. For example, if the name is, `sample.jpg`, the output files will be `sample_shot_001.jpg`, `sample_shot_002.jpg`, *etc.*

Automatic

A new image file is produced every time the image window changes. The image file name is used as the stem of the output files. For example, if the name is, `sample.jpg`, the output files will be `sample_001.jpg`, `sample_002.jpg`, *etc.*

7.9 Morphing

The displayed surface distributions can be morphed to to another subject's brain using the spherical morphing procedure, see Chapter 8. In addition to the morphing surfaces loaded through *File/Load morphing surface...* surface patches for the same subject can be loaded through *File/Load morphing surface patch....* Switching between main and morphing surfaces is discussed in Section 7.8.3.

Any labels displayed are visible on any of the surfaces displayed in the main surface display. Time points can be picked in any of the surfaces. As a result, the corresponding timecourses will be shown in the MNE amplitude window, see Section 7.13.

7.10 The viewer

7.10.1 Overview

When *Windows/Show viewer...* is selected, the following additional surfaces will be loaded:

1. The left and right hemisphere pial surfaces,
2. The surface representing the inner helmet shaped wall of the dewar on which the MEG sensors are located,
3. The scalp surface, and
4. The BEM surfaces.

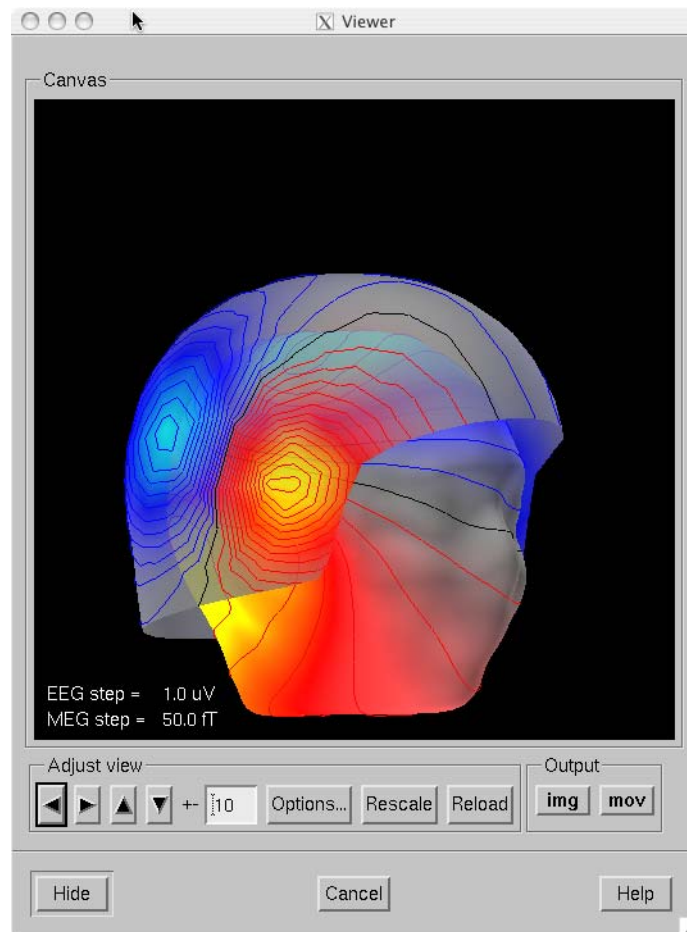


Figure 7.17 The viewer window with a visualization of MEG and EEG contour maps.

The scalp surface is loaded from the file `bem/<subject>-head.fif` under the subject's FreeSurfer directory. This surface is automatically prepared if you use the watershed algorithm as described in Section A.1. If you have another source for the head triangulation you can use the utility `mne_surf2bem` to create the fif format scalp surface file, see Section 5.4.

If a file called `bem/<subject>-bem.fif` under the subject's FreeSurfer directory is present, `mne_analyze` tries to load the BEM surface triangulations from there. This file can be a symbolic link to one of the `-bem.files` created by `mne_prepare_bem_model`, see Section 5.5. If the BEM file contains a head surface triangulation, it will be used instead of the one present in the `bem/<subject>-head.fif` file.

Once all required surfaces have been loaded, the viewer window shown in Figure 7.17 pops up. In addition to the display canvas, the viewer has *Adjust view* controls similar to the main surface display and options for graphics output. The *Adjust view* controls do not have the option menu for standard viewpoints and has two additional buttons:

The output options only include graphics output as snapshots (*img*) or as movies (*mov*).

Options...

This button pops up the viewer options window which controls the appearance of the viewer window.

Rescale

This button adjusts the contour level spacing in the magnetic field and electric potential contour maps so that the the number of contour lines is reasonable.

Reload

Checks the modification dates of the surface files loaded to viewer and reloads the data if the files have been changed. This is useful, *e.g.*, for display of different BEM tessellations.

The display can be also adjusted using keyboard shortcuts, which are available once you click in the viewer display with the left mouse button:

Arrow keys

Rotate the surface by increments specified in degrees in the *Adjust View* section.

+

Enlarge the image.

-

Reduce the image.

=

Return to the default size.

r

Rotate the image one full revolution around z axis using the currently specified rotation step. This is useful for producing a sequence of images when automatic image saving is on, see Section 7.8.6.

s

Produces a graphics file which contains a snapshot of the currently displayed image. For information on snapshot mode, see Section 7.8.6.

.

Stops the rotation invoked with the 'r' key, see above.

The left mouse button can be also used to inquire estimated magnetic field potential values on the helmet and head surfaces if the corresponding maps have been calculated and displayed.

7.10.2 Viewer options

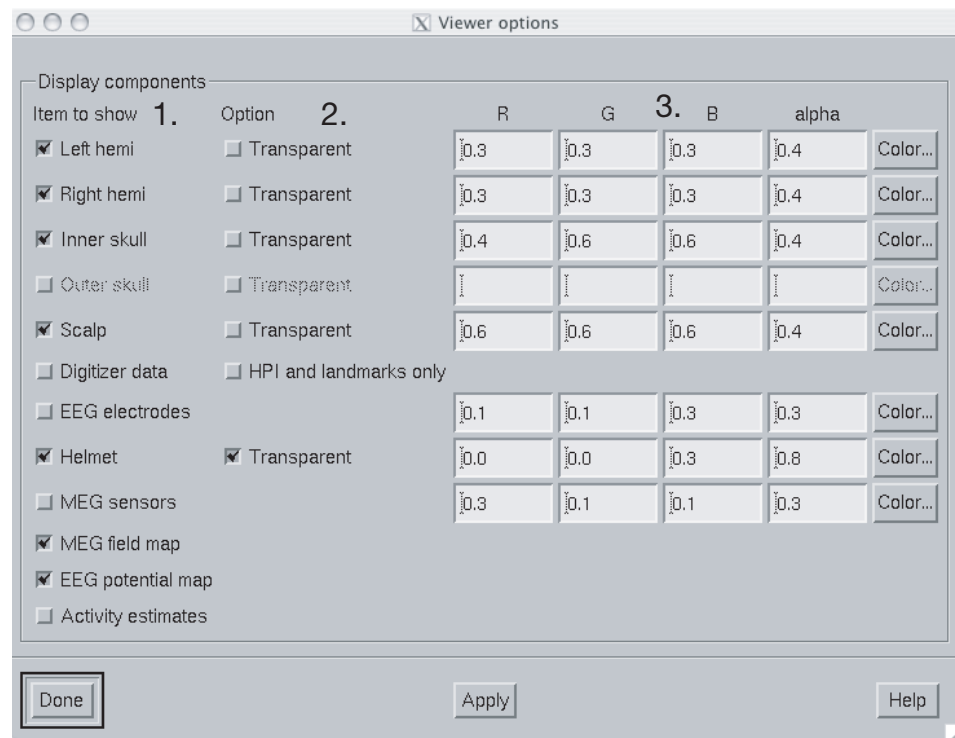


Figure 7.18 The viewer options window

The viewer options window shown above contains three main sections to control the appearance of the viewer:

1. Selectors for various items to show,
2. Options for some of the items, and
3. Control of the color and transparency of the items, if applicable. The color can be adjusted either by entering numeric values in the range 0...1 or with help of a color editor which appears from the *Color...* button. The transparency value has the same range as the other color components, zero indicating a fully transparent (invisible) surface and one a fully opaque one.

The available items are:

Left hemi

The pial surface of the left hemisphere. This surface can be made transparent. Naturally, this surface will only be visible if the scalp is made transparent.

Right hemi

The pial surface of the right hemisphere.

Inner skull

The inner skull surface. This surface can be made transparent. If parts of the pial surface are outside of the inner skull surface, they will be visible, indicating that the inner skull surface is obviously inside the inner skull. Note that this criterion is more conservative than the one imposed during the computation of the forward solution since the source space points are located on the white matter surface rather than on the pial surface. This surface is displayable only if the BEM file is present, see Section 7.10.1.

Outer skull

The outer skull surface. This surface can be made transparent. This surface is displayable only if the BEM file is present and contains the outer skull surface, see Section 7.10.1.

Scalp

The scalp surface. This surface can be made transparent. The display of this surface requires that the scalp triangulation file is present, see Section 7.10.1.

Digitizer data

The 3D digitizer data collected before the MEG/EEG acquisition. These data are loaded from *File/Load digitizer data....* The display can be restricted to HPI coil locations and cardinal landmarks with the option. The digitizer points are shown as disks whose radius is equal to the distance of the corresponding point from the scalp surface. Points outside the scalp are shown in red and those inside in blue. Distinct shades of cold and warm colors are used for the fiducial landmarks. The HPI coils are shown in green. Further information on these data and their use in coordinate system alignment is given in Section 7.16.

Helmet

The MEG measurement surface, *i.e.*, inner surface of the dewar.

EEG electrodes

The EEG electrode locations. These will be only available if your data set contains EEG channels.

MEG sensors

Outlines of MEG sensors.

MEG field map

Estimated contour map of the magnetic field component normal to the helmet surface or normal to the scalp, see Section 7.11.

EEG potential map

Interpolated EEG potential map on the scalp surface, see Section 7.11.

Activity estimates

Current estimates on the pial surface.

7.11 Magnetic field and electric potential maps

7.11.1 Overview

In *mne_analyze*, the magnetic field and potential maps displayed in the viewer window are computed using an MNE-based interpolation technique. This approach involves the following steps:

1. Establish an inverse operator to compute a minimum norm solution on a spherical surface using a spherically symmetric forward model. Instead of assuming a discrete grid of sources, a continuous distribution of tangential currents is employed. In this case the lead field dot products can be computed in closed form. Separate solutions are computed for MEG and EEG.
2. The normal component of the magnetic field or the electric potential on the helmet or head surface is computed from the MEG-based and EEG-based MNE. Since the MNE predicts the the original measurements accurately, it can also interpolate and extrapolate the data reliably. The grid of interpolation or extrapolation points can be located on the helmet or scalp surface for MEG and on the scalp surface for EEG.

The magnetic field and potential maps appear automatically whenever they are enabled from the viewer options, see Section 7.10.2.

7.11.2 Mathematical description

Let x_k be an MEG or an EEG signal at channel $k = 1 \dots N$. This signal is related to the primary current distribution $\mathbf{J}^p(\mathbf{r})$ through the lead field $\mathbf{L}_k(\mathbf{r})$:

$$x_k = \int_G \mathbf{L}_k(\mathbf{r}) \cdot \mathbf{J}^p(\mathbf{r}) dG ,$$

where the integration space G in our case is a spherical surface. The oblique boldface characters denote three-component locations vectors and vector fields.

The inner product of two leadfields is defined as:

$$\langle \mathbf{L}_j | \mathbf{L}_k \rangle = \int_G \mathbf{L}_j(\mathbf{r}) \cdot \mathbf{L}_k(\mathbf{r}) dG ,$$

These products constitute the Gram matrix $\Gamma_{jk} = \langle \mathbf{L}_j | \mathbf{L}_k \rangle$. The minimum-norm estimate can be expressed as a weighted sum of the lead fields:

$$\mathbf{J}^* = \mathbf{w}^T \mathbf{L} ,$$

where \mathbf{w} is a weight vector and \mathbf{L} is a vector composed of the continuous lead-field functions. The weights are determined by the requirement

$$\mathbf{x} = \langle \mathbf{L} | \mathbf{J}^* \rangle = \Gamma \mathbf{w} ,$$

i.e., the estimate must predict the measured signals. Hence,

$$\mathbf{w} = \Gamma^{-1} \mathbf{x} .$$

However, the Gram matrix is ill conditioned and regularization must be employed to yield a stable solution. With help of the SVD

$$\Gamma = \mathbf{U} \Lambda \mathbf{V}^T$$

a regularized minimum-norm can now found by replacing the data matching condition by

$$\mathbf{x}^{(p)} = \Gamma^{(p)} \mathbf{w}^{(p)} ,$$

where

$$\mathbf{x}^{(p)} = (\mathbf{U}^{(p)})^T \mathbf{x} \quad \text{and} \quad \Gamma^{(p)} = (\mathbf{U}^{(p)})^T \Gamma ,$$

respectively. In the above, the columns of $\mathbf{U}^{(p)}$ are the first k left singular vectors of Γ . The weights of the regularized estimate are

$$\mathbf{w}^{(p)} = \mathbf{V} \Lambda^{(p)} \mathbf{U}^T \mathbf{x} ,$$

where $\Lambda^{(p)}$ is diagonal with

$$\Lambda_{jj}^{(p)} = \begin{cases} 1/\lambda_j , j \leq p \\ \text{otherwise} \end{cases}$$

λ_j being the j^{th} singular value of Γ . The truncation point p is selected in *mne_analyze* by specifying a tolerance ϵ , which is used to determine p such that

$$1 - \frac{\sum_{j=1}^p \lambda_j}{N} < \varepsilon$$

The extrapolated and interpolated magnetic field or potential distribution estimates \mathbf{x}' in a virtual grid of sensors can be now easily computed from the regularized minimum-norm estimate. With

$$\Gamma'_{jk} = \langle L'_j | L_k \rangle ,$$

where L'_j are the lead fields of the virtual sensors,

$$\hat{\mathbf{x}}' = \Gamma' \mathbf{w}^{(k)} .$$

7.11.3 Field mapping preferences

The parameters of the field maps can be adjusted from the *Field mapping preferences* dialog shown in Figure 7.19 which is accessed through the *Adjust/Field mapping...* menu item.

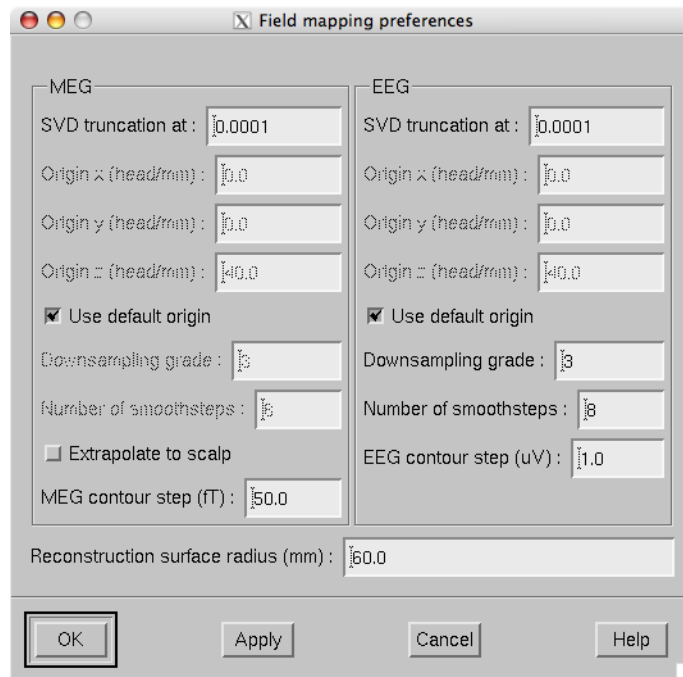


Figure 7.19 Field mapping preferences dialog.

The *Field mapping preferences* dialog has the following controls, arranged in *MEG*, *EEG*, and common sections:

SVD truncation at

Adjusts the smoothing of the field and potential patterns. This parameter specifies the eigenvalue truncation point as described in Section 7.11.2. Smaller values correspond to noisier field patterns with less smoothing.

Use default origin

The location of the origin of the spherical head model used in these computations defaults to (0 0 40) mm. If this box is unchecked the origin coordinate fields are enabled to enter a custom origin location. Usually the default origin is appropriate.

Downsampling grade

This option only applies to EEG potential maps and MEG field maps extrapolated to the head surface and controls the number of virtual electrodes or point magnetometers used in the interpolation. Allowed values are: 2 (162 locations), 3 (642 locations), and 4 (2562 locations). Usually the default value 3 is appropriate.

Number of smoothsteps

This option controls how much smoothing, see Section 8.3, is applied to the interpolated data before computing the contours. Usually the default value is appropriate.

Reconstruction surface radius

Distance of the spherical reconstruction surface from the sphere model origin. Usually default value is appropriate. For children it may be necessary to make this value smaller.

7.12 Working with current estimates

7.12.1 Preferences

The characteristics of the current estimates displayed are controlled from the *MNE preferences* dialog which pops up from *Adjust/Estimates...*

This dialog, shown in Figure 7.20, has the following controls:

SNR estimate

This controls the regularization of the estimate, i.e., the amount of allowed mismatch between the measured data and those predicted by the estimated current distribution. Smaller SNR means larger allowed mismatch. Typical range of SNR values is 1...7. As discussed in Section 6.2, the SNR value can be translated to the current variance values expressed in the source-covariance matrix R . This

translation is presented as the equivalent current standard-deviation value

Show

This radio button box selects the quantity to display. *MNE* is the minimum norm estimate (estimated value of the current), *dSPM* is the noise-normalized MNE, and *sLORETA* is another version of the noise-normalized solution which is claimed to have a smaller location bias than the dSPM.

Mask with

If MNE is selected in the *Show* radio button box, it is possible to mask the solution with one of the statistical maps. The masking map is thresholded at the value given in the *Threshold* text field and the MNE is only shown in areas with statistical values above this threshold.

Value histogram

This part of the dialog shows the distribution of the currently shown estimate values over the surface. The histogram is colored to reflect the current scale settings. The *fthresh*, *fmid*, and *fmax* values are indicated with vertical bars. The histogram is updated when the dialog is popped up and when the estimate type to show changes, not at every new time point selection. The *Refresh* button makes the histogram current at any time.

Color scale

These text fields control the color scale as described in Table 7.1.

Options

Various options controlling the estimates.

Parameter	Meaning
fthresh	If the value is below this level, it will not be shown.
fmid	Positive values at this level will show as red. Negative values will be dark blue.
fmax	Positive values at and above this level will be bright yellow. Negative values will be bright blue.

Table 7.1 The color scale parameters.

Parameter	Meaning
fmult	Apply this multiplier to the above thresholds. Default is 1 for statistical maps and 1^{-10} for currents (MNE). The vertical bar locations in the histogram take this multiplier into account but the values indicated are the threshold parameters without the multiplier.
tcmult	The upper limit of the timecourse vertical scale will be $tc_{\text{mult}} f_{\text{mult}} f_{\text{max}}$.

Table 7.1 The color scale parameters.

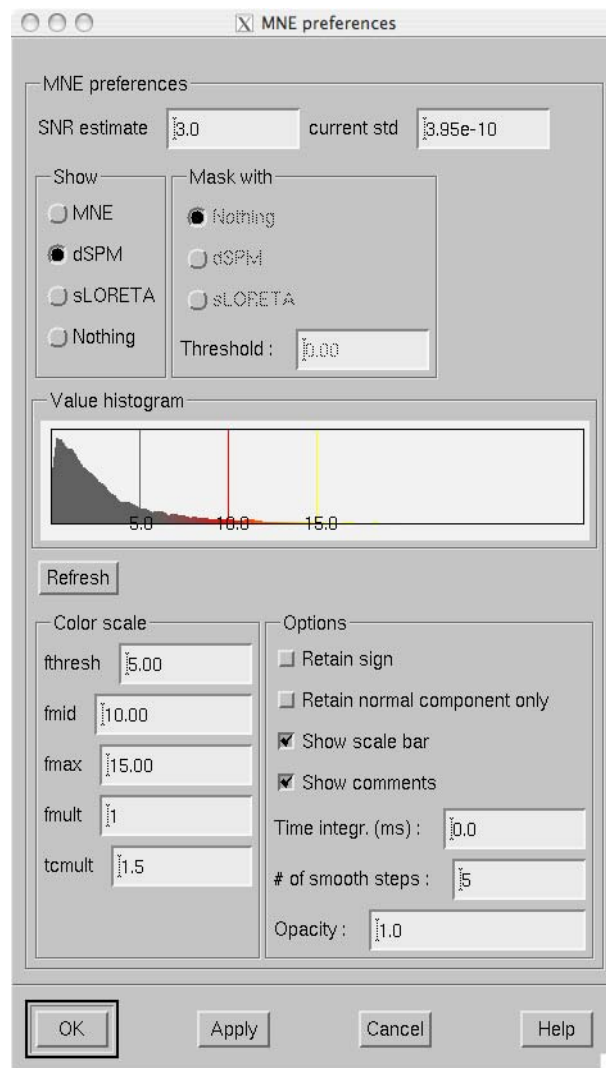


Figure 7.20 Estimate preferences dialog.

The optional parameters are:

Retain sign

With this option, the sign of the dot product between the current direction and the cortical surface normal will be used as the sign of the values to be displayed. This option yields meaningful data only if a strict or a loose orientation constraint was used in the computation of the inverse operator decomposition.

Retain normal component only

Consider only the current component normal to the cortical mantle. This option is not meaningful with completely free source orientations.

Show scale bar

Show the color scale bar at the lower right corner of the display.

Show comments

Show the standard comments at the lower left corner of the display.

Time integr. (ms)

Integration time for each frame (Δt). Before computing the estimates time integration will be performed on sensor data. If the time specified for a frame is t_0 , the integration range will be $t_0 - \Delta t/2 \leq t \leq t_0 + \Delta t/2$.

of smooth steps

Before display, the data will be smoothed using this number of steps, see Section 8.3.

Opacity

The range of this parameter is 0...1. The default value 1 means that the map overlaid on the cortical surface is completely opaque. With lower opacities the color of the cortical surface will be visible to facilitate understanding the underlying folding pattern from the curvature data displayed.

7.12.2 The SNR display

The *SNR estimate* display shows the SNR estimated from the whitened data in red and the apparent SNR inferred from the mismatch between the measured and predicted data in green.

The SNR estimate is computed from the whitened data $\tilde{\mathbf{x}}(t)$, related to the measured data $\mathbf{x}(t)$ by

$$\tilde{\mathbf{x}}(t) = \mathbf{C}^{-1/2} \mathbf{x}(t),$$

where $\mathbf{C}^{-1/2}$ is the whitening operator, introduced in Section 6.2.3.

The computation of the apparent SNR will be explained in future revisions of this manual.

7.13 Inquiring timecourses

7.13.1 Timecourses at vertices

Timecourses at individual vertices can be inquired by clicking on a desired point on the surface with the left mouse button. If the control key was down at the time of a click, the timecourse will be added to the timecourse manager but left off. With both control and shift down, the timecourse will be added to the timecourse manager and switched on. For more information on the timecourse manager, see Section 7.13.3.

The timecourses are affected by the *Retain sign* and *Retain normal component only* settings in the *MNE preferences dialog*, see Section 7.12.1.

7.13.2 Timecourses at labels

The labels provide means to interrogate timecourse information from ROIs. The label files can be created in *mne_analyze*, see Section 7.13.4 or in *tksurfer*, which is part of the FreeSurfer software. For *mne_analyze* left-hemisphere and right-hemisphere label files should be named `<name>-lh.label` and `<name>-rh.label`, respectively.

Individual label files can be loaded from *Labels/Load label...* All label files in a directory can be loaded from *Labels/Load all labels...* Once labels are loaded, the label list shown in Figure 7.21 appears. This list can be also brought up from *Labels/Show label list*. The list can be cleared from *Labels/Discard all labels*.

Warning: Because of the format of the label files *mne_analyze* can not certify that the label files loaded belong to the cortical surfaces of the present subject.

When a label is selected from the label list, the corresponding timecourse appears. The *Keep* button stores the timecourse to the timecourse manager, Section 7.13.3.

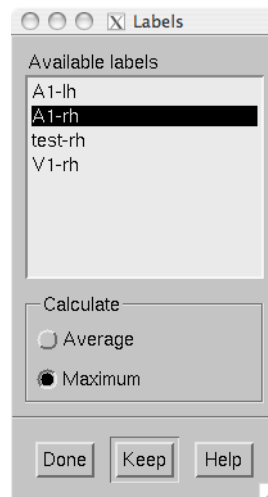


Figure 7.21 The label list.

The timecourse shown in the *MNE amplitude* window is a compound measure of all timecourses within a label. Two measures are available:

Average

Compute the average over all label vertices at each time point.

Maximum

Compute the maximum absolute value over all vertices at each time point. If the data are signed, the value is assigned the sign of the value at the maximum vertex. This may make the timecourse jump from positive to negative abruptly if vertices with different signs are included in the label.

7.13.3 The timecourse manager

The timecourse manager shown in Figure 7.22 has the following controls for each timecourse stored:

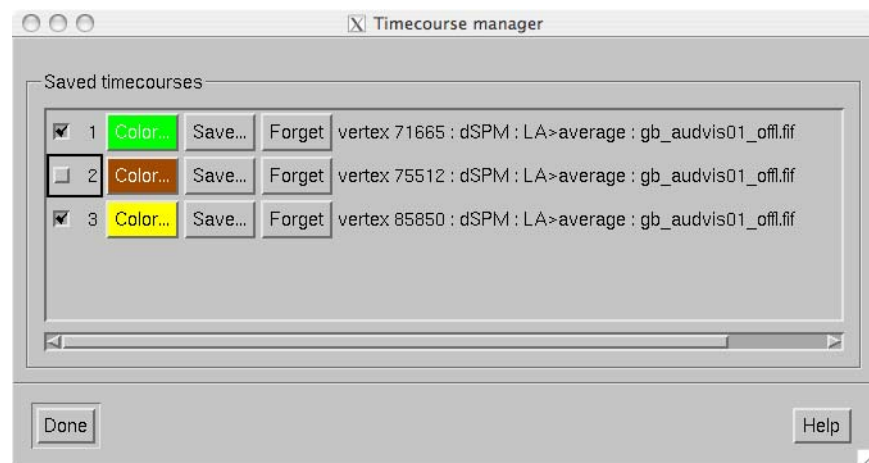


Figure 7.22 The timecourse manager.

Numbered checkbox

Switches the display of this timecourse on and off.

Color...

This button shows the color of the timecourse curve. The color can be adjusted from the color editor which appears when the button is pressed.

Save...

Saves the timecourse. If a single vertex is selected, the time course file will contain some comment lines starting with the the percent sign, one row of time point values in seconds and another with the data values. The format of the timecourse data is explained in Section 7.13.3.1, below.

Forget

Delete this timecourse from memory.

7.13.3.1 Label timecourse files

When timecourse corresponding to a label is saved, the default is to save the displayed single timecourse in a format identical to the vertex timecourses. If *Save all timecourses within the label* is selected, the *Time-by-time output* changes the output to be listed time by time rather than vertex by vertex, *Include coordinates* adds the vertex location information to the output file, and *Include vertex numbers* adds the indices of picked vertices to the output, see Figure 7.23. The vertex-by-vertex output formats is summarized in Table 7.2.

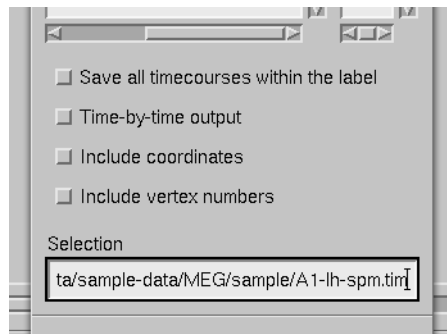


Figure 7.23 Label timecourse saving options.

Line	Contents
$1 - n_{\text{com}}$	Comment lines beginning with %
$n_{\text{com}} + 1$	$\{0.0\} [0.0 \ 0.0 \ 0.0] \ t_1 \dots t_{n_t}$
$(n_{\text{com}} + 1) - (n_p + n_{\text{com}} + 1)$	$\{p\} [x_p \ y_p \ z_p] \ v_p^{(1)} \dots v_p^{(n_t)}$

Table 7.2 Vertex-by-vertex output format. n_p is the number of vertices, n_t is the number of timepoints, n_{com} is the number of comment lines, $t_1 \dots t_{n_t}$ indicate the times in milliseconds, p is a vertex number, $x_p \ y_p \ z_p$ are the coordinates of vertex p in millimeters, and $v_p^{(1)} \dots v_p^{(n_t)}$ are the values at vertex p . Items in brackets are only included if *Include coordinates* is active. In the time-by-time output format the data portion of the file is transposed.

7.13.4 Creating new label files

It is easy to create new label files in *mne_analyze*. For this purpose, an inflated surface should be visible in the main display. Follow these steps:

1. Clear all previously selected vertices either by choosing *Labels/Clear marked vertices* or do a right button click on the surface display with the shift key down.
2. Mark vertices on the surface with right button click or by right button drag. The vertices should be defined in the desired order on the new label outline. The outline will follow the shortest path along the surface. If the viewer window has been shown at least once, the distances are calculated along the pial surface. Otherwise they will be computed along the surface visible in the main display, which normally the

inflated surface. Note that sometimes the shortest paths appear to be nonintuitive on the inflated surface.

3. Do a right button click with control key down inside the label. The outline will be completed and shown as a yellow line. The inside of the label will be filled and shown in green. A file selection box will appear to save the label. Enter the stem of the file name here. The file name will be augmented with `-lh.label` or `-rh.label`, depending on the hemisphere on which the label is specified.

7.14 Overlays

In addition to source estimates derived from MEG and EEG data, *mne_analyze* can be used to display other surface-based data. These overlay data can be imported from `w` and `stc` files containing single timeslice (static) and dynamic data (movies), respectively. These data files can be produced by *mne_make_movie*, FreeSurfer software, and custom programs or Matlab scripts.

The names of the files to be imported should end with `-<hemi>.<type>`, where `<hemi>` indicates the hemisphere (`lh` or `rh` and `<type>` is `w` or `stc`).

Overlays are managed from the dialog shown in Figure 7.24 which is invoked from *File/Manage overlays...*

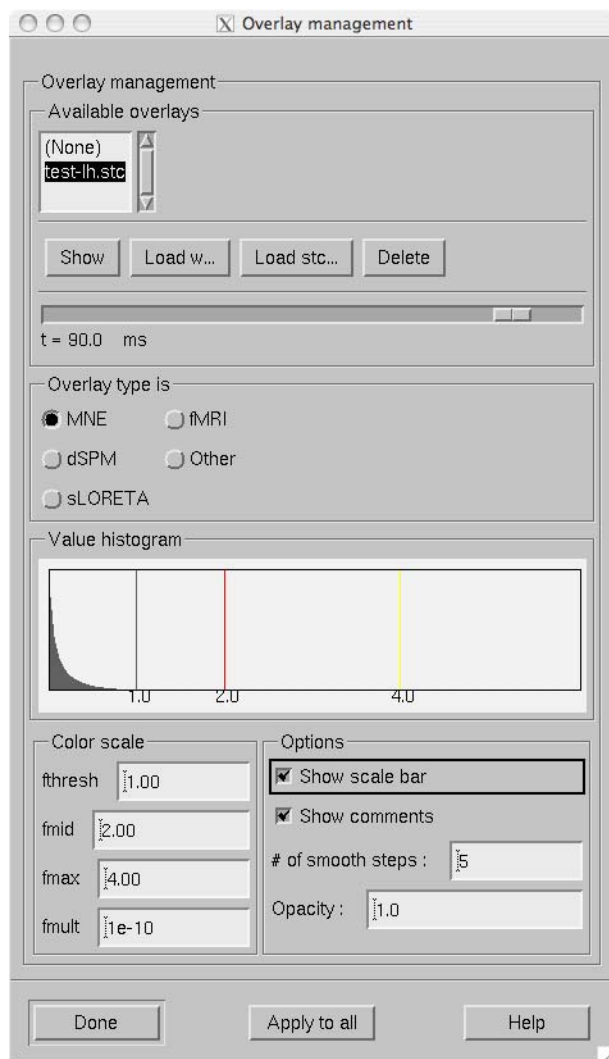


Figure 7.24 The overlay management dialog.

This dialog contains the following controls:

List of overlays loaded

Lists the names of the overlays loaded so far.

Show

Show the selected overlay and assign the settings to the current overlay.

Load w...

Load a static overlay from a w file. In the open dialog it is possible to specify whether this file contains data for the cortical surface or for scalp. Scalp overlays can be viewed in the viewer window.

Load stc...

Load a dynamic overlay from an stc file. In the open dialog it is possible to specify whether this file contains data for the cortical sur-

face or for scalp. Scalp overlays can be viewed in the viewer window.

Delete

Delete the selected overlay from memory.

Time scale slider

Will be activated if a dynamic overlay is selected. Changes the current time point.

Overlay type is

Selects the type of the data in the current overlay. Different default color scales are provided each overlay type.

Value histogram

Shows the distribution of the values in the current overlay. For large stc files this may take a while to compute since all time points are included. The histogram is colored to reflect the current scale settings. The *fthresh*, *fnid*, and *fmax* values are indicated with vertical bars.

Color scale

Sets the color scale of the current overlay. To activate the values, press *Show*. For information on color scale settings, see Table 7.1.

Options

Display options. This a subset of the options in the *MNE preferences* dialog. For details, see Section 7.12.1.

Apply to all

Apply the current settings to all loaded overlays.

It is also possible to inquire timecourses of vertices and labels from dynamic (stc) cortical overlays in the same way as from original data and store the results in text files. If a static overlay (w file) or a scalp overlay is selected, the timecourses are picked from the data loaded, if available.

7.15 Importing current dipoles

There is a possibility import current dipole locations from the Neuromag source modelling program *xfit* to *mne_analyze*. Dipoles can be imported in two ways:

1. Bring up the dipole list window from *Windows/Show dipole list....*
Drag and drop selected dipoles from one of the *xfit* dipole list to this list using the middle mouse button.
2. Drag and drop dipoles from one of the *xfit* dipole lists over the main surface display. The dipole list will appear and contain the dropped dipoles.

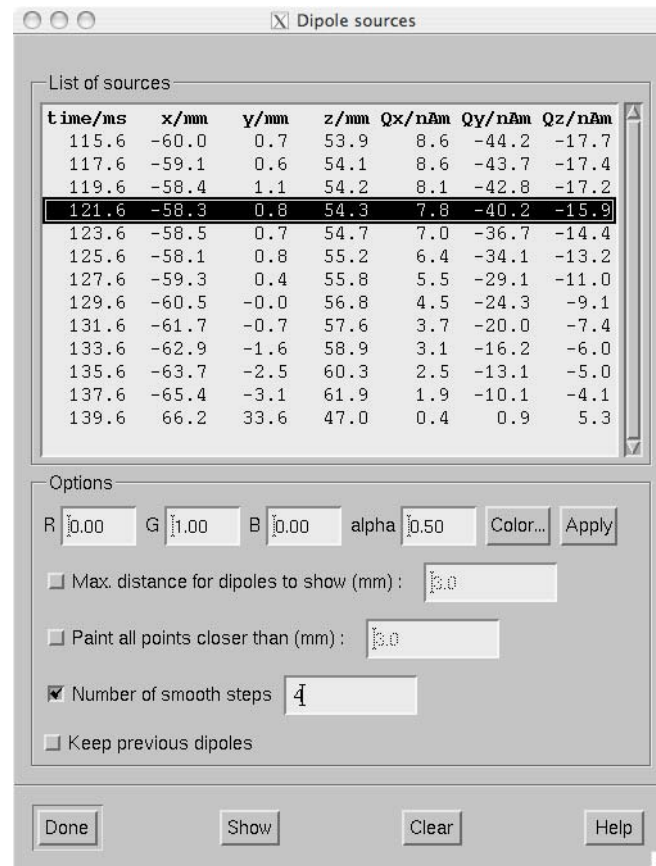


Figure 7.25 The dipole list.

The dipole list window is shown in Figure 7.25. When you double click on one of the dipoles or select several dipoles and click *Show* points on the surface displayed in the vicinity of the dipoles will be painted according to the specifications given in the *Options* section of the dialog:

Color

By the default, the dipoles are marked in green with transparency (alpha) set to 0.5. If you click on one of the dipoles, you can adjust the color of this dipole by editing the color values or from the color editor appearing when you click *Color...* When you click *Apply*, the new color values are attached to the selected dipole.

Max. distance for dipoles to show (mm)

If this option is on, only dipoles which are closer to the surface than the distance specified in the adjacent text field are displayed.

Paint all point closer than (mm)

Instead of indicating the point closest to the dipole all points closer than the distance given in the text field will be painted if this option is on. This choice is useful for understanding the shape of the neighborhood of a dipole on the cortical surface.

Number of smooth steps

This option spreads out the dipole marking by the given number of smooth steps to make the dipoles more clearly visible. A suitable choice is 3 or 4.

Keep previous dipoles

If this option is on, previously marked dipoles are not cleared from the display before new ones are shown.

Note: The surfaces, data, and an inverse operator must be loaded to display dipole locations. In particular, the inverse operator is needed to calculate the distance from the dipoles to the white matter source space surface. Depending on the precise location of the fitted dipole, the green spot may easily appear on a different wall of a fissure than could be expected. The fissural walls can be far apart from each other in the inflated view of the cortex even if they are physically separated by just a few millimeters. The size of the spots indicating the dipole locations do not relate to the dipole strengths or their confidence limits in any way.

7.16 Coordinate frame alignment

The MRI-MEG coordinate frame alignment tools included in *mne_analyze* utilized the 3D digitizer (Polhemus) data acquired in the beginning of each MEG/EEG session and the scalp surface triangulation shown in the viewer window. To access the coordinate frame alignment tools:

1. Load digitizer data. You can either load a data set containing digitizer information or load digitizer data from a file through the *File/Load digitizer data...* menu choice.
2. Set up the viewer window and make it visible, see Section 7.10. The viewer options should be set to show the digitizer data, see Section 7.10.2.
3. Bring up the *Adjust coordinate alignment dialog* from *Adjust/Coordinate alignment....*



Figure 7.26 The coordinate frame alignment dialog.

The coordinate frame alignment dialog contains the following sections:

1. Buttons for picking the fiducial points from the scalp surface and one for setting an initial alignment using these points. When one of the fiducials is selected, the viewer display automatically rotates to a suitable orientation to make the corresponding fiducial accessible.
2. Controls for fine tuning the alignment. These include movements along the three orthogonal coordinate axes and rotations around them. The buttons marked *L* and *R* indicate rotations in counterclockwise and clockwise directions, respectively. The amount of movement (mm) or rotation (degrees) is given in the text fields next to the adjustment buttons.
3. Access to an automatic alignment procedure, which employs the Iterative Closest Point (ICP) algorithm.

4. Listing of the current coordinate transformation.
5. Buttons for discarding outlier points (*Discard...*), for saving a stand-alone coordinate transformation (*Save...*), and for saving a complete MRI description file (*Save MRI set*).

Each iteration step of the Iterative Closest Point (ICP) consists of two mathing procedures:

1. For each digitizer point, transformed from MEG to the MRI coordinate frame using the current coordinate transformation, the closest point on the triangulated surface is determined.
2. The best coordinate transformation aligning the digitizer points with the closest points on the head surface is computed.

These two steps are iterated the designated number of times. If the *Try to keep nasion in place* option is on, the present location of the nasion receives a strong weight in the second part of each iteration step so that nasion movements are discouraged.

Warning: The ICP alignment is a recent addition to *mne_analyze*. Use it with caution. The iteration will not converge to a reasonable solution unless an initial alignment is performed first. Outlier points should be excluded as described above. No attempt is made to compensate for the possible distance of the digitized EEG electrode locations from the scalp.

Tip: One possible approach to coordinate frame alignment is discussed in Section 12.11.1.

7.16.1 Using a high-resolution head surface tessellation

The newest (development) version of FreeSurfer contains a script called *mkheadsurf* which can be used for coordinate alignment purposes. For more information, try `mkheadsurf --help`. This script produces a file called `surf/lh.smseghead`, which can be converted into a `fif` file using `mne_surf2bem`.

Suggested usage:

1. Set the `SUBJECTS_DIR` correctly.
2. Run `mkheadsurf`: `mkheadsurf -subjid <subject>`.
3. Goto the directory `$SUBJECTS_DIR/<subject>/bem`.
4. Convert the head surface file: `mne_surf2bem --surf ../surf/lh.smseghead --id 4 --check --fif <subject>-head-dense.fif`
5. Rename the existing head surface file to `<subject>-head-sparse.fif`
6. Copy `<subject>-head-dense.fif` to `<subject>-head.fif`
7. Click *Reload* in the viewer window.

After this you can switch between the dense and smooth head surface tessellations by copying either `<subject>-head-dense.fif` or `<subject>-head-sparse.fif` to `<subject>-head.fif`.

Important: While the dense head surface tessellation may help in coordinate frame alignment, it will slow down the operation of the viewer window considerably. Furthermore, it cannot be used in forward modelling due to the huge number of triangles. For the BEM, the dense tessellation does not provide much benefit because the potential distributions are quite smooth and widespread on the scalp.

7.17 Viewing continuous HPI data

The newest versions of Neuromag software allow continuous acquisition of signals from the HPI coils. On the basis of these data the relative position of the dewar and the head can be computed a few times per second. The resulting location data, expressed in the form of unit quaternions (see <http://mathworld.wolfram.com/Quaternion.html>) and a translation.

The continuous HPI data can be through the *File/View continuous HPI data...* menu item, which pops up a standard file selection dialog. If the file specified ends with `.fif` a fif file containing the continuous coordinate transformation information is expected. Otherwise, a text log file is read. Both files are produced by the Neuromag *maxfilter* software.

Once the data have been successfully loaded, the dialog shown in Figure 7.27 appears. It contains the following information:

1. Currently selected time point and overview of the data at the current time point,
2. MEG device to MEG head coordinate transformation at the current time point and the incremental transformation from the initial time-point to the current file.
3. Graphical display of the data.
4. Controls for the graphical display.

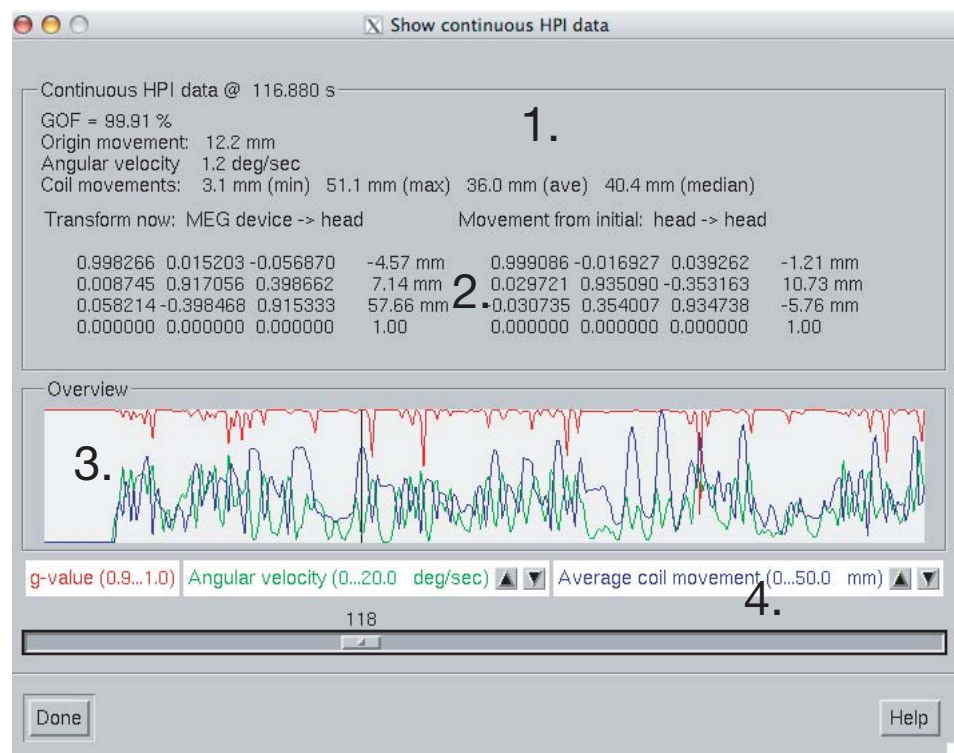


Figure 7.27 Continuous HPI data overview.

The overview items are:

GOF

Geometric mean of the goodness of fit values of the HPI coils at this time point.

Origin movement

The distance between the head coordinate origins at the first and current time points.

Angular velocity

Estimated current angular velocity of the head.

Coil movements

Comparison of the sensor locations between the first and current time points. The minimum, maximum, average, and median sensor movements are listed.

The graphical display contains the following data:

1. The geometric mean of the HPI coil goodness of fits (red curve). The scale for this curve is always 0.9...1.0.
2. The average coil (sensor) movement value (blue curve). The scale is adjustable from the buttons below the display.

3. The estimated angular velocity (deg/s, green curve). The scale is adjustable from the buttons below the display.
4. The current time point indicated with a black cursor.

The slider below the display can be used to select the time point. If you click on the slider, the current time can be adjusted with the arrow keys. The current head position with respect to the sensor array is shown in the viewer window if it is visible, see Section 7.10. Note that a complete set of items listed above is only available if a data file has been previously loaded, see Section 7.5.

CHAPTER 8 Morphing and averaging

8.1 Overview

The spherical morphing of the surfaces accomplished by *FreeSurfer* can be employed to bring data from different subjects into a common anatomical frame. This chapter describes utilities which make use of the spherical morphing procedure. *mne_morph_labels* morphs label files between subjects allowing the definition of labels in a one brain and transforming them to anatomically analogous labels in another. *mne_average_estimates* offers the capability to compute averages of data computed with the MNE software across subjects.

8.2 The morphing maps

The MNE software accomplishes morphing with help of morphing maps which can be either computed on demand or precomputed using *mne_make_morph_maps*, see Section 8.4. The morphing is performed with help of the registered spherical surfaces (`lh.sphere.reg` and `rh.sphere.reg`) which must be produced in *FreeSurfer*. A morphing map is a linear mapping from cortical surface values in subject A ($\mathbf{x}^{(A)}$) to those in another subject B ($\mathbf{x}^{(B)}$)

$$\mathbf{x}^{(B)} = \mathbf{M}^{(AB)} \mathbf{x}^{(A)},$$

where $\mathbf{M}^{(AB)}$ is a sparse matrix with at most three nonzero elements on each row. These elements are determined as follows. First, using the aligned spherical surfaces, for each vertex $\mathbf{x}_j^{(B)}$, find the triangle $T_j^{(A)}$ on the spherical surface of subject A which contains the location $\mathbf{x}_j^{(B)}$. Next, find the numbers of the vertices of this triangle and set the corresponding elements on the j th row of $\mathbf{M}^{(AB)}$ so that that $\mathbf{x}_j^{(B)}$ will be a linear interpolation between the triangle vertex values reflecting the location $\mathbf{x}_j^{(B)}$ within the triangle $T_j^{(A)}$.

It follows from the above definition that in general

$$\mathbf{M}^{(AB)} \neq (\mathbf{M}^{(BA)})^{-1},$$

i.e.,

$$\mathbf{x}^{(A)} \neq \mathbf{M}^{(BA)} \mathbf{M}^{(AB)} \mathbf{x}^{(A)},$$

even if

$$\mathbf{x}^{(A)} \approx \mathbf{M}^{(BA)} \mathbf{M}^{(AB)} \mathbf{x}^{(A)},$$

i.e., the mapping is *almost* a bijection.

8.3 About smoothing

The current estimates are normally defined only in a decimated grid which is a sparse subset of the vertices in the triangular tessellation of the cortical surface. Therefore, any sparse set of values is distributed to neighboring vertices to make the visualized results easily understandable. This procedure has been traditionally called *smoothing* but a more appropriate name might be *smudging* or *blurring* in accordance with similar operations in image processing programs.

In MNE software terms, smoothing of the vertex data is an iterative procedure, which produces a blurred image $\mathbf{x}^{(N)}$ from the original sparse image $\mathbf{x}^{(0)}$ by applying in each iteration step a sparse blurring matrix:

$$\mathbf{x}^{(p)} = \mathbf{S}^{(p)} \mathbf{x}^{(p-1)}.$$

On each row j of the matrix $\mathbf{S}^{(p)}$ there are $N_j^{(p-1)}$ nonzero entries whose values equal $1/N_j^{(p-1)}$. Here $N_j^{(p-1)}$ is the number of immediate neighbors of vertex j which had non-zero values at iteration step $p-1$. Matrix $\mathbf{S}^{(p)}$ thus assigns the average of the non-zero neighbors as the new value for vertex j . One important feature of this procedure is that it tends to preserve the amplitudes while blurring the surface image.

Once the indices non-zero vertices in $\mathbf{x}^{(0)}$ and the topology of the triangulation are fixed the matrices $\mathbf{S}^{(p)}$ are fixed and independent of the data. Therefore, it would be in principle possible to construct a composite blurring matrix

$$\mathbf{S}^{(N)} = \prod_{p=1}^N \mathbf{S}^{(p)}.$$

However, it turns out to be computationally more effective to do blurring with an iteration. The above formula for $\mathbf{S}^{(N)}$ also shows that the smudging (smoothing) operation is linear.

8.4 Precomputing the morphing maps

The utility *mne_make_morph_maps* was created to assist *mne_analyze* and *mne_make_movie* in morphing. Since the morphing maps described above take a while to compute, it is beneficial to construct all necessary maps in advance before using *mne_make_movie*. The precomputed morphing maps are located in `$SUBJECTS_DIR/morph-maps`. *mne_make_morph_maps* creates this directory automatically if it does not exist. If this directory exists when *mne_analyze* or *mne_make_movie* is run and morphing is requested, the software first looks for already existing morphing maps there. Also, if *mne_analyze* or *mne_make_movie* have to recompute any morphing maps, they will be saved to `$SUBJECTS_DIR/morph-maps` if this directory exists.

The names of the files in `$SUBJECTS_DIR/morph-maps` are of the form:

`<A>--morph.fif`,

where `<A>` and `` are names of subjects. These files contain the maps for both hemispheres, and in both directions, *i.e.*, both $\mathbf{M}^{(AB)}$ and $\mathbf{M}^{(BA)}$, as defined above. Thus the files `<A>--morph.fif` or `-<A>-morph.fif` are functionally equivalent. The name of the file produced by *mne_analyze* or *mne_make_movie* depends on the role of `<A>` and `` in the analysis.

If you choose to compute the morphing maps in batch in advance, use *mne_make_morph_maps*, which accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--redo

Recompute the morphing maps even if they already exist.

--from <subject>

Compute morphing maps from this subject.

--to <subject>

Compute morphing maps to this subject.

--all

Do all combinations. If this is used without either `--from` or `--to` options, morphing maps for all possible combinations are computed. If `--from` or `--to` is present, only maps between the specified subject and all others are computed.

Note: Because all morphing map files contain maps in both directions, the choice of `--from` and `--to` options only affect the naming of the morphing map files to be produced. *mne_make_morph_maps* creates directory `$SUBJECTS_DIR/morph-maps` if necessary.

8.5 Morphing label data

In some instances it is desirable to use anatomically equivalent labels for all subjects in a study. This can be accomplished by creating a set of labels in one subject and morphing them to another subjects anatomy using the spherical morphing procedure. *mne_morph_labels* was created to facilitate this task. It has the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--from <subject>

Name of the subject for which the labels were originally defined.

--to <subject>

Name of the subject for which the morphed labels should be created.

--labeldir <directory>

A directory containing the labels to morph.

--smooth <number>

Apply smoothing with the indicated number of iteration steps (see Section 8.3) to the labels before morphing them. This is advisable because otherwise the resulting labels may have little holes in them since the morphing map is not a bijection. By default, two smooth-steps are taken.

As the labels are morphed, a directory with the name of the subject specified with the `--to` option is created under the directory specified with `--labeldir` to hold the morphed labels.

8.6 Averaging

8.6.1 Overview

As illustrated in Figure 8.1, cross-subject averaging involves three straightforward steps:

1. Use *mne_make_movie* to create stc files morphed to a single subject. This requires the use of the `--morph` option, see Section 6.5.5. The resulting files will have identical selections of vertices on the cortical surface of the subject used in averaging. This step can be speeded up by precomputing the morphing maps employed in the process, see Section 8.4.
2. Employ *mne_average_estimates* or a Matlab script to read the data from the stc files and to produce an output stc file containing the averaged data. The MNE Matlab toolbox routines for reading and writing stc files are documented in Chapter 10.
3. Use *mne_analyze* or *mne_make_movie* to visualize the result or use the stc files from the previous step in your own Matlab routines in further processing.

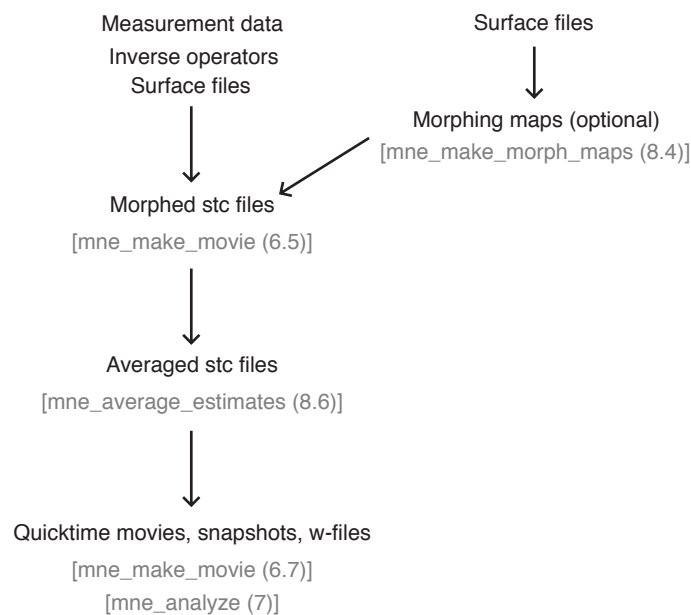


Figure 8.1 Workflow of the cross-subject averaging process in MNE software. References in parenthesis indicate sections and chapters of this manual

Note: The old utility *mne_grand_average* has been removed from the MNE software because of its inefficiency. All users should adopt the combination of *mne_make_movie* and *mne_average_estimates* instead.

Warning: With the `--ico` option it is now possible to generate source spaces with equal number of vertices in each subject. This may lead to the wrong conclusion that stc data could be averaged without doing the morphing step first. Even with identical number vertices in the source spaces it is mandatory to process the data through *mne_make_movie* to create corresponding source locations before using *mne_average_estimates*.

8.6.2 The averager

mne_average_estimates is the new utility for averaging data in stc files. It requires that all stc files represent data on one individual's cortical surface and contain identical sets of vertices. *mne_average_estimates* uses linear interpolation to resample data in time as necessary. The command line arguments are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--desc <filename>

Specifies the description file for averaging. The format of this file is described in xx below.

8.6.2.1 The description file

The description file for *mne_average_estimates* consists of a sequence of tokens, separated by whitespace (space, tab, or newline). If a token consists of several words it has to be enclosed in quotes. One or more tokens constitute a phrase, which has a meaning for the averaging definition. Any line starting with the pound sign (#) is considered to be a comment line. There are two kinds of phrases in the description file: global and contextual. The global phrases have the same meaning independent on their location in the file while the contextual phrases have different effects depending on their location in the file.

There are three types of contexts in the description file: the *global* context, an *input* context, and the *output* context. In the beginning of the file the context is *global* for defining global parameters. The *input* context defines one of the input files (subjects) while the *output* context specifies the destination for the average.

The global phrases are:

tmin <value/ms>

The minimum time to be considered. The output stc file starts at this time point if the time ranges of the stc files include this time. Otherwise the output starts from the next later available time point.

tstep <step/ms>

Time step between consecutive movie frames, specified in milliseconds.

tmax <value/ms>

The maximum time point to be considered. A multiple of *tstep* will be added to the first time point selected until this value or the last time point in one of the input stc files is reached.

integ < Δt /ms>

Integration time for each frame. Defaults to zero. The integration will be performed on sensor data. If the time specified for a frame is t_0 , the integration range will be $t_0 - \Delta t/2 \leq t \leq t_0 + \Delta t/2$.

stc <filename>

Specifies an input stc file. The filename can be specified with one of the `-lh.stc` and `-rh.stc` endings or without them. This phrase ends the present context and starts an *input* context.

deststc <filename>

Specifies the output stc file. The filename can be specified with one of the `-lh.stc` and `-rh.stc` endings or without them. This phrase ends the present context and starts the *output* context.

lh

Process the left hemisphere. By default, both hemispheres are processed.

rh

Process the right hemisphere. By default, both hemispheres are processed.

The contextual phrases are:

weight <value>

Specifies the weight of the current data set. This phrase is valid in the *input* and *output* contexts.

abs

Specifies that the absolute value of the data should be taken. Valid in all contexts. If specified in the global context, applies to all subsequent input and output contexts. If specified in the input or output contexts, applies only to the data associated with that context.

pow <value>

Specifies that the data should be raised to the specified power. For negative values, the absolute value of the data will be taken and the negative sign will be transferred to the result, unless *abs* is specified. Valid in all contexts. Rules of application are identical to *abs*.

sqrt

Means *pow 0.5*

The effects of the options can be summarized as follows. Suppose that the description file includes P contexts and the temporally resampled data are organized in matrices $\mathbf{S}^{(p)}$, where $p = 1 \dots P$ is the subject index, and the rows are the signals at different vertices of the cortical surface. The average computed by *mne_average_estimates* is then:

$$A_{jk} = w[\text{sgn}(B_{jk})]^\alpha |B_{jk}|^\beta$$

with

$$B_{jk} = \sum_{p=1}^P \bar{w}_p [\text{sgn}(S_{jk}^{(p)})]^{\alpha_p} |S_{jk}^{(p)}|^{\beta_p}$$

and

$$\bar{w}_p = \sum_{p=1}^P |w_p|.$$

In the above β_p and w_p are the powers and weights assigned to each of the subjects whereas β and w are the output weight and power value, respectively. The sign is either included ($\alpha_p = 1$, $\alpha = 1$) or omitted ($\alpha_p = 2$, $\alpha = 2$) depending on the presence of *abs* phrases in the description file.

Note: *mne_average_estimates* requires that the number of vertices in the stc files are the same and that the vertex numbers are identical. This will be the case if the files have been produced in *mne_make_movie* using the `--morph` option.

Note: More comprehensive averaging options will be included in the future versions of *mne_average_estimates*.

CHAPTER 9 Data conversion

9.1 Overview

This Chapter describes the data conversion utilities included with the MNE software suite.

9.2 Importing data from other MEG/EEG systems

This section describes the utilities to convert data from other MEG/EEG systems into the fif format.

9.2.1 Importing 4-D Neuroimaging data

The newest version of 4-D Magnes software includes the possibility to export data in fif. Please consult the documentation of the Magnes system for details of this export utility. However, the exported fif file does not include information about the compensation channels and the weights to be applied to realize software gradient compensation. To augment the Magnes fif files with the necessary information, the MNE software includes the utilities *mne_insert_4D_comp*, *mne_create_comp_data*, and *mne_add_to_meas_info*.

As a result, the complete 4D Magnes data conversion process involves the following steps:

1. Export the raw data fif file from the Magnes system.
2. If the data comes from a Magnes system where the primary (helmet) sensors are gradiometers instead of magnetometers, run *mne_fix_mag_coil_types* with the `--magnes` option to correct the channel information in the file, see Section 11.4.4.
3. Export a text file containing the Magnes compensation sensor data.
4. Create a text file containing the appropriate compensation channel weights.
5. Run *mne_insert_4D_comp* with the files created in the first two steps to merge compensation channel data with the original Magnes fif file.

6. Run *mne_create_comp_data* on the file created in step 3. to make a fif file containing the compensation weights.
7. Run *mne_add_to_meas_info* with the fif files created in steps 4. and 5. as input to result in a complete fif file containing all the necessary data.

Note: Including the compensation channel data is recommended but not mandatory. If the data are saved in the Magnes system are already compensated, there will be a small error in the forward calculations whose significance has not been evaluated carefully at this time.

9.2.2 Importing CTF data

The MNE software includes a utility *mne_ctf2fiff*, based on the BrainStorm Matlab code by Richard Leahy, John Mosher, and Sylvain Baillet, to convert data in CTF ds directory to fif format.

The command-line options of *mne_ctf2fiff* are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--verbose

Produce a verbose listing of the conversion process to stdout.

--ds <directory>

Read the data from this directory

--omit <filename>

Read the names of channels to be omitted from this text file. Enter one channel name per line. The names should match exactly with those listed in the CTF data structures. By default, all channels are included.

--fif <filename>

The name of the output file. If the length of the raw data exceeds the 2-GByte fif file file limit, several output files will be produced. These additional 'extension' files will be tagged with `_001.fif`, `_002.fif`, etc.

--evoked

Produce and evoked-response fif file instead of a raw data file. Each trial in the CTF data file is included as a separate category (condition). The maximum number of samples in each trial is limited to 25000.

--infoonly

Write only the measurement info to the output file, do not include data.

During conversion, the following files are consulted from the ds directory:

<name>.res4

This file contains most of the header information pertaining the acquisition.

<name>.hc

This file contains the HPI coil locations in sensor and head coordinates.

<name>.meg4

This file contains the actual MEG data. If the data are split across several files due to the 2-GByte file size restriction, the 'extension' files are called **<name>.<number>_meg4**.

<name>.eeg

This is an optional input file containing the EEG electrode locations. More details are given below.

If the **<name>.eeg** file, produced from the Polhemus data file with CTF software, is present, it is assumed to contain lines with the format:

```
<number> <name> <x/cm> <y/cm> <z/cm>
```

The field **<number>** is a sequential number to be assigned to the converted data point in the fif file. **<name>** is either a name of an EEG channel, one of **left**, **right**, or **nasion** to indicate a fiducial landmark, or any word which is not a name of any channel in the data. If **<name>** is a name of an EEG channel available in the data, the location is included in the Polhemus data as an EEG electrode locations and inserted as the location of the EEG electrode. If the name is one of the fiducial landmark names, the point is included in the Polhemus data as a fiducial landmark. Otherwise, the point is included as an additional head surface points.

The standard **eeg** file produced by CTF software does not contain the fiducial locations. If desired, they can be manually copied from the **pos** file which was the source of the **eeg** file.

Important: *mne_ctf2fiff* converts both epoch mode and continuous raw data file into raw data fif files. It is not advisable to use epoch mode files with time gaps between the epochs because the data will be discontinuous in the resulting fif file with jumps at the junctions between epochs. These discontinuities produce artefacts if the raw data is filtered in *mne_browse_raw*, *mne_process_raw*, or *graph*.

Note: The conversion process includes a transformation from the CTF head coordinate system convention to that used in the Neuromag systems.

9.2.3 Importing CTF Polhemus data

The CTF MEG systems store the Polhemus digitization data in text files. The utility *mne_ctf_dig2fiff* was created to convert these data files into the *fif* and *hpts* formats.

The input data to *mne_ctf_dig2fiff* is a text file, which contains the coordinates of the digitization points in centimeters. The first line should contain a single number which is the number of points listed in the file. Each of the following lines contains a sequential number of the point, followed by the three coordinates. *mne_ctf_dig2fiff* ignores any text following the *z* coordinate on each line. If the *--numfids* option is specified, the first three points indicate the three fiducial locations (1 = nasion, 2 = left auricular point, 3 = right auricular point). Otherwise, the input file must contain lines beginning with *left*, *right*, or *nasion* to indicate the locations of the fiducial landmarks.

The *hpts* output is a text file which can contain comment lines, starting with the pound sign (*#*), and data lines. Each of the data lines has the following items:

<point type> <sequential number> <x/mm> <y/mm> <z/mm>

All coordinate values are given in Neuromag head coordinates as indicated in the header comments included in each file output by *mne_ctf_dig2fiff*. The point type can be one of the following:

cardinal	Fiducial landmarks,
hpi	Head-position indicator coil location,
eeg	EEG electrode location, or
extra	Additional points acquired.

Points within each of the above categories are numbered sequentially from one.

The command-line options for *mne_ctf_dig2fiff* are:

- version**
Show the program version and compilation date.
- help**
List the command-line options.
- dig <name>**
Specifies the input data file in CTF output format.

--numfids

Fiducial locations are numbered instead of labeled, see above.

--hpts <name>

Specifies the output hpts file.

--fif <name>

Specifies the output fif file.

9.2.4 Applying software gradient compensation

Since the software gradient compensation employed in CTF systems is a reversible operation, it is possible to change the compensation status of CTF data in the data files as desired. This section contains information about the technical details of the compensation procedure and a description of *mne_compensate_data*, which is a utility to change the software gradient compensation state in evoked-response data files.

The fif files containing CTF data converted using the utility *mne_ctf2fiff* contain several compensation matrices which are employed to suppress external disturbances with help of the reference channel data. The reference sensors are located further away from the brain than the helmet sensors and are thus measuring mainly the external disturbances rather than magnetic fields originating in the brain. Most often, a compensation matrix corresponding to a scheme nicknamed *Third-order gradient compensation* is employed.

Let us assume that the data contain n_1 MEG sensor channels, n_2 reference sensor channels, and n_3 other channels. The data from all channels can be concatenated into a single vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \mathbf{x}_3^T \end{bmatrix}^T,$$

where \mathbf{x}_1 , \mathbf{x}_2 , and \mathbf{x}_3 are the data vectors corresponding to the MEG sensor channels, reference sensor channels, and other channels, respectively. The data before and after compensation, denoted here by $\mathbf{x}_{(0)}$ and $\mathbf{x}_{(k)}$, respectively, are related by

$$\mathbf{x}_{(k)} = \mathbf{M}_{(k)} \mathbf{x}_{(0)},$$

where the composite compensation matrix is

$$\mathbf{M}_{(k)} = \begin{bmatrix} \mathbf{I}_{n_1} & \mathbf{C}_{(k)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{n_3} \end{bmatrix}.$$

In the above, $\mathbf{C}_{(k)}$ is a n_1 by n_2 compensation data matrix corresponding to compensation “grade” k . It is easy to see that

$$\mathbf{M}_{(k)}^{-1} = \begin{bmatrix} \mathbf{I}_{n_1} & -\mathbf{C}_{(k)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{n_3} \end{bmatrix}.$$

To convert from compensation grade k to p one can simply multiply the inverse of one compensite compensation matrix by another and apply the product to the data:

$$\mathbf{x}_{(k)} = \mathbf{M}_{(k)} \mathbf{M}_{(p)}^{-1} \mathbf{x}_{(p)}.$$

This operation is performed by *mne_compensate_data*, which has the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--in <name>

Specifies the input data file.

--out <name>

Specifies the output data file.

--grad <number>

Specifies the desired compensation grade in the output file. The value can be 1, 2, 3, or 101. The values starting from 101 will be used for 4D Magnes compensation matrices.

Note: Only average data is included in the output. Evoked-response data files produced with *mne_browse_raw* or *mne_process_raw* may include standard errors of mean, which can not be recompensated using the above method and are thus omitted.

Tip: Raw data cannot be compensated using *mne_compensate_data*. For this purpose, load the data to *mne_browse_raw* or *mne_process_raw*, specify the desired compensation grade, and save a new raw data file.

9.2.5 Importing Magnes compensation channel data

At present, it is not possible to include reference channel data to fif files containing 4D Magnes data directly using the conversion utilities available for the Magnes systems. However, it is possible to export the compensation channel signals in text format and merge them with the MEG helmet channel data using *mne_insert_4D_comp*. This utility has the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--in <name>

Specifies the input fif file containing the helmet sensor data.

--out <name>

Specifies the output fif file which will contain both the helmet sensor data and the compensation channel data.

--ref <name>

Specifies a text file containing the reference sensor data.

Each line of the reference sensor data file contains the following information:

epoch #	is always one,
time/s	time point of this sample,
data/T	the reference channel data values.

The standard locations of the MEG (helmet) and compensation sensors in a Magnes WH3600 system are listed in `$MNE_ROOT/setup/mne/Magnes_WH3600.pos`. *mne_insert_4D_comp* matches the helmet sensor positions in this file with those present in the input data file and transforms the standard compensation channel locations accordingly to be included in the output. Since a standard position file is only provided for Magnes WH600, *mne_insert_4D_comp* only works for that type of a system.

The fif files exported from the Magnes systems may contain slightly smaller number of samples than originally acquired because the total number of samples may not be evenly divisible with a reasonable number of samples which will be used as the fif raw data file buffer size. There-

fore, the reference channel data may contain more samples than the fif file. The superfluous samples will be omitted from the end.

9.2.6 Creating software gradient compensation data

The utility *mne_create_comp_data* was written to create software gradient compensation weight data for 4D Magnes fif files. This utility takes a text file containing the compensation data as input and writes the corresponding fif file as output. This file can be merged into the fif file containing 4D Magnes data with the utility *mne_add_to_meas_info*.

The command line options of *mne_create_comp_data* are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--in <name>

Specifies the input text file containing the compensation data.

--kind <value>

The compensation type to be stored in the output file with the data. This value defaults to 101 for the Magnes compensation and does not need to be changed.

--out <name>

Specifies the output fif file containing the compensation channel weight matrix $\mathbf{C}_{(k)}$, see Section 9.2.4.

The format of the text-format compensation data file is:

```
<number of MEG helmet channels>
<number of compensation channels included>
<cname1> <cname2> ...
<name1> <weights>
<name2> <weights>
...
```

In the above $\langle name_k \rangle$ denote names of MEG helmet channels and $\langle name_k \rangle$ those of the compensation channels, respectively. If the channel names contain spaces, they must be surrounded by quotes, for example, "MEG 0111".

9.2.7 Importing EEG data saved in the Tufts University format

The utility *mne_tufts2fiff* was created in collaboration with Phillip Holcomb and Annette Schmid from Tufts University to import their EEG data to the MNE software.

The Tufts EEG data is included in three files:

1. The raw data file containing the acquired EEG data. The name of this file ends with the suffix *.raw*.
2. The calibration raw data file. This file contains known calibration signals and is required to bring the data to physical units. The name of this file ends with the suffix *c.raw*.
3. The electrode location information file. The name of this file ends with the suffix *.elp*.

The utility *mne_tufts2fiff* has the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--raw <filename>

Specifies the name of the raw data file to process.

--cal <filename>

The name of the calibration data file. If calibration data are missing, the calibration coefficients will be set to unity.

--elp <filename>

The name of the electrode location file. If this file is missing, the electrode locations will be unspecified.

Note: The conversion process includes a transformation from the Tufts head coordinate system convention to that used in the Neuromag systems.

9.2.8 Importing KIT MEG system data

The utility *mne_kit2fiff* was created in collaboration with Alec Maranz and Asaf Bachrach to import their MEG data acquired with the 160-channel KIT MEG system to MNE software.

To import the data, the following input files are mandatory:

1. The Polhemus data file (*elp* file) containing the locations of the fiducials and the head-position indicator (HPI) coils. These data are usually given in the CTF/4D head coordinate system. However, *mne_kit2fiff*

does not rely on this assumption. This file can be exported directly from the KIT system.

2. A file containing the locations of the HPI coils in the MEG device coordinate system. These data are used together with the *elp* file to establish the coordinate transformation between the head and device coordinate systems. This file can be produced easily by manually editing one of the files exported by the KIT system.
3. A sensor data file (*sns* file) containing the locations and orientations of the sensors. This file can be exported directly from the KIT system.

Note: The output *fif* file will use the Neuromag head coordinate system convention, see Section 5.2. A coordinate transformation between the CTF/4D head coordinates and the Neuromag head coordinates is included. This transformation can be read with MNE Matlab Toolbox routines, see Chapter 10.

The following input files are optional:

1. A head shape data file (*hsp* file) containing locations of additional points from the head surface. These points must be given in the same coordinate system as that used for the *elp* file and the fiducial locations must be within 1 mm from those in the *elp* file.
2. A raw data file containing the raw data values, sample by sample, as text. If this file is not specified, the output *fif* file will only contain the measurement info block.

By default *mne_kit2fiff* includes the first 157 channels, assumed to be the MEG channels, in the output file. The compensation channel data are not converted by default but can be added, together with other channels, with the `--type`. The channels from 160 onwards are designated as miscellaneous input channels (MISC 001, MISC 002, etc.). The channel names and types of these channels can be afterwards changed with the *mne_rename_channels* utility, see Section 11.4.5. In addition, it is possible to synthesize the digital trigger channel (STI 014) from available analog trigger channel data, see the `--stim` option, below. The synthesized trigger channel data value at sample k will be:

$$s(k) = \sum_{p=1}^n t_p(k) 2^{p-1},$$

where $t_p(k)$ are the thresholded from the input channel data $d_p(k)$:

$$t_p(k) = \begin{cases} 0 & \text{if } d_p(k) \leq t \\ 1 & \text{if } d_p(k) > t \end{cases}.$$

The threshold value t can be adjusted with the `--stimthresh` option, see below.

`mne_kit2fiff` accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--elp <filename>

The name of the file containing the locations of the fiducials and the HPI coils. This option is mandatory.

--hsp <filename>

The name of the file containing the locations of the fiducials and additional points on the head surface. This file is optional.

--sns <filename>

The name of file containing the sensor locations and orientations. This option is mandatory.

--hpi <filename>

The name of a text file containing the locations of the HPI coils in the MEG device coordinate frame, given in millimeters. The order of the coils in this file does not have to be the same as that in the *elp* file. This option is mandatory.

--raw <filename>

Specifies the name of the raw data file. If this file is not specified, the output fif file will only contain the measurement info block.

--sfreq <value/Hz>

The sampling frequency of the data. If this option is not specified, the sampling frequency defaults to 1000 Hz.

--lowpass <value/Hz>

The lowpass filter corner frequency used in the data acquisition. If not specified, this value defaults to 200 Hz.

--highpass <value/Hz>

The highpass filter corner frequency used in the data acquisition. If not specified, this value defaults to 0 Hz (DC recording).

--out <filename>

Specifies the name of the output fif format data file. If this file is not specified, no output is produced but the *elp*, *hpi*, and *hsp* files are processed normally.

--stim <chs>

Specifies a colon-separated list of numbers of channels to be used to synthesize a digital trigger channel. These numbers refer to the scanning order channels as listed in the *sns* file, starting from one. The digital trigger channel will be the last channel in the file. If this option is absent, the output file will not contain a trigger channel.

--stimthresh <value>

The threshold value used when synthesizing the digital trigger channel, see above. Defaults to 1.0.

--add <chs>

Specifies a colon-separated list of numbers of channels to include between the 157 default MEG channels and the digital trigger channel. These numbers refer to the scanning order channels as listed in the *sns* file, starting from one.

Important: The *mne_kit2fiff* utility has not been extensively tested yet.

9.3 Listing source space data

The utility *mne_list_source_space* outputs the source space information into text files suitable for loading into the Neuromag MRILab software.

The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--src <name>

The source space to be listed. This can be either the output from *mne_make_source_space* (*-src.fif), output from the forward calculation (*-fwd.fif), or the output from the inverse operator decomposition (*-inv.fif).

--mri <name>

A file containing the transformation between the head and MRI coordinates is specified with this option. This file can be either a Neuromag MRI description file, the output from the forward calculation (*-fwd.fif), or the output from the inverse operator decomposition (*-inv.fif). If this file is included, the output will be in head coordinates. Otherwise the source space will be listed in MRI coordinates.

--dip <name>

Specifies the ‘stem’ for the Neuromag text format dipole files to be output. Two files will be produced: *<stem>-lh.dip* and *<stem>-rh.dip*. These correspond to the left and right hemisphere part of the source space, respectively. This source space data can be imported to MRILab through the *File/Import/Dipoles* menu item.

--pnt <name>

Specifies the ‘stem’ for Neuromag text format point files to be output. Two files will be produced: *<stem>-lh.pnt* and *<stem>-rh.pnt*. These correspond to the left and right hemisphere part of the source space, respectively. This source space data can be imported to MRILab through the *File/Import/Strings* menu item.

--exclude <name>

Exclude the source space points defined by the given FreeSurfer ‘label’ file from the output. The name of the file should end with *-lh.label* if it refers to the left hemisphere and with *-rh.label* if it lists points in the right hemisphere, respectively.

--include <name>

Include only the source space points defined by the given FreeSurfer ‘label’ file to the output. The file naming convention is the same as described above under the **--exclude** option. Are ‘include’ labels are processed before the ‘exclude’ labels.

--all

Include all nodes in the output files instead of only those active in the source space. Note that the output files will be huge if this option is active.

9.4 Listing BEM mesh data

The utility *mne_list_bem* outputs the BEM meshes in text format. The default output data contains the *x*, *y*, and *z* coordinates of the vertices, listed in millimeters, one vertex per line.

The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--bem <name>

The BEM file to be listed. The file name normally ends with *-bem.fif* or *-bem-sol.fif*.

--out <name>

The output file name.

--surfno <number>

The surface number to be listed. The surfaces are numbered starting with the innermost surface. Thus, for a three-layer model the surface numbers are: 1 = scalp, 2 = outer skull, 3 = inner skull
Default value is 1.

--gdipoli

List the surfaces in the format required by Thom Oostendorp's *gdipoli* program. This is also the default input format for *mne_surf2bem*.

--meters

List the surface coordinates in meters instead of millimeters.

--surf

Write the output in the binary FreeSurfer format.

--xfit

Write a file compatible with *xfit*. This is the same effect as using the options **--gdipoli** and **--meters** together.

9.5 Converting an ncov covariance matrix file to fif

The *ncov* file format was used to store the noise-covariance matrix file. The MNE software requires that the covariance matrix files are in *fif* format. The utility *mne_convert_ncov* converts *ncov* files to *fif* format.

The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--ncov <name>

The *ncov* file to be converted.

--meas <name>

A *fif* format measurement file used to assign channel names to the noise-covariance matrix elements. This file should have precisely the same channel order within MEG and EEG as the *ncov* file. Typically, both the *ncov* file and the measurement file are created by the now mature off-line averager, *meg_average*.

9.6 Converting a lisp covariance matrix to fif

The utility `mne_convert_lspcov` converts a LISP-format noise-covariance file, produced by the Neuromag signal processor, graph into fif format.

The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--lspcov <name>

The LISP noise-covariance matrix file to be converted.

--meas <name>

A fif format measurement file used to assign channel names to the noise-covariance matrix elements. This file should have precisely the same channel order within MEG and EEG as the LISP-format covariance matrix file.

--out <name>

The name of a fif format output file. The file name should end with `-cov.fif.text` format output file. No information about the channel names is included. The covariance matrix file is listed row by row. This file can be loaded to MATLAB, for example

--outasc <name>

The name of a text format output file. No information about the channel names is included. The covariance matrix file is listed row by row. This file can be loaded to MATLAB, for example

9.7 The MNE data file conversion tool

This utility, called `mne_convert_mne_data`, allows the conversion of various fif files related to the MNE computations to other formats. The two principal purposes of this utility are to facilitate development of new analysis approaches with Matlab and conversion of the forward model and noise covariance matrix data into evoked-response type fif files, which can be accessed and displayed with the Neuromag source modelling software.

Tip: Most of the functions of `mne_convert_mne_data` are now covered by the MNE Matlab toolbox covered in Chapter 10. This toolbox is recommended to avoid creating unnecessary additional files.

9.7.1 Command-line options

The command-line options recognize by *mne_convert_mne_data* are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--fwd <name>

Specify the name of the forward solution file to be converted. Channels specified with the **--bad** option will be excluded from the file.

--fixed

Convert the forward solution to the fixed-orientation mode before outputting the converted file. With this option only the field patterns corresponding to a dipole aligned with the estimated cortex surface normal are output.

--surfsrc

When outputting a free-orientation forward model (three orthogonal dipole components present) rotate the dipole coordinate system at each source node so that the two tangential dipole components are output first, followed by the field corresponding to the dipole aligned with the estimated cortex surface normal. The orientation of the first two dipole components in the tangential plane is arbitrarily selected to create an orthogonal coordinate system.

--noiseonly

When creating a ‘measurement’ fif file, do not output a forward model file, just the noise-covariance matrix.

--senscov <name>

Specifies the fif file containing a sensor covariance matrix to be included with the output. If no other input files are specified only the covariance matrix is output

--srccov <name>

Specifies the fif file containing the source covariance matrix to be included with the output. Only diagonal source covariance files can be handled at the moment.

--bad <name>

Specifies the name of the file containing the names of the channels to be omitted, one channel name per line. This does not affect the output of the inverse operator since the channels have been already selected when the file was created.

--fif

Output the forward model and the noise-covariance matrix into 'measurement' fif files. The forward model files are tagged with `<modalities>-meas-fwd.fif` and the noise-covariance matrix files with `<modalities>-meas-cov.fif`. Here, `modalities` is `-meg` if MEG is included, `-eeg` if EEG is included, and `-meg-eeg` if both types of signals are present. The inclusion of modalities is controlled by the `--meg` and `--eeg` options.

--mat

Output the data into MATLAB mat files. This is the default. The forward model files are tagged with `<modalities>-fwd.mat` forward model and noise-covariance matrix output, with `-inv.mat` for inverse operator output, and with `-inv-meas.mat` for combined inverse operator and measurement data output, respectively. The meaning of `<modalities>` is the same as in the fif output, described above.

--tag <name>

By default, all variables in the matlab output files start with `MNE_`. This option allows to change this prefix to `<name>_`.

--meg

Include MEG channels from the forward solution and noise-covariance matrix.

--eeg

Include EEG channels from the forward solution and noise-covariance matrix.

--inv <name>

Output the inverse operator data from the specified file into a mat file. The source and noise covariance matrices as well as active channels have been previously selected when the inverse operator was created with `mne_inverse_operator`. Thus the options `--meg`, `--eeg`, `--senscov`, `--srccov`, `--noiseonly`, and `--bad` do not affect the output of the inverse operator.

--meas <name>

Specifies the file containing measurement data to be output together with the inverse operator. The channels corresponding to the inverse operator are automatically selected from the file if `--inv` option is present. Otherwise, the channel selection given with `--sel` option will be taken into account.

--set <number>

Select the data set to be output from the measurement file.

--bmin <value/ms>

Specifies the baseline minimum value setting for the measurement signal output.

--bmax <value/ms>

Specifies the baseline maximum value setting for the measurement signal output.

Note: The `--tmin` and `--tmax` options which existed in previous versions of *mne_converted_mne_data* have been removed. If output of measurement data is requested, the entire averaged epoch is now included.

9.7.2 Guide to combining options

The combination of options is quite complicated. The Table 9.1 should be helpful to determine the combination of options appropriate for your needs.

Desired output	Format	Required options	Optional options
forward model	fif	--fwd <name> --out <name> --meg and/or --eeg --fif	--bad <name> --surfsrc
forward model	mat	--fwd <name> --out <name> --meg and/or --eeg	--bad <name> --surfsrc
forward model and sensor covariance	mat	--fwd <name> --out <name> --senscov <name> --meg and/or --eeg	--bad <name> --surfsrc
sensor covariance	fif	--fwd <name> --out <name> --senscov <name> --noiseonly --fif --meg and/or --eeg	--bad <name>
sensor covariance	mat	--senscov <name> --out <name>	--bad <name>
sensor covariance eigenvalues	text	--senscov <name> --out <name> --eig	--bad <name>

Table 9.1 Guide to combining *mne_convert_mne_data* options.

Desired output	Format	Required options	Optional options
evoked MEG/EEG data	mat	--meas <name> --out <name>	--sel <name> --set <number>
evoked MEG/EEG data forward model	mat	--meas <name> --fwd <name> --out <name>	--bad <name> --set <number>
inverse operator data	mat	--inv <name> --out <name>	
inverse operator data evoked MEG/EEG data	mat	--inv <name> --meas <name> --out <name>	

Table 9.1 Guide to combining *mne_convert_mne_data* options.

9.7.3 Matlab data structures

The Matlab output provided by *mne_convert_mne_data* is organized in structures, listed in Table 9.2. The fields occurring in these structures are listed in Table 9.3.

The symbols employed in variable size descriptions are:

<i>nloc</i>	Number of source locations
<i>nsource</i>	Number of sources. For fixed orientation sources $nsource = nloc$ whereas $nsource = 3*nloc$ for free orientation sources
<i>nchan</i>	Number of measurement channels.

ntime Number of time points in the measurement data.

Structure	Contents
<code><tag>_meas</code>	Measured data
<code><tag>_inv</code>	The inverse operator decomposition
<code><tag>_fwd</code>	The forward solution
<code><tag>_noise</code>	A standalone noise-covariance matrix

Table 9.2 Matlab structures produced by *mne_convert_mne_data*. The prefix given with the `--tag` option is indicated `<tag>`, see Section 9.7.1. Its default value is MNE.

Variable	Size	Description
fwd	nsource x nchan	The forward solution, one source on each row. For free orientation sources, the fields of the three orthogonal dipoles for each location are listed consecutively.
names ch_names	nchan (string)	String array containing the names of the channels included
ch_types	nchan x 2	The column lists the types of the channels (1 = MEG, 2 = EEG). The second column lists the coil types, see Tables 5.1 and 5.2. For EEG electrodes, this value equals one.
ch_pos	nchan x 3	The location information for each channel. The first three values specify the origin of the coil coordinate system or the location of the electrode. For MEG channels, the following nine number specify the <i>x</i> , <i>y</i> , and <i>z</i> -direction unit vectors of the coil coordinate system. For EEG electrodes the first unit vector specifies the location of the reference electrode. If the reference is not specified this value is all zeroes. The remaining unit vectors are irrelevant for EEG electrodes.
ch_lognos	nchan x 1	Logical channel numbers as listed in the fiff file

Table 9.3 The fields of Matlab structures.

Variable	Size	Description
ch_units	nchan x 2	Units and unit multipliers as listed in the fif file. The unit of the data is listed in the first column (T = 112, T/m = 201, V = 107). At present, the second column will be always zero, <i>i.e.</i> , no unit multiplier.
ch_cals	nchan x 2	Even if the data comes from the conversion already calibrated, the original calibration factors are included. The first column is the <i>range</i> member of the fif data structures and while the second is the <i>cal</i> member. To get calibrated values in the units given in ch_units from the raw data, the data must be multiplied with the product of <i>range</i> and <i>cal</i> .
sfreq	1	The sampling frequency in Hz.
lowpass	1	Lowpass filter frequency (Hz)
highpass	1	Highpass filter frequency (Hz)
source_loc	nloc x 3	The source locations given in the coordinate frame indicated by coord_frame
source_ori	nsource x 3	The source orientations
source_selection	nsource x 2	Indication of the sources selected from the complete source spaces. Each row contains the number of the source in the complete source space (starting with 0) and the source space number (1 or 2). These numbers refer to the order the two hemispheres where listed when mne_make_source_space was invoked. mne_setup_source_space lists the left hemisphere first.
coord_frame	string	Name of the coordinate frame employed in the forward calculations. Possible values are 'head' and 'mri'.
mri_head_trans	4 x 4	The coordinate frame transformation from mri the MEG 'head' coordinates.
meg_head_trans	4 x 4	The coordinate frame transformation from the MEG device coordinates to the MEG head coordinates
noise_cov	nchan x nchan	The noise covariance matrix
source_cov	nsource	The elements of the diagonal source covariance matrix.

Table 9.3 The fields of Matlab structures.

Variable	Size	Description
sing	nchan	The singular values of $\mathbf{A} = \mathbf{C}_0^{-1/2} \mathbf{G} \mathbf{R}^C = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T$ with \mathbf{R} selected so that $\text{trace}(\mathbf{A} \mathbf{A}^T) / \text{trace}(\mathbf{I}) = 1$ as discussed in Section 6.2.3
eigen_fields	nchan x nchan	The rows of this matrix are the left singular vectors of \mathbf{A} , i.e., the columns of \mathbf{U} , see above.
eigen_leads	nchan x nsource	The rows of this matrix are the right singular vectors of \mathbf{A} , i.e., the columns of \mathbf{V} , see above.
noise_eigenval	nchan	In terms of Section 6.2.3, eigenvalues of \mathbf{C}_0 , i.e., not scaled with number of averages.
noise_eigenvect	nchan	Eigenvectors of the noise covariance matrix. In terms of Section 6.2.3, \mathbf{U}_C^T .
data	nchan x ntime	The measured data. One row contains the data at one time point.
times	ntime	The time points in the above matrix in seconds
nave	1	Number of averages as listed in the data file.
meas_times	ntime	The time points in seconds.

Table 9.3 The fields of Matlab structures.

9.7.4 Example matlab programs

The directory `$MNE_ROOT/setup/mne/templates` will eventually contain some matlab example programs to use the inverse-operator data produced by `mne_convert_mne_data`.

9.8 Converting surface data to Matlab format

The utility `mne_convert_surface` converts text or FreeSurfer format surface files into Matlab mat files.

9.8.1 command-line options

`mne_convert_surface` accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--tri <name>

Specifies a text format input file. The format of this file is described in Section 5.4.3.

--surf <name>

Specifies a FreeSurfer format input file.

--tag <tag>

By default, all Matlab variables included in the output file start with MNE_. This option changes the prefix to <tag>_.

--mat <name>

Specifies the output file.

9.8.2 Matlab data structure

The Matlab output contains a structure called <tag>_surf. The default value of <tag> is MNE. It can be changed with the `--tag` option, see above. The fields of this structure are listed in Table 9.4.

The symbols employed in variable size descriptions are:

nvert Number of vertices on the surface
ntri Number of triangles on the surface

Variable	Size	Description
vert	nvert x 3	The vertex locations.
vert_norm	nvert x 3	The estimated vertex normals.
tris	ntri x 3	List of triangles. Each row is a triplet of indices to the vertex list.

Table 9.4 The Matlab variables for surfaces.

9.8.3 Conversion of both hemispheres

The convenience script *mne_surface2mat* facilitates the conversion of both hemispheres of a given surface to Matlab format. Its command-line options are:

--subject <subject>

Defines the name of the subject. If the environment variable SUBJECT is set correctly this option is not required.

--surface <name>

Name of one of the surfaces in the surf directory of the subject.

The resulting output files are `lh.<name>.mat` and `rh.<name>.mat`. The variables in these files will be prefixed by `<name>_lh` and `<name>_rh`, respectively.

9.9 Converting raw data to Matlab format

The utility `mne_raw2mat` converts all or selected channels from a raw data file to a Matlab mat file. In addition, this utility can provide information about the raw data file so that the raw data can be read directly from the original fif file using Matlab file I/O routines.

Tip: The MNE Matlab toolbox described in Chapter 10 provides direct access to raw fif files without a need for conversion to mat file format first.

9.9.1 Command-line options

`mne_raw2mat` accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--raw <name>

Specifies the name of the raw data fif file to convert.

--mat <name>

Specifies the name of the destination Matlab file.

--info

With this option present, only information about the raw data file is included. The raw data itself is omitted.

--sel <name>

Specifies a text file which contains the names of the channels to include in the output file, one channel name per line. If the `--info` option is specified, `--sel` does not have any effect.

--tag <tag>

By default, all Matlab variables included in the output file start with MNE_. This option changes the prefix to <tag>_.

9.9.2 Matlab data structures

The Matlab files output by *mne_raw2mat* can contain two data structures, <tag>_raw and <tag>_raw_info. If --info option is specified, the file contains the latter structure only.

The <tag>_raw structure contains only one field, *data* which is a matrix containing the raw data. Each row of this matrix constitutes the data from one channel in the original file. The data type of this matrix is the same of the original data (2-byte signed integer, 4-byte signed integer, or single-precision float).

The fields of the <tag>_raw_info structure are listed in Table 9.5. Further explanation of the *bufs* field is provided in Table 9.6.

Variable	Size	Description
orig_file	string	The name of the original fif file specified with the --raw option.
nchan	1	Number of channels.
nsamp	1	Total number of samples
bufs	nbuf x 4	This field is present if --info option was specified on the command line. For details, see Table 9.6.
sfreq	1	The sampling frequency in Hz.
lowpass	1	Lowpass filter frequency (Hz)
highpass	1	Highpass filter frequency (Hz)
ch_names	nchan (string)	String array containing the names of the channels included
ch_types	nchan x 2	The column lists the types of the channels (1 = MEG, 2 = EEG). The second column lists the coil types, see Tables 5.1 and 5.2. For EEG electrodes, this value equals one.
ch_lognos	nchan x 1	Logical channel numbers as listed in the fif file

Table 9.5 The fields of the raw data info structure.

Variable	Size	Description
ch_units	nchan x 2	Units and unit multipliers as listed in the fif file. The unit of the data is listed in the first column (T = 112, T/m = 201, V = 107). At present, the second column will be always zero, <i>i.e.</i> , no unit multiplier.
ch_pos	nchan x 12	The location information for each channel. The first three values specify the origin of the coil coordinate system or the location of the electrode. For MEG channels, the following nine number specify the <i>x</i> , <i>y</i> , and <i>z</i> -direction unit vectors of the coil coordinate system. For EEG electrodes the first vector after the electrode location specifies the location of the reference electrode. If the reference is not specified this value is all zeroes. The remaining unit vectors are irrelevant for EEG electrodes.
ch_cals	nchan x 2	The raw data output by <i>mne_raw2mat</i> is uncalibrated. The first column is the <i>range</i> member of the fif data structures and while the second is the <i>cal</i> member. To get calibrated data values in the units given in <i>ch_units</i> from the raw data, the data must be multiplied with the product of <i>range</i> and <i>cal</i> .
meg_head_trans	4 x 4	The coordinate frame transformation from the MEG device coordinates to the MEG head coordinates.

Table 9.5 The fields of the raw data info structure.

Column	Contents
1	The raw data type (2 or 16 = 2-byte signed integer, 3 = 4-byte signed integer, 4 = single-precision float). All data in the fif file are written in the big-endian byte order. The raw data are stored sample by sample.
2	Byte location of this buffer in the original fif file.
3	First sample of this buffer. Since raw data storing can be switched on and off during the acquisition, there might be gaps between the end of one buffer and the beginning of the next.
4	Number of samples in the buffer.

Table 9.6 The *bufs* member of the raw data info structure.

9.10 Converting epochs to Matlab format

The utility *mne_epochs2mat* converts epoch data including all or selected channels from a raw data file to a simple binary file with an associated description file in Matlab mat file format. With help of the description file, a matlab program can easily read the epoch data from the simple binary file. Signal space projection and bandpass filtering can be optionally applied to the raw data prior to saving the epochs.

9.10.1 Command-line options

mne_epochs2mat accepts the following command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--raw <name>

Specifies the name of the raw data fif file to use as input.

--mat <name>

Specifies the name of the destination file. Anything following the last period in the file name will be removed before composing the output file name. The binary epoch file will be called *<trimmed name>.epochs* and the corresponding Matlab description file will be *<trimmed name>_desc.mat*.

--tag <tag>

By default, all Matlab variables included in the description file start with MNE_. This option changes the prefix to *<tag>_*.

--events <name>

The file containing the event definitions. This can be a text or fif format file produced by *mne_process_raw* or *mne_browse_raw*, see Section 4.10. With help of this file it is possible to select virtually any data segment from the raw data file. If this option is missing, the digital trigger channel in the raw data file or a fif format event file produced automatically by *mne_process_raw* or *mne_browse_raw* is consulted for event information.

--event <name>

Event number identifying the epochs of interest.

--tmin <time/ms>

The starting point of the epoch with respect to the event of interest.

--tmax <time/ms>

The endpoint of the epoch with respect to the event of interest.

--sel <name>

Specifies a text file which contains the names of the channels to include in the output file, one channel name per line. If the **--inv** option is specified, **--sel** is ignored. If neither **--inv** nor **--sel** is present, all MEG and EEG channels are included. The digital trigger channel can be included with the **--includetrig** option, described below.

--inv <name>

Specifies an inverse operator, which will be employed in two ways. First, the channels included to output will be those included in the inverse operator. Second, any signal-space projection operator present in the inverse operator file will be applied to the data. This option cancels the effect of **--sel** and **--proj** options.

--digtrig <name>

Name of the digital trigger channel. The default value is 'STI 014'. Underscores in the channel name will be replaced by spaces.

--includetrig

Add the digital trigger channel to the list of channels to output. This option should not be used if the trigger channel is already included in the selection specified with the **--sel** option.

--filtersize <size>

Adjust the length of the FFT to be applied in filtering. The number will be rounded up to the next power of two. If the size is N , the corresponding length of time is N/f_s , where f_s is the sampling frequency of your data. The filtering procedure includes overlapping tapers of length $N/2$ so that the total FFT length will actually be $2N$. The default value is 4096.

--highpass <value/Hz>

Highpass filter frequency limit. If this is too low with respect to the selected FFT length and data file sampling frequency, the data will not be highpass filtered. You can experiment with the interactive version to find the lowest applicable filter for your data. This value can be adjusted in the interactive version of the program. The default is 0, i.e., no highpass filter in effect.

--highpassw <value/Hz>

The width of the transition band of the highpass filter. The default is 6 frequency bins, where one bin is $f_s/(2N)$.

--lowpass <value/Hz>

Lowpass filter frequency limit. This value can be adjusted in the interactive version of the program. The default is 40 Hz.

--lowpassw <value/Hz>

The width of the transition band of the lowpass filter. This value can be adjusted in the interactive version of the program. The default is 5 Hz.

--filteroff

Do not filter the data.

--proj <name>

Include signal-space projection (SSP) information from this file. If the `--inv` option is present, `--proj` has no effect.

Note: Baseline has not been subtracted from the epochs. This has to be done in subsequent processing with Matlab if so desired.

9.10.2 The binary epoch data file

mne_epochs2mat saves the the epoch data extracted from the raw data file as a simple binary file. The data are stored as big-endian single-precision floating point numbers. Assuming that each of the total of p epochs contains n channels and m time points, the data s_{jkl} are ordered as

$$s_{111} \cdots s_{1n1} s_{211} \cdots s_{mn1} \cdots s_{mnp},$$

where the first index stands for the time point, the second for the channel, and the third for the epoch number, respectively. The data are uncalibrated, i.e., the calibration factors present in the Matlab description file have to be applied to get to physical units as described below.

Note: The maximum size of an epoch data file is 2 Gbytes, i.e., 0.5 Gsamples.

9.10.3 Matlab data structures

The Matlab description files output by *mne_epochs2mat* contain a data structure `<tag>_epoch_info`. The fields of this structure are listed in Table 9.5. Further explanation of the *epochs* member is provided in Table 9.8.

Variable	Size	Description
orig_file	string	The name of the original fif file specified with the <code>--raw</code> option.

Table 9.7 The fields of the raw data info structure.

Variable	Size	Description
epoch_file	string	The name of the epoch data file produced by <i>mne_epocs2mat</i> .
nchan	1	Number of channels.
nepoch	1	Total number of epochs.
epochs	nepoch x 5	Description of the content of the epoch data file, see Table 9.8.
sfreq	1	The sampling frequency in Hz.
lowpass	1	Lowpass filter frequency (Hz)
highpass	1	Highpass filter frequency (Hz)
ch_names	nchan (string)	String array containing the names of the channels included
ch_types	nchan x 2	The column lists the types of the channels (1 = MEG, 2 = EEG). The second column lists the coil types, see Tables 5.1 and 5.2. For EEG electrodes, this value equals one.
ch_lognos	nchan x 1	Logical channel numbers as listed in the fif file
ch_units	nchan x 2	Units and unit multipliers as listed in the fif file. The unit of the data is listed in the first column (T = 112, T/m = 201, V = 107). At present, the second column will be always zero, <i>i.e.</i> , no unit multiplier.
ch_pos	nchan x 12	The location information for each channel. The first three values specify the origin of the coil coordinate system or the location of the electrode. For MEG channels, the following nine number specify the <i>x</i> , <i>y</i> , and <i>z</i> -direction unit vectors of the coil coordinate system. For EEG electrodes the first vector after the electrode location specifies the location of the reference electrode. If the reference is not specified this value is all zeroes. The remaining unit vectors are irrelevant for EEG electrodes.
ch_cals	nchan x 2	The raw data output by <i>mne_raw2mat</i> is uncalibrated. The first column is the <i>range</i> member of the fif data structures and while the second is the <i>cal</i> member. To get calibrated data values in the units given in <i>ch_units</i> from the raw data, the data must be multiplied with the product of <i>range</i> and <i>cal</i> .

Table 9.7 The fields of the raw data info structure.

Variable	Size	Description
meg_head_trans	4 x 4	The coordinate frame transformation from the MEG device coordinates to the MEG head coordinates.

Table 9.7 The fields of the raw data info structure.

Column	Contents
1	The raw data type (2 or 16 = 2-byte signed integer, 3 = 4-byte signed integer, 4 = single-precision float). The epoch data are written using the big-endian byte order. The data are stored sample by sample.
2	Byte location of this epoch in the binary epoch file.
3	First sample of this epoch in the original raw data file.
4	First sample of the epoch with respect to the event.
5	Number of samples in the epoch.

Table 9.8 The *epochs* member of the raw data info structure.

Tip: For source modelling purposes, it is recommended that the MNE Matlab toolbox, see Chapter 10 is employed to read the measurement info instead of using the channel information in the raw data info structure described in Table 9.7.

CHAPTER 10 The Matlab toolbox

10.1 Overview

The MNE software contains a collection Matlab m-files to facilitate interfacing with binary file formats of the MNE software. The toolbox is located at `$MNE_ROOT/matlab/toolbox`. The names of the MNE Matlab toolbox functions begin either with *mne_* or with *fiff_*. To gain access the MNE Matlab toolbox, proceed as follows:

1. Copy the file `$MNE_ROOT/matlab/toolbox/mne_setup_toolbox.m` to a directory which is already on your Matlab search path, for example, `$HOME/matlab`,
2. Add the line `mne_setup_toolbox` to your `$HOME/matlab/startup.m` file.

A summary of the available routines is provided in Tables 10.1 – 10.13. The directory `$MNE_ROOT/matlab/examples` contains demonstrations of the use of the library. These routines are listed in Table 10.13.

Important: The MNE Matlab Toolbox is compatible with Matlab versions 7.0 or later.

Function	Purpose
<code>fiff_find_evoked</code>	Find all evoked data sets from a file.
<code>fiff_read_bad_channels</code>	Read the bad channel list.
<code>fiff_read_ctf_comp</code>	Read CTF software gradient compensation data.
<code>fiff_read_evoked</code>	Read evoked-response data.
<code>fiff_read_evoked_all</code>	Read all evoked-response data from a file.
<code>fiff_read_meas_info</code>	Read measurement information.
<code>fiff_read_mri</code>	Read an MRI description file.

Table 10.1 High-level reading routines.

Function	Purpose
fiff_read_proj	Read signal-space projection data.
fiff_read_raw_segment	Read a segment of raw data with time limits are specified in samples.
fiff_read_raw_segment_times	Read a segment of raw data with time limits specified in seconds.
fiff_setup_read_raw	Set up data structures before using fiff_read_raw_segment or fiff_read_raw_segment_times

Table 10.1 High-level reading routines.

Function	Purpose
fiff_pick_channels	Create a selector to pick desired channels from data according to include and exclude lists.
fiff_pick_channels_evoked	Pick desired channels from evoked-response data according to include and exclude lists
fiff_pick_info	Modify measurement info to include only selected channels.
fiff_pick_types	Create a selector to pick desired channels from data according to channel types (MEG, EEG, STIM) in combination with include and exclude lists.
fiff_pick_types_evoked	Pick desired channels from evoked-response data according to channel types (MEG, EEG, STIM) in combination with include and exclude lists.

Table 10.2 Channel selection utilities.

Function	Purpose
fiff_invert_transform	Invert a coordinate transformation structure.
fiff_reset_ch_pos	Reset channel position transformation to the default values present in the file.
fiff_transform_eeg_chs	Transform electrode positions to another coordinate frame.
fiff_transform_meg_chs	Transform MEG coil coordinate transformations to bring the integration points to another coordinate frame.

Table 10.3 Coordinate transformation utilities.

Function	Purpose
fiff_define_constants	Define a structure which contains the constant relevant to fif files.
fiff_dir_tree_find	Find nodes of a given type in a directory tree structure.
fiff_list_dir_tree	List a directory tree structure.
fiff_make_dir_tree	Create a directory tree structure.
fiff_open	Open a fif file and create the directory tree structure.
fiff_read_named_matrix	Read a named matrix from a fif file.
fiff_read_tag	Read one tag from a fif file.
fiff_read_tag_info	Read the info of one tag from a fif file.
fiff_split_name_list	Split a colon-separated list of names into a cell array of strings.

Table 10.4 Basic reading routines.

Function	Purpose
fiff_end_block	Write a FIFF_END_BLOCK tag.
fiff_end_file	Write the standard closing.
fiff_start_block	Write a FIFF_START_BLOCK tag.
fiff_start_file	Write the appropriate beginning of a file.
fiff_write_ch_info	Write a channel information structure.
fiff_write_coord_trans	Write a coordinate transformation structure.
fiff_write_ctf_comp	Write CTF compensation data.
fiff_write_dig_point	Write one digitizer data point.
fiff_write_complex	Write single-precision complex numbers.
fiff_write_complex_matrix	Write a single-precision complex matrix.
fiff_write_double	Write double-precision floats.
fiff_write_double_complex	Write double-precision complex numbers.
fiff_write_double_complex_matrix	Write a double-precision complex matrix
fiff_write_double_matrix	Write a double-precision matrix
fiff_write_evoked	Write an evoked-reponse data file.
fiff_write_float	Write single-precision floats.

Table 10.5 Writing routines.

Function	Purpose
fiff_write_float_matrix	Write a single-precision matrix.
fiff_write_id	Write an id tag.
fiff_write_int	Write 32-bit integers.
fiff_write_name_list	Write a name list.
fiff_write_named_matrix	Write a named matrix.
fiff_write_proj	Write SSP data.
fiff_write_short	Write 16-bit integers.
fiff_write_string	Write a string.

Table 10.5 Writing routines.

Function	Purpose
mne_add_coil_defs	Add coil definitions to an array of channel information structures.
mne_load_coil_def	Load a coil definition file.

Table 10.6 Coil definition utilities.

Function	Purpose
mne_compensate_to	Apply or remove CTF software gradient compensation from evoked-response data.
mne_get_current_comp	Get the state of software gradient compensation from measurement info.
mne_make_compensator	Make a compensation matrix which switches the status of CTF software gradient compensation from one state to another.
mne_make_projector	Create a signal-space projection operator with the projection item definitions and cell arrays of channel names and bad channel names as input.
mne_make_projector_info	Like mne_make_projector but uses the measurement info structure as input.
mne_set_current_comp	Change the information about the compensation status in measurement info.

Table 10.7 Routines for software gradient compensation and signal-space projection.

Function	Purpose
<code>mne_pick_channels_cov</code>	Pick desired channels from a sensor covariance matrix.
<code>mne_read_bem_surfaces</code>	Read triangular tessellations of surfaces for boundary-element models
<code>mne_read_cov</code>	Read a covariance matrix.
<code>mne_read_epoch</code>	Read an epoch of data from the output file of <i>mne_epochs2mat</i> .
<code>mne_read_forward_solution</code>	Read a forward solution from a fif file.
<code>mne_read_inverse_operator</code>	Read an inverse operator from a fif file.
<code>mne_read_noise_cov</code>	Read a noise-covariance matrix from a fif file.
<code>mne_read_source_spaces</code>	Read source space information from a fif file.

Table 10.8 High-level routines for reading MNE data files.

Function	Purpose
<code>mne_write_cov</code>	Write a covariance matrix to an open file.
<code>mne_write_cov_file</code>	Write a complete file containing just a covariance matrix.
<code>mne_write_inverse_sol_stc</code>	Write stc files containing inverse solution on the cortical surface.

Table 10.9 High-level routines for writing MNE data files.

Function	Purpose
<code>mne_read_stc_file</code>	Read data from one stc file.
<code>mne_read_w_file</code>	Read data from one w file.
<code>mne_write_stc_file</code>	Write a new stc file.
<code>mne_write_w_file</code>	Write a new w file.

Table 10.10 Routines for reading stc and w files.

Function	Purpose
<code>mne_read_curvature</code>	Read a curvature file.
<code>mne_read_surface</code>	Read one surface, return the vertex locations and triangulation info.

Table 10.11 Routines for reading FreeSurfer surfaces.

Function	Purpose
<code>mne_read_surfaces</code>	Read surfaces corresponding to one or both hemispheres. Optionally read curvature information and add derived surface data.

Table 10.11 Routines for reading FreeSurfer surfaces.

Function	Purpose
<code>mne_block_diag</code>	Create a sparse block-diagonal matrix out of a vector.
<code>mne_combine_xyz</code>	Calculate the square sum of the three Cartesian components of several vectors listed in one row or column vector.
<code>mne_file_name</code>	Compose a file name relative to <code>\$MNE_ROOT</code>
<code>mne_find_channel</code>	Find a channel by name from measurement info.
<code>mne_find_source_space_hemi</code>	Determine whether a given source space belongs to the left or right hemisphere.
<code>mne_fread3</code>	Read a three-byte integer.
<code>mne_fwrite3</code>	Write a three-byte integer.
<code>mne_omit_first_line</code>	Omit first line from a multi-line message. This routine is useful for formatting error messages.
<code>mne_prepare_inverse_operator</code>	Prepare inverse operator data for calculating L2 minimum-norm solutions and dSPM.
<code>mne_setup_toolbox</code>	Set up the MNE Matlab toolbox.
<code>mne_transpose_named_matrix</code>	Create a transpose of a named matrix.
<code>mne_transform_source_space_to</code>	Transform source space data to another coordinate frame.

Table 10.12 Utility functions.

Function	Purpose
<code>mne_ex_average_epochs</code>	Example of averaging epoch data produced by <i>mne_epochs2mat</i> , see Section 9.10.
<code>mne_ex_cancel_noise</code>	Example of noise cancellation procedures.
<code>mne_ex_compute_inverse</code>	Example of computing a L2 minimum-norm estimate or a dSPM solution
<code>mne_ex_data_sets</code>	Example of listing evoked-response data sets.

Table 10.13 Examples demonstrating the use of the toolbox.

Function	Purpose
<code>mne_ex_evoked_grad_amp</code>	Compute tangential gradient amplitudes from planar gradiometer data.
<code>mne_ex_read_evoked</code>	Example of reading evoked-response data.
<code>mne_ex_read_raw</code>	Example of reading raw data.
<code>mne_ex_write_evoked</code>	Example of writing evoked-response data.

Table 10.13 Examples demonstrating the use of the toolbox.

10.2 Some data structures

The MNE Matlab toolbox relies heavily on structures to organize the data. This section gives detailed information about fields of the essential data structures employed in the MNE Matlab toolbox. In the structure definitions, data types referring to other MNE Matlab toolbox structures are italicized. In addition, Table 10.14 lists the values of various FIFF constants defined by *fiff_define_constants.m*. The documented structures are:

tag

Contains one tag from the fif file, see Table 10.15.

taginfo

Contains the information about one tag, see Table 10.16.

directory

Contains the tag directory as a tree structure, see Table 10.17.

id

A fif ID, see Table 10.18.

named matrix

Contains a matrix with names for rows and/or columns, see Table 10.19. A named matrix is used to store, *e.g.*, SSP vectors and forward solutions.

trans

A 4 x 4 coordinate-transformation matrix operating on augmented column vectors. Indication of the coordinate frames to which this transformation relates is included, see Table 10.20.

dig

A Polhemus digitizer data point, see Table 10.21.

coildef

The coil definition structure useful for forward calculations and array visualization, see Table 10.22. For more detailed information on coil definitions, see Section 5.6.

ch

Channel information structure, see Table 10.23.

proj

Signal-space projection data, see Table 10.24.

comp

Software gradiometer compensation data, see Table 10.25.

measurement info

Translation of the FIFFB_MEAS_INFO entity, see Table 10.26. This data structure is returned by *fiff_read_meas_info*.

surf

Used to represent triangulated surfaces and cortical source spaces, Table 10.27.

cov

Used for storing covariance matrices, Table 10.28.

fwd

Forward solution data returned by *mne_read_forward_solution*, see Table 10.29.

inv

Inverse operator decomposition data returned by *mne_read_inverse_operator*, see Table 10.30. For more information on inverse operator decomposition, see Section 6.2. For an example on how to compute inverse solution using this data, see the sample routine *mne_ex_compute_inverse*

Name	Value	Description
FIFFV_MEG_CH	1	This is a MEG channel.
FIFFV_REF_MEG_CH	301	This a reference MEG channel, located far away from the head.
FIFFV_EEF_CH	2	This is an EEG channel.
FIFFV_MCG_CH	201	This a MCG channel.
FIFFV_STIM_CH	3	This is a digital trigger channel.
FIFFV_EOG_CH	202	This is an EOG channel.

Table 10.14 FIFF constants.

Name	Value	Description
FIFFV_EMG_CH	302	This is an EMG channel.
FIFFV_ECG_CH	402	This is an ECG channel.
FIFFV_MISC_CH	502	This is a miscellaneous analog channel.
FIFFV_RESP_CH	602	This channel contains respiration monitor output.
FIFFV_COORD_UNKNOWN	0	Unknown coordinate frame
FIFFV_COORD_ISOTRAK	1	The MEG device coordinate frame
FIFFV_COORD_ISOTRAK	2	The Polhemus digitizer coordinate frame (does not appear in data files).
FIFFV_COORD_HPI	3	HPI coil coordinate frame (does not appear in data files)
FIFFV_COORD_HEAD	4	The MEG head coordinate frame (Neuromag convention).
FIFFV_COORD_MRI	5	The MRI coordinate frame
FIFFV_COORD_MRI_SLICE	6	The coordinate frame of a single MRI slice.
FIFFV_COORD_MRI_DISPLAY	7	The preferred coordinate frame for displaying the MRIs (used by MRILab).
FIFFV_COORD_DICOM_DEVICE	8	The DICOM coordinate frame (does not appear in files).
FIFFV_COORD_IMAGING_DEVICE	9	A generic imaging device coordinate frame (does not appear in files).
FIFFV_MNE_COORD_TUFTS_EEG	300	The Tufts EEG data coordinate frame
FIFFV_MNE_COORD_CTF_DEVICE	1001	The CTF device coordinate frame (does not appear in files)
FIFFV_MNE_COORD_CTF_HEAD	1004	The CTF/4D head coordinate frame
FIFFV_ASPECT_AVERAGE	100	Data aspect: average.
FIFFV_ASPECT_STD_ERR	101	Data aspect: standard error of mean.
FIFFV_ASPECT_SINGLE	102	Single epoch.
FIFFV_ASPECT_SUBAVERAGE	103	One subaverage.
FIFFV_ASPECT_ALTAVERAGE	104	One alternating (plus-minus) subaverage.
FIFFV_ASPECT_SAMPLE	105	A sample cut from raw data.

Table 10.14 FIFF constants.

Name	Value	Description
FIFFV_ASPECT_POWER_DENSITY	106	Power density spectrum.
FIFFV_ASPECT_DIPOLE_WAVE	200	The time course of an equivalent current dipole
FIFFV_BEM_SURF_ID_UNKNOWN	-1	Unknown BEM surface
FIFFV_BEM_SURF_ID_BRAIN	1	The inner skull surface
FIFFV_BEM_SURF_ID_SKULL	3	The outer skull surface
FIFFV_BEM_SURF_ID_HEAD	4	The scalp surface
FIFFV_MNE_SURF_LEFT_HEMI	101	Left hemisphere cortical surface
FIFFV_MNE_SURF_RIGHT_HEMI	102	Right hemisphere cortical surface
FIFFV_POINT_CARDINAL	1	Digitization point which is a cardinal landmark aka. fiducial point
FIFFV_POINT_HPI	2	Digitized HPI coil location
FIFFV_POINT_EEG	3	Digitized EEG electrode location
FIFFV_POINT_ECG	3	Digitized ECG electrode location
FIFFV_POINT_EXTRA	4	Additional head surface point
FIFFV_POINT_LPA	1	Identifier for left auricular landmark
FIFFV_POINT_NASION	2	Identifier for nasion
FIFFV_POINT_RPA	3	Identifier for right auricular landmark
FIFFV_MNE_FIXED_ORI	1	Fixed orientation constraint used in the computation of a forward solution.
FIFFV_MNE_FREE_ORI	2	No orientation constraint used in the computation of a forward solution
FIFFV_MNE_MEG	1	Indicates an inverse operator based on MEG only
FIFFV_MNE_EEG	2	Indicates an inverse operator based on EEG only.
FIFFV_MNE_MEG_EEG	3	Indicates an inverse operator based on both MEG and EEG.
FIFFV_MNE_UNKNOWN_COV	0	An unknown covariance matrix
FIFFV_MNE_NOISE_COV	1	Indicates a noise covariance matrix.
FIFFV_MNE_SENSOR_COV	1	Synonym for FIFFV_MNE_NOISE_COV

Table 10.14 FIFF constants.

Name	Value	Description
FIFFV_MNE_SOURCE_COV	2	Indicates a source covariance matrix
FIFFV_MNE_fmRI_PRIOR_COV	3	Indicates a covariance matrix associated with fmRI priors
FIFFV_MNE_SIGNAL_COV	4	Indicates the data (signal + noise) covariance matrix
FIFFV_PROJ_ITEM_NONE	0	The nature of this projection item is unknown
FIFFV_PROJ_ITEM_FIELD	1	This is projection item is a generic field pattern or field patterns.
FIFFV_PROJ_ITEM_DIP_FIX	2	This projection item is the field of one dipole
FIFFV_PROJ_ITEM_DIP_ROT	3	This projection item corresponds to the fields of three or two orthogonal dipoles at some location.
FIFFV_PROJ_ITEM_HOMOGRAD	4	This projection item contains the homogeneous gradient fields as seen by the sensor array.
FIFFV_PROJ_ITEM_HOMOGRFIELD	5	This projection item contains the three homogeneous field components as seen by the sensor array.
FIFFV_MNE_PROJ_ITEM_EEG_AVREF	10	This projection item corresponds to the average EEG reference.

Table 10.14 FIFF constants.

Field	Data type	Description
kind	int32	The kind of the data item.
type	uint32	The data type used to represent the data.
size	int32	Size of the data in bytes.
next	int32	Byte offset of the next tag in the file.
data	various	The data itself.

Table 10.15 The *tag* structure.

Field	Data type	Description
kind	double	The kind of the data item.

Table 10.16 The *taginfo* structure.

Field	Data type	Description
type	double	The data type used to represent the data.
size	double	Size of the data in bytes.
pos	double	Byte offset to this tag in the file.

Table 10.16 The *taginfo* structure.

Field	Data type	Description
block	double	The block id of this directory node.
id	<i>id</i>	The unique identifier of this node.
parent_id	<i>id</i>	The unique identifier of the node this node was derived from.
nent	double	Number of entries in this node.
nchild	double	Number of children to this node.
dir	<i>taginfo</i>	Information about tags in this node.
children	<i>directory</i>	The children of this node.

Table 10.17 The *directory* structure.

Field	Data type	Description
version	int32	The fif file version (major << 16 minor).
machid	int32(2)	Unique identifier of the computer this id was created on.
secs	int32	Time since January 1, 1970 (seconds).
usecs	int32	Time since January 1, 1970 (microseconds).

Table 10.18 The *id* structure.

Field	Data type	Description
nrow	int32	Number of rows.
ncol	int32	Number of columns.
row_names	cell(*)	The names of associated with the rows. This member may be empty.

Table 10.19 The *named matrix* structure.

Field	Data type	Description
col_names	cell(*)	The names of associated with the columns. This member may be empty.
data	various	The matrix data, usually of type single or double.

Table 10.19 The *named matrix* structure.

Field	Data type	Description
from	int32	The source coordinate frame, see Table 10.14. Look for entries starting with FIFFV_COORD or FIFFV_MNE_COORD.
to	int32	The destination coordinate frame.
trans	double(4,4)	The 4-by-4 coordinate transformation matrix. This operates from augmented position column vectors given in <i>from</i> coordinates to give results in <i>to</i> coordinates.

Table 10.20 The *trans* structure.

Field	Data type	Description
kind	int32	The type of digitizing point. Possible values are listed in Table 10.14. Look for entries starting with FIFF_POINT.
ident	int32	Identifier for this point.
r	single(3)	The location of this point.

Table 10.21 The *dig* structure.

Field	Data type	Description
class	double	The coil (or electrode) class
id	double	The coil (or electrode) id
accuracy	double	Representation accuracy
num_points	double	Number of integration points.
size	double	Coil size

Table 10.22 The *coildef* structure. For more detailed information, see Section 5.6.

Field	Data type	Description
baseline	double	Coil baseline
description	char(*)	Coil description
coildefs	double(num_points,7)	Each row contains the integration point weight, followed by location [m] and normal
FV	struct	Contains the faces and vertices which can be used to draw the coil for visualization.

Table 10.22 The *coildef* structure. For more detailed information, see Section 5.6.

Field	Data type	Description
scanno	int32	Scanning order number, starting from 1.
logno	int32	Logical channel number, conventions in the usage of this number vary.
kind	int32	The location of this point.
range	double	The hardware-oriented part of the calibration factor. This should be only applied to the continuous raw data.
cal	double	The calibration factor to bring the channels to physical units.
loc	double(12)	The channel location. The first three numbers indicate the location [m], followed by the three unit vectors of the channel-specific coordinate frame. These data contain the values saved in the fif file and should not be changed. The values are specified in device coordinates for MEG and in head coordinates for EEG channels, respectively.
coil_trans	double(4,4)	Initially, transformation from the the channel coordinates to device coordinates. This transformation is updated by calls to <i>fiff_transform_meg_chs</i> and <i>fiff_transform_eeg_chs</i> .
eeg_loc	double(6)	The location of the EEG electrode in <i>coord_frame</i> coordinates. The first three values contain the location of the electrode [m]. If six values are present, the remaining ones indicate the location of the reference electrode for this channel.
coord_frame	int32	Initially, the coordinate frame is FIFFV_COORD_DEVICE for MEG channels and FIFFV_COORD_HEAD for EEG channels.

Table 10.23 The *ch* structure.

Field	Data type	Description
unit	int32	Unit of measurement. Relevant values are: 201 = T/m, 112 = T, 107 = V, and 202 = Am.
unit_mul	int32	The data are given in <i>units</i> multiplied by $10^{\text{unit_mul}}$. Presently, <i>unit_mul</i> is always zero.
ch_name	char(*)	Name of the channel.
coil_def	<i>coildef</i>	The coil definition structure. This is present only if <i>mne_add_coil_defs</i> has been successfully called.

Table 10.23 The *ch* structure.

Field	Data type	Description
kind	int32	The type of the projection item. Possible values are listed in Table 10.14. Look for entries starting with FIFFV_PROJ_ITEM or FIFFV_MNE_PROJ_ITEM.
active	int32	Is this item active, <i>i.e.</i> , applied or about to be applied to the data.
data	<i>named matrix</i>	The projection vectors. The column names indicate the names of the channels associated to the elements of the vectors.

Table 10.24 The *proj* structure.

Field	Data type	Description
ctfkind	int32	The kind of the compensation as stored in file.
kind	int32	<i>ctfkind</i> mapped into small integer numbers.
save_calibrated	logical	Were the compensation data saved in calibrated form. If this field is false, the matrix will be decalibrated using the fields <i>row_cals</i> and <i>col_cals</i> when the compensation data are saved by the toolbox.
row_cals	double(*)	Calibration factors applied to the rows of the compensation data matrix when the data were read.
col_cals	double(*)	Calibration factors applied to the columns of the compensation data matrix when the data were read.

Table 10.25 The *comp* structure.

Field	Data type	Description
data	<i>named matrix</i>	The compensation data matrix. The <i>row_names</i> list the names of the channels to which this compensation applies and the <i>col_names</i> the compensation channels. For more information, see Section 9.2.4

Table 10.25 The *comp* structure.

Field	Data type	Description
file_id	<i>id</i>	The fif ID of the measurement file.
meas_id	<i>id</i>	The ID assigned to this measurement by the acquisition system or during file conversion.
nchan	int32	Number of channels
sfreq	double	Sampling frequency
highpass	double	Highpass corner frequency [Hz]. Zero indicates a DC recording.
lowpass	double	Lowpass corner frequency [Hz].
chs	<i>ch</i> (nchan)	An array of channel information structures.
ch_names	cell(nchan)	Cell array of channel names.
dev_head_t	<i>trans</i>	The device to head transformation
ctf_head_t	<i>trans</i>	The transformation from 4D/CTF head coordinates to Neuromag head coordinates. This is only present in 4D/CTF data.
dev_ctf_t	<i>trans</i>	The transformation from device coordinates to 4D/CTF head coordinates. This is only present in 4D/CTF data
dig	<i>dig</i> (*)	The Polhemus digitization data in head coordinates.
bads	cell(*)	Bad channel list
projs	<i>proj</i> (*)	SSP operator data.
comps	<i>comp</i> (*)	Software gradient compensation data.

Table 10.26 The *meas info* structure.

Field	Data type	Description
id	int32	The surface ID

Table 10.27 The *surf* structure.

Field	Data type	Description
sigma	double	The electrical conductivity of the compartment bounded by this surface. This field is present in BEM surfaces only.
np	int32	Number of vertices on the surface
ntri	int32	Number of triangles on the surface
coord_frame	int32	Coordinate frame in which the locations and orientations are expressed.
rr	double(np,3)	The vertex locations
nn	double(np,3)	The vertex normals. If derived surface data was not requested, this is empty.
tris	int32(ntri,3)	Vertex numbers of the triangles in counterclockwise order as seen from the outside.
nuse	int32	Number of active vertices, <i>i.e.</i> , vertices included in a decimated source space.
inuse	int32(np)	Which vertices are in use.
verno	int32(nuse)	Indices of the vertices in use.
curv	double(np)	Curvature values at the vertices. If curvature information was not requested, this field is empty or absent.
tri_area	double(ntri)	The triangle areas in m ² . If derived surface data was not requested, this field will be missing.
tri_cent	double(ntri,3)	The triangle centroids. If derived surface data was not requested, this field will be missing.
tri_nn	double(ntri,3)	The triangle normals. If derived surface data was not requested, this field will be missing.
nuse_tri	int32	Number of triangles in use. This is present only if the surface corresponds to a source space created with the <code>-ico</code> option.
use_tris	int32(nuse_tri)	The vertices of the triangles in use in the complete triangulation. This is present only if the surface corresponds to a source space created with the <code>--ico</code> option.
nearest	int32(np)	This field is present only if patch information has been computed for a source space. For each vertex in the triangulation, these values indicate the nearest active source space vertex.

Table 10.27 The *surf* structure.

Field	Data type	Description
nearest_dist	double(np)	This field is present only if patch information has been computed for a source space. For each vertex in the triangulation, these values indicate the distance to the nearest active source space vertex.

Table 10.27 The *surf* structure.

Field	Data type	Description
kind	double	What kind of a covariance matrix (1 = noise covariance, 2 = source covariance).
diag	double	Is this a diagonal matrix.
dim	int32	Dimension of the covariance matrix.
names	cell(*)	Names of the channels associated with the entries (may be empty).
data	double(dim,dim)	The covariance matrix. This a double(dim) vector for a diagonal covariance matrix.
projs	<i>proj</i> (*)	The SSP vectors applied to these data.
bads	cell(*)	Bad channel names.
nfree	int32	Number of data points used to compute this matrix.
eig	double(dim)	The eigenvalues of the covariance matrix. This field may be empty for a diagonal covariance matrix.
eigvec	double(dim,dim)	The eigenvectors of the covariance matrix.

Table 10.28 The *cov* structure.

Field	Data type	Description
source_ori	int32	Has the solution been computed for the current component normal to the cortex only (1) or all three source orientations (2).
coord_frame	int32	Coordinate frame in which the locations and orientations are expressed.
nsource	int32	Total number of source space points.
nchan	int32	Number of channels.
sol	<i>named matrix</i>	The forward solution matrix.

Table 10.29 The *fwd* structure.

Field	Data type	Description
mri_head_t	<i>trans</i>	Transformation from the MRI coordinate frame to the (Neuromag) head coordinate frame.
src	<i>surf</i> (*)	The description of the source spaces
source_rr	double(nsource,3)	The source locations..
source_nn	double(*,3)	The source orientations. Number of rows is either <i>nsource</i> (fixed source orientations) or $3*nsource$ (all source orientations).

Table 10.29 The *fwd* structure.

Field	Data type	Description
methods	int32	Has the solution been computed using MEG data (1), EEG data (2), or both (3).
source_ori	int32	Has the solution been computed for the current component normal to the cortex only (1) or all three source orientations (2)
nsource	int32	Total number of source space points.
nchan	int32	Number of channels.
coord_frame	int32	Coordinate frame in which the locations and orientations are expressed.
source_nn	double(*,3)	The source orientations. Number of rows is either <i>nsource</i> (fixed source orientations) or $3*nsource$ (all source orientations).
sing	double(nchan)	The singular values, <i>i.e.</i> , the diagonal values of Λ , see Section 6.2.5.
eigen_leads	double(*,nchan)	The matrix \mathbf{V} , see Section 6.2.5.
eigen_fields	double(nchan,nchan)	The matrix \mathbf{U}^T , see Section 6.2.5.
noise_cov	<i>cov</i>	The noise covariance matrix \mathbf{C} .
source_cov	<i>cov</i>	The source covariance matrix \mathbf{R} .
src	<i>surf</i> (*)	The description of the source spaces
mri_head_t	<i>trans</i>	Transformation from the MRI coordinate frame to the (Neuromag) head coordinate frame.
nave	double	The number of averages.

Table 10.30 The *inv* structure. Note: The fields *proj*, *whitener*, *reginv*, and *noisenorm* are filled in by the routine *mne_prepare_inverse_operator*.

Field	Data type	Description
projs	<i>proj</i> (*)	The SSP vectors which were active when the decomposition was computed
proj	double(nchan)	The projection operator computed using <i>projs</i> .
whitener		A sparse matrix containing the noise normalization factors. Dimension is either <i>nsource</i> (fixed source orientations) or $3*nsource$ (all source orientations).
reginv	double(nchan)	The diagonal matrix Γ , see Section 6.2.5.
noisenorm	double(*,*)	A sparse matrix containing the noise normalization factors. Dimension is either <i>nsource</i> (fixed source orientations) or $3*nsource$ (all source orientations).

Table 10.30 The *inv* structure. Note: The fields *proj*, *whitener*, *reginv*, and *noisenorm* are filled in by the routine *mne_prepare_inverse_operator*.

10.3 On-line documentation for individual routines

Each of the routines listed in Tables 10.1 – 10.13 has on-line documentation accessible by saying `help <routine name>` in Matlab.

CHAPTER 11 Miscellaneous utilities

11.1 Overview

This Chapter describes various utility programs included with the MNE software. Each utility documentation consists of a brief description of the purpose followed by the specification of command-line options.

11.2 Finding software versions

The utility *mne_list_versions* lists version numbers and compilation dates of all software modules that provide this information. This administration utility is located in `$MNE_ROOT/bin/admin`. The output from *mne_list_versions* or output of individual modules with `--version` option is useful when bugs are reported to the developers of MNE software.

11.3 Listing contents of a fif file

Using the utility *mne_show_fiff* it is possible to display information about the contents of a fif file to the standard output. The command line options for *mne_show_fiff* are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--in <name>

Specifies the fif file whose contents will be listed.

--verbose

Produce a verbose output. The data of most tags is included in the output. This excludes matrices and vectors. Only the first 80 characters of strings are listed unless the `--long` option is present.

--long

List all data from string tags instead of the first 80 characters. This options has no effect unless the **--verbose** option is also present.

--tag <number>

List only tags of this kind. Multiple **--tag** options can be specified to list several different kinds of data.

mne_show_fiff reads the explanations of tag kinds, block kinds, and units from `$MNE_ROOT/setup/mne/fiff_explanations.txt`.

11.4 Data file conditioning utilities

This section contains utilities which can be used to add information or fix existing information in MEG/EEG data fif files. Unless otherwise noted these utilities can be applied to both raw and evoked data files.

11.4.1 Designating bad channels: *mne_mark_bad_channels*

This utility adds or replaces information about unusable (bad) channels. The command line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--bad <filename>

Specify a text file containing the names of the bad channels, one channel name per line. The names of the channels in this file must match those in the data file exactly. If this option is missing, the bad channel information is cleared.

<data file name>

The remaining arguments are taken as data file names to be modified.

11.4.2 Fixing the encoding of the trigger channel: *mne_fix_stim14*

Some earlier versions of the Neuromag acquisition software had a problem with the encoding of the eighth bit on the digital stimulus channel STI 014. This problem has been now fixed. Old data files can be fixed with *mne_fix_stim14*, which takes raw data file names as arguments. *mne_fix_stim14* also changes the calibration of STI 014 to unity. If the

encoding of STI 014 is already correct, running *mne_fix_stim14* will not have any effect on the raw data.

11.4.3 Updating EEG location info: *mne_check_eeg_locations*

Some versions of the Neuromag acquisition software did not copy the EEG channel location information properly from the Polhemus digitizer information data block to the EEG channel information records if the number of EEG channels exceeds 60. The purpose of *mne_check_eeg_locations* is to detect this problem and fix it, if requested. The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--file <name>

Specify the measurement data file to be checked or modified.

--dig <name>

Name of the file containing the Polhemus digitizer information. Default is the data file name.

--fix

By default *mne_check_eeg_locations* only checks for missing EEG locations (locations close to the origin). With **--fix** *mne_check_eeg_locations* reads the Polhemus data from the specified file and copies the EEG electrode location information to the channel information records in the measurement file. There is no harm running *mne_check_eeg_locations* on a data file even if the EEG channel locations were correct in the first place.

11.4.4 Updating magnetometer coil types: *mne_fix_mag_coil_types*

The purpose of *mne_fix_mag_coil_types* is to change coil type 3022 to 3024 in the MEG channel definition records in the data files specified on the command line.

As shown in Tables 5.1 and 5.2, the Neuromag Vectorview systems can contain magnetometers with two different coil sizes (coil types 3022 and 3023 vs. 3024). The systems incorporating coils of type 3024 were introduced last. At some sites (including the Martinos Center) the data files have still defined the magnetometers to be of type 3022 to ensure compatibility with older versions of Neuromag software. In the MNE software as well as in the present version of Neuromag software coil type 3024 is

fully supported. Therefore, it is now safe to upgrade the data files to use the true coil type.

If the `--magnes` option is specified, the 4D Magnes magnetometer coil type (4001) is changed to 4D magnes gradiometer coil type (4002). Use this option *always* and *only if* your Magnes data comes from a system with axial gradiometers instead of magnetometers. The fif converter included with the Magnes system does not assign the gradiometer coil type correctly.

Note: The effect of the difference between the coil sizes of magnetometer types 3022 and 3024 on the current estimates computed by the MNE software is very small. Therefore the use of *mne_fix_mag_coil_types* is not mandatory.

11.4.5 Modifying channel names and types: `mne_rename_channels`

Sometimes it is necessary to change the names types of channels in MEG/EEG data files. Such situations include:

- Designating an EEG as an EOG channel. For example, the EOG channels are not recognized as such in the fif files converted from CTF data files.
- Changing the name of the digital trigger channel of interest to STI 014 so that *mne_browse_raw* and *mne_process_raw* will recognize the correct channel without the need to specify the `--digtrig` option every time the data file is loaded.

The utility *mne_rename_channels* was designed to meet the above needs. It recognizes the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--fif <name>

Specifies the name of the data file to modify.

--alias <name>

Specifies the the text file which contains the modifications to be applied, see below.

Each line in the alias file contains the old name and new name for a channel, separated by a colon. The old name is a name of one of the channels presently in the file and the new name is the name to be assigned to it. The old name must match an existing channel name in the file exactly. The new name may be followed by another colon and a number which is the

channel type to be assigned to this channel. The channel type options are listed in Table 11.1.

Channel type	Corresponding number
MEG	1
MCG	201
EEG	2
EOG	202
EMG	302
ECG	402
MISC	502
STIM	3

Table 11.1 Channel types.

Warning: Do not attempt to designate MEG channels to EEG channels or vice versa. This may result in strange errors during source estimation.

11.4.6 Modifying trigger channel data: `mne_add_triggers`

11.4.6.1 Purpose

The utility `mne_add_triggers` modifies the digital trigger channel (STI 014) in raw data files to include additional transitions. Since the raw data file is modified, it is possible to make irreversible changes. Use this utility with caution. It is recommended that you never run `mne_add_triggers` on an original raw data file.

11.4.6.2 Command line options

`mne_add_triggers` accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--raw <name>

Specifies the raw data file to be modified.

--trg <name>

Specifies the trigger line modification list. This text file should contain two entries per line: the sample number and the trigger number to be added into the file. The number of the first sample in the file is zero. It is recommended that trigger numbers whose binary equivalent has lower eight bits equal to zero are used to avoid conflicts with the ordinary triggers occurring in the file.

--delete

Delete the triggers defined by the trigger file instead of adding them. This enables changing the file to its original state, provided that the trigger file is preserved.

Note: Since *mne_browse_raw* and *mne_process_raw* can employ an event file which effectively adds new trigger instants, *mne_add_triggers* is for the most part obsolete but it has been retained in the MNE software suite for backward compatibility.

11.5 Creating a derivation file

11.5.1 Purpose

In *mne_browse_raw*, channel derivations are defined as linear combinations of real channels existing in the data files. The utility *mne_make_derivations* reads derivation data from a suitably formatted text file and produces a fif file containing the weights of derived channels as a sparse matrix. Two input file formats are accepted:

1. A file containing arithmetic expressions defining the derivations and
2. A file containing a matrix which specifies the weights of the channels in each derivation.

Both of these formats are described in

11.5.2 Command-line options

mne_make_derivations recognizes the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--in <name>

Specifies a measurement file which contains the EEG electrode locations. This file is not modified by *mne_make_eeg_layout*.

--inmat <name>

Specifies the output file where the layout is stored. Suffix `.lout` is recommended for layout files. `mne_analyze` and `mne_browse_raw` look for the custom layout files from the directory `$HOME/.mne/lout`.

--trans

Indicates that the file specified with the `--inmat` option contains a transpose of the derivation matrix.

--thresh <value>

Specifies the threshold between values to be considered zero and non-zero in the input file specified with the `--inmat` option. The default threshold is 10^{-6} .

--out <name>

Specifies output fif file to contain the derivation data. The recommended name of the derivation file has the format `<name>-deriv.fif`.

--list <name>

List the contents of a derivation file to standard output. If this option is missing and `--out` is specified, the content of the output file will be listed once it is complete. If neither `--list` nor `--out` is present, and `--in` or `--inmat` is specified, the interpreted contents of the input file is listed.

11.5.3 Derivation file formats

All lines in the input files starting with the pound sign (#) are considered to be comments. The format of a derivation in a arithmetic input file is:

$$\langle name \rangle = [\langle w_1 \rangle *] \langle name_1 \rangle + [\langle w_2 \rangle *] \langle name_2 \rangle \dots$$

where `<name>` is the name of the derived channel, $name_k$ are the names of the channels comprising the derivation, and w_k are their weights. Note that spaces are necessary between the items. Channel names containing spaces must be put in quotes. For example,

```
EEG-diff = "EEG 003" - "EEG 002"
```

defines a channel `EEG-diff` which is a difference between `EEG 003` and `EEG 002`. Similarly,

```
EEG-der = 3 * "EEG 010" - 2 * "EEG 002"
```

defines a channel which is three times `EEG 010` minus two times `EEG 002`.

The format of a matrix derivation file is:

```
<nrow>
<ncol>
<names of the input channels>
<name1> <weights>
...
```

The combination of the two arithmetic examples, above can be thus represented as:

```
2 3
      "EEG 002" "EEG 003" "EEG 010"
EEG-diff -1      1      0
EEG-der  -2      0      3
```

Before a derivation is accepted to use by *mne_browse_raw*, the following criteria have to be met:

1. All channels to be combined into a single derivation must have identical units of measure.
2. All channels in a single derivation have to be of the same kind, *e.g.*, MEG channels or EEG channels.
3. All channels specified in a derivation have to be present in the currently loaded data set.

The validity check is done when a derivation file is loaded into *mne_browse_raw*, see Section 4.4.1.10.

11.6 Creating a custom EEG layout

11.6.1 Purpose

Both MNE software (*mne_analyze* and *mne_browse_raw*) and Neuromag software (*xplotter* and *xfit*) employ text layout files to create topographical displays of MEG and EEG data. While the MEG channel layout is fixed, the EEG layout varies from experiment to experiment, depending on the number of electrodes used and the electrode cap configuration. The utility *mne_make_eeg_layout* was created to produce custom EEG layout files based on the EEG electrode location information included in the channel description records.

mne_make_eeg_layout uses azimuthal equidistant projection to map the EEG channel locations onto a plane. The mapping consists of the following steps: (1) A sphere is fitted to the electrode locations and the locations are translated by the location of the origin of the best-fitting sphere. (2)

1. A sphere is fitted to the electrode locations and the locations are translated by the location of the origin of the best-fitting sphere.
2. The spherical coordinates (r_k , θ_k , and ϕ_k) corresponding to each translated electrode location are computed.
3. The projected locations $u_k = R\theta_k \cos\phi_k$ and $v_k = R\theta_k \sin\phi_k$ are computed. By default, $R = 20$. This projection radius can be adjusted with the `--prad` option. Increasing or decreasing R makes the spacing between the channel viewports larger or smaller, respectively.
4. A viewport with width 5 and height 4 is placed centered at the projected location.

11.7 Adding topology information to a source space

11.7.1 Purpose

The utility *mne_add_patch_info* uses the surface topology information to add data about cortical patches corresponding to each source space point to a source space file. A new copy of the source space is created with the patch information included.

Note: Depending on the speed of your computer, *mne_add_patch_info* takes 5 - 30 minutes to run.

11.7.2 Command line options

mne_add_patch_info accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--in <name>

The input source space file. The source space files usually end with `-src.fif`.

--out <name>

The output source space file which will contain the patch information. If the file exists it will be overwritten without asking for permission. A recommended naming convention is to add the letter `p` after the source spacing included in the file name. For example, if the input file is `mh-7-src.fif`, a recommended output file name is `mh-7p-src.fif`.

--w <name>

Name of a w file, which will contain the patch area information. Two files will be created: *<name>-lh.w* and *<name>-rh.w*. The numbers in the files are patch areas in mm^2 . The source space vertices are marked with value 150.

11.8 Averaging forward solutions

11.8.1 Purpose

One possibility to make a grand average over several runs of a experiment is to average the data across runs and average the forward solutions accordingly. For this purpose, *mne_average_forward_solutions* computes a weighted average of several forward solutions. This program only averages the MEG forward solutions.

11.8.2 Command line options

mne_add_triggers accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--fwd <name>:[<weight>]

Specifies a forward solution to include. If no weight is specified, 1.0 is assumed. In the averaging process the weights are divided by their sum. For example, if two forward solutions are averaged and their specified weights are 2 and 3, the average is formed with a weight of 2/5 for the first solution and 3/5 for the second one.

--out <name>

Specifies the output file which will contain the averaged forward solution.

11.9 Converting covariance data into an SSP operator

11.9.1 Purpose

The utility *mne_cov2proj* picks eigenvectors from a covariance matrix and outputs them as a signal-space projection (SSP) file.

11.9.2 Command line options

mne_cov2proj accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--cov <name>

The covariance matrix file to be used as a source. The covariance matrix files usually end with `-cov.fif`.

--proj <name>

The output file to contain the projection. It is recommended that the file name ends with `-proj.fif`.

--bad <name>

Specify channels not to be included when an eigenvalue decomposition of the covariance matrix is computed.

--include <val₁>[:<val₂>]

Select an eigenvector or a range of eigenvectors to include. It is recommended that magnetometers, gradiometers, and EEG data are handled separately with help of the `--bad`, `--meg`, `--megmag`, `--meggrad`, and `--eeg` options.

--meg

After loading the covariance matrix, modify it so that only elements corresponding to MEG channels are included.

--eeg

After loading the covariance matrix, modify it so that only elements corresponding to EEG channels are included.

--megmag

After loading the covariance matrix, modify it so that only elements corresponding to MEG magnetometer channels are included.

--meggrad

After loading the covariance matrix, modify it so that only elements corresponding to MEG planar gradiometer channels are included.

Important: The `--megmag` and `--meggrad` employ the Vectorview channel numbering scheme to recognize MEG magnetometers (channel names ending with '1') and planar gradiometers (other channels). Therefore, these options are only meaningful in conjunction with data acquired with a Neuromag Vectorview system.

11.10 Fitting a sphere to a surface

11.10.1 Purpose

The utility *mne_fit_sphere_to_surf* finds the sphere which best fits a given surface.

11.10.2 Command line options

mne_fit_sphere_to_surf accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--bem <name>

A BEM file to use. The names of these files usually end with `bem.fif` or `bem-sol.fif`.

--surf <name>

A FreeSurfer surface file to read. This is an alternative to using a surface from the BEM file.

--scalp

Use the scalp surface instead of the inner skull surface in sphere fitting. If the surface is specified with the `--surf` option, this one is irrelevant.

--mrtrtrans <name>

A file containing a transformation matrix between the MEG head coordinates and MRI coordinates. With this option, the sphere origin will be output in MEG head coordinates. Otherwise the output will be in MRI coordinates.

11.11 Applying signal-space projection to raw data

11.11.1 Purpose

mne_project_raw applies a signal-space projection operator to a raw data file. With the advent of *mne_browse_raw* and *mne_process_raw* this tool is almost obsolete.

Important: This tool is only able to process raw data files with short integer data samples, *i.e.*, not data processed with *graph* and stored as floats. Also, only one projection file can be specified.

11.11.2 Command line options

mne_project_raw accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--raw <name>

Specifies the input file.

--out <name>

Specifies the output file.

--proj <name>

Specifies the signal-space projection file.

11.12 Computing sensitivity maps

11.12.1 Purpose

mne_sensitivity_map computes the size of the columns of the forward operator and outputs the result in w files.

11.12.2 Command line options

mne_sensitivity_map accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--fwd <name>

Specifies a forward solution file to analyze. By default the MEG forward solution is considered.

--eeg

Use the EEG forward solution instead of the MEG one. It does not make sense to consider a combination because of the different units of measure. For the same reason, gradiometers and magnetometers have to be handled separately, see **--mag** option below. By default MEG gradiometers are included.

--mag

Include MEG magnetometers instead of gradiometers

--w <name>

Specifies the stem of the output w files. To obtain the final output file names, **-lh.w** and **-rh.w** is appended for the left and right hemisphere, respectively.

--smooth <number>

Specifies the number of smoothsteps to apply to the resulting w files. Default: no smoothing.

--map <number>

Select the type of a sensitivity map to compute. At present, valid numbers are 1, 2, and 3. For details, see Section 11.12.3, below.

11.12.3 Available sensitivity maps

In the following, let

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{g}_{xk} & \mathbf{g}_{yk} & \mathbf{g}_{zk} \end{bmatrix}$$

denote the three consecutive columns of the gain matrix \mathbf{G} corresponding to the fields of three orthogonal dipoles at source space location k . Further, let's assume that the source coordinate system has been selected so that the z -axis points to the cortical normal direction and the xy plane is thus the tangent plane of the cortex at the source space location k . Next, compute the SVD

$$\mathbf{G}_k = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{V}_k$$

and let $\mathbf{g}_{1k} = \mathbf{u}_{1k} \lambda_{1k}$, where λ_{1k} and \mathbf{u}_{1k} are the largest singular value and the corresponding left singular vector of \mathbf{G}_k , respectively. It is easy to see that \mathbf{g}_{1k} has the largest power among the signal distributions produced by unit dipoles at source space location k .

With these definitions the map selections defined with the **--map** option correspond to the following

--map 1

Compute $\sqrt{\mathbf{g}_{1k}^T \mathbf{g}_{1k}} = \lambda_{1k}$ at each source space point. Normalize the result so that the maximum values equals one.

--map 2

Compute $\sqrt{\mathbf{g}_z^T \mathbf{g}_z}$ at each source space point. Normalize the result so that the maximum values equals one. This is the amplitude of the signals produced by unit dipoles normal to the cortical surface.

--map 3

Compute $\sqrt{\mathbf{g}_z^T \mathbf{g}_z / \mathbf{g}_{1k}^T \mathbf{g}_{1k}}$ at each source space point.

--map 4

Compute $1 - \sqrt{\mathbf{g}_z^T \mathbf{g}_z / \mathbf{g}_{1k}^T \mathbf{g}_{1k}}$ at each source space point. This could be called the *radiality index*.

11.13 Transforming locations

11.13.1 Purpose

mne_transform_points applies the coordinate transformation relating the MEG head coordinates and the MRI coordinates to a set of locations listed in a text file.

11.13.2 Command line options

mne_transform_points accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--in <name>

Specifies the input file. The file must contain three numbers on each line which are the *x*, *y*, and *z* coordinates of point in space. By default, the input is in millimeters.

--iso <name>

Specifies a name of a fif file containing Isotrak data. If this option is present file will be used as the input instead of the text file specified with the **--in** option.

--trans <name>

Specifies the name of a fif file containing the coordinate transformation between the MEG head coordinates and MRI coordinates. If

this file is not present, the transformation will be replaced by a unit transform.

--out <name>

Specifies the output file. This file has the same format as the input file.

--hpts

Output the data in the *head points* (hpts) format accepted by *tkmedit*. In this format, the coordinates are preceded by a point category (hpi, cardinal, eeg, extra) and a sequence number.

--meters

The coordinates are listed in meters rather than millimeters.

--tomri

By default, the coordinates are transformed from MRI coordinates to MEG head coordinates. This option reverses the transformation to be from MEG head coordinates to MRI coordinates.

--tomrivol

This option makes the head points output compatible with the present version of *tkmedit*. The output will be in MRI voxel coordinates divided by 1000.

11.14 Inquiring and changing baselines

The utility *mne_change_baselines* computes baseline values and applies them to an evoked-response data file. The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--in <name>

Specifies the input data file.

--set <number>

The data set number to compute baselines from or to apply baselines to. If this option is omitted, all average data sets in the input file are processed.

--out <name>

The output file.

--baselines <name>

Specifies a text file which contains the baseline values to be applied. Each line should contain a channel name, colon, and the baseline value given in ‘native’ units (T/m, T, or V). If this option is encountered, the `--bmin` and `--bmax` options will be unset.

--list <name>

Specifies a text file to contain the baseline values. Listing is provided only if a specific data set is selected with the `--set` option.

--bmin <value/ms>

Lower limit of the baseline. Effective only if `--baselines` option is not present. Both `--bmin` and `--bmax` must be present to compute the baseline values. If either `--bmin` or `--bmax` is encountered, the `--baselines` option will be unset.

--bmax <value/ms>

Upper limit of the baseline.

11.15 Data simulator

11.15.1 Purpose

The utility *mne_simu* creates simulated evoked response data for investigation of the properties of the inverse solutions. It computes MEG signals generated by dipoles normal to the cortical mantle at one or several ROIs defined with label files. Colored noise can be added to the signals.

11.15.2 Command-line options

mne_simu has the the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--fwd <name>

Specify a forward solution file to employ in the simulation.

--label <name>

Specify a label

--meg

Provide MEG data in the output file.

--eeg

Provide EEG data in the output file.

--out <name>

Specify the output file. This will be an evoked data file in the fif format.

--mat

Produce Matlab output of the simulated fields instead of the fif evoked file.

--label <name>

Define an ROI. Several label files can be present. The sources in the labels will have non-overlapping timecourses, see below.

--do_all

Activate all sources on the cortical surface uniformly. This overrides the **--label** options.

11.15.3 Noise simulation

Noise is added to the signals if the **--senscov** and **--nave** options are present. If **--nave** is omitted the number of averages is set to $L = 100$. The noise is computed by first generating vectors of Gaussian random numbers $\mathbf{n}(t)$ with $n_j(t) \sim N(0, 1)$. Thereafter, the noise-covariance matrix \mathbf{C} is used to color the noise:

$$\mathbf{n}_c(t) = \frac{1}{\sqrt{L}} \mathbf{\Lambda} \mathbf{U}^T \mathbf{n}(t),$$

where we have used the eigenvalue decomposition positive-definite covariance matrix:

$$\mathbf{C} = \mathbf{U} \mathbf{\Lambda}^2 \mathbf{U}^T.$$

Note that it is assumed that the noise-covariance matrix is given for raw data, *i.e.*, for $L = 1$.

11.15.4 Simulated data

The source waveform q_k for the k^{th} label is nonzero at times $t_{kp} = (100(k-1) + p)/f_s$, $p = 0 \dots 100$ with:

$$q_k(t_{kp}) = Q_k \cos^2\left(\frac{\pi p}{100} - \frac{\pi}{2}\right),$$

i.e., the source waveforms are non-overlapping 100-samples wide \cos^2 pulses. The sampling frequency $f_s = 600\text{Hz}$. The source amplitude Q_k is determined so that the strength of each of the dipoles in a label will be $50 \text{ nAm}/N_k$.

The base fields \mathbf{x}_k are sums of the magnetic fields and electric potentials produced by the dipoles normal to the cortical mantle at each label. The simulated signals are then:

$$\mathbf{x}(t_j) = \sum_{k=1}^{N_s} q_k(t_j) \mathbf{x}_k + \mathbf{n}_c(t_j),$$

where N_s is the number of sources.

11.16 Converting parcellation data into labels

The utility *mne_annot2labels* converts cortical parcellation data into a set of labels. The parcellation data are read from the directory `$SUBJECTS_DIR/$SUBJECT/label` and the resulting labels are written to the current directory. *mne_annot2labels* requires that the environment variable `$SUBJECTS_DIR` is set. The command line options for *mne_annot2labels* are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--subject <name>

Specifies the name of the subject. If this option is not present the `$SUBJECT` environment variable is consulted. If the subject name cannot be determined, the program quits.

--parc <name>

Specifies the parcellation name to convert. The corresponding parcellation file names will be `$SUBJECTS_DIR/$SUBJECT/label/<hemi>h.<name>.annot` where `<hemi>` is `l` or `r` for the left and right hemisphere, respectively.

CHAPTER 12 The sample data set

12.1 Purpose

This Chapter gives a detailed description of the processing of a sample data set, which can be employed to familiarize with the workflow described in Chapter 3.

Important: Going through the analysis exercise in this chapter is not a substitute for reading other chapters of this manual and understanding the concepts underlying MNE software.

12.2 Overview

The MNE software is accompanied by a sample data set which includes the MRI reconstructions created with FreeSurfer and the an MEG/EEG data set. These data were acquired with the Neuromag Vectorview system at MGH/HMS/MIT Athinoula A. Martinos Center Biomedical Imaging. EEG data from a 60-channel electrode cap was acquired simultaneously with the MEG. The original MRI data set was acquired with a Siemens 1.5 T Sonata scanner using an MPRAGE sequence.

Important: These data are provided solely for the purpose of getting familiar with the MNE software. They should not be redistributed to third parties. The data should not be used to evaluate the performance of the MEG or MRI system employed.

In the MEG/EEG experiment, checkerboard patterns were presented into the left and right visual field, interspersed by tones to the left or right ear. The interval between the stimuli was 750 ms. Occasionally an smiley face was presented at the center of the visual field. The subject was asked to press a key with the right index finger as soon as possible after the

appearance of the face. A listing of the corresponding trigger codes is provided in Table 12.1

Name	#	Contents
LA	1	Response to left-ear auditory stimulus
RA	2	Response to right-ear auditory stimulus
LV	3	Response to left visual field stimulus
RV	4	Response to right visual field stimulus
smiley	5	Response to the smiley face
button	32	Response triggered by the button press

Table 12.1 Trigger codes for the sample data set.

12.3 Setting up

The sample data set is distributed with the MNE software as a compressed tar archive located at `$MNE_ROOT/sample-data/MNE-sample-data.tar.gz`. To make a personal copy of the sample data set, follow these steps:

1. Set up for using the MNE software as instructed in Section 2.5 of this manual.
2. Create a directory for your personal copy: `mkdir <yourdir>`, where `<yourdir>` is the location where you want your personal copy to reside. To store the sample data set and to finish the tutorials in this Chapter, you need approximately 600 MBytes of space on the disk where `<yourdir>` is located.
3. Go to your newly created sample data directory: `cd <yourdir>`.
4. Extract the sample data: `tar zxvf <dir>/MNE-sample-data.tar.gz`, where `<dir>` is the location of the tar archive, provided by your system administrator.

To start the tutorials you need to:

1. Set up MNE software user environment, see Section 2.5.
2. Set the `SUBJECTS_DIR` environment variable:
`setenv SUBJECTS_DIR <yourdir>/subjects` (csh and tcsh)
or `export SUBJECTS_DIR=<yourdir>/subjects` (POSIX-compatible shell). Most users at the Martinos Center have tcsh as their login shell.
3. Assign the `SUBJECT` environment variable the value `sample`.

4. For convenience, you can also set the environment variable `SAMPLE` to `<yourdir>`. The following tutorial examples assume you have done this.
5. Set up the *FreeSurfer* environment using the commands specific to your site. The *FreeSurfer* license is needed for the source space creation covered in Section 12.5.2.

Note: From this point on, directories and files under your personal copy of the sample data set under `<yourdir>` will be referred to by relative pathnames. For example, the file `<yourdir>/MEG/sample/audvis.ave` will be simply called `MEG/sample/audvis.ave`.

Note: You can also proceed without *FreeSurfer* installed if you choose to use source space creation using the recursively subdivided octahedron or icosahedron method. For more information, see the Note in Section 12.5.2.

12.4 Contents of the data set

The sample data set contains two main directories: `MEG/sample` (the MEG/EEG data) and `subjects/sample` (the MRI reconstructions). An overview of the data provided is given in Tables 12.2 and 12.3. In addition to subject *sample*, the MRI surface reconstructions from another subject, *morph*, are provided to demonstrate the morphing capabilities of the MNE software.

File	Contents
<code>sample/audvis_raw.fif</code>	The raw MEG/EEG data
<code>audvis.ave</code>	A template script for off-line averaging
<code>auvis.cov</code>	A template script for the computation of a noise-covariance matrix

Table 12.2 Contents of the MEG/sample directory.

File / directory	Contents
<code>bem</code>	Directory for the forward modelling data
<code>bem/watershed</code>	BEM surface segmentation data computed with the watershed algorithm
<code>bem/inner_skull.surf</code>	Inner skull surface for BEM
<code>bem/outer_skull.surf</code>	Outer skull surface for BEM

Table 12.3 Overview of the contents of the subjects/sample directory.

File / directory	Contents
bem/outer_skin.surf	Skin surface for BEM
sample-head.fif	Skin surface in fif format for mne_analyze visualizations
surf	Surface reconstructions
mri/T1	The T1-weighted MRI data employed in visualizations

Table 12.3 Overview of the contents of the subjects/sample directory.

The following preprocessing steps have been already accomplished in the sample data set:

1. The MRI surface reconstructions have been computed using the FreeSurfer software.
2. The BEM surfaces have been created with the watershed algorithm, see Section A.1.
3. The MEG/EEG raw data file has been checked with the utilities described in Sections 3.9.1 and 3.9.2.
4. Template scripts for averaging and computation of the noise-covariance matrices have been written.

12.5 Setting up subject-specific data

12.5.1 Structural MRIs

To set up the structural MRIs for processing with the Neuromag MRI viewer, MRILab, say

```
mne_setup_mri
```

This command sets up the directories `subjects/sample/mri/T1-neuromag` and `subjects/sample/mri/brain-neuromag`. For more information, see Section 3.4.

12.5.2 Source space

The source space with a 5-mm grid spacing is set up by saying:

```
mne_setup_source_space --ico -6
```

This command sets up the source-space related files in directory `subjects/sample/bem` as described in Section 3.5.

12.5.3 Boundary-element models

The geometry calculations for the single-layer boundary-element model are accomplished with the command:

```
mne_setup_forward_model --homog --surf --ico 4
```

This command sets up the homogeneous BEM-model related files in directory `subjects/sample/bem` as described in section Section 3.7. In addition to the homogeneous BEM, you also need the three-layer BEM model, which can be used for both EEG and MEG:

```
mne_setup_forward_model --surf --ico 4
```

The above commands employ the `inner_skull.surf`, `outer_skull.surf`, and `outer_skin.surf` triangulation files located in `subjects/sample/bem`. The option `--ico 4` will create a model with 5120 triangles on each surface. Depending on the speed of your computer, the three-layer model may take quite a while to set up.

12.6 Setting up a custom EEG layout

A data specific EEG layout will facilitate viewing of the EEG data in *mne_browse_raw*, *mne_analyze*, and the Neuromag plotting program *xplotter*. The MNE programs *mne_browse_raw* and *mne_analyze* look for user-specific layouts in `$HOME/.mne/lout`. Thus, you can create an EEG layout for the sample data set with the following commands:

```
mkdir -p $HOME/.mne/lout
cd $SAMPLE/MEG/sample
mne_make_eeg_layout --fif sample_audvis_raw.fif \
--lout $HOME/.mne/lout/sample-EEG.lout
```

Please refer to Section 11.6 for more information on *mne_make_eeg_layout*.

Note: It is usually sufficient to create one EEG layout for each electrode cap you are using in your experiment rather than using a different layout file for each data file generated using the same cap.

12.7 Previewing the data

Previewing your raw data before proceeding to averaging and computation of the current estimates is most important to avoid unintentional errors caused by noisy or dysfunctional channels, frequent eye blinks, inappropriate bandpass filtering etc.

One possible routemap for the preview session is outlined below:

1. Go to the MEG/EEG data directory: `cd $SAMPLE/MEG/sample.`
2. Launch *mne_browse_raw*.
3. Open the raw data file `sample_audvis_raw.fif` from *File/Open....*
4. Switch all SSP vectors off from *Adjust/Projection....*
5. Set the lowpass filter corner to a high value, e.g., 150 Hz from *Adjust/Filter...*
6. Browse through all channels using the selections available under *Adjust/Selection...* and look for very noisy or flat channels. You should be able to readily identify two such channels among all MEG and EEG channels. You may need to click *Remove DC* to reliably associate the noisy or flat waveform with the channel name on the left. Also, experiment with switching the EEG average reference projection on and off and you will notice that the EEG bad channel cannot be seen after the projection.
7. Mark the the channels you identified in step 6. bad for this viewing session by clicking on their channel names on the left. You can save the bad channel selection to the file from *File/Apply bad channels*. Bad channel marking can be removed by clicking on their channel names again and selecting *File/Apply bad channels*. Alternatively, you can use the utility *mne_mark_bad_channels* to set a bad channel selection, see Section 11.4.1.
8. Switch the projections back on and change filter to a 40-Hz lowpass.
9. Compute a preliminary average for the left-ear auditory stimulus:
 - a. Open the averaging preferences dialog (*Adjust/Averaging preferences....*)
 - b. Set the time scale to -100...300 ms.
 - c. Click on the text next to *Average:* in the main window and press return. After a while, a topographical display appears with the averaged responses. Notice that the bad channels are not displayed.
 - d. Change to different layouts from *Adjust/Full view layout....* Inspect both the MEG and EEG waveforms.
5. Compute a preliminary average for the right-ear auditory stimulus:
 - a. Open the averaging preferences.
 - b. Click on the *Trace color...* button and change the trace color to something different from the default yellow.
 - c. Change the text next to *Average:* to 2 and press return. Average to the right-ear tones will be computed. Compare the to sets of averages and verify that all channels show reasonable data.
4. Goto *Adjust/Manage averages...* and delete the preliminary averages just computed.

After these steps, you are ready to proceed to the actual analysis.

12.8 Off-line averaging

Go to directory `$SAMPLE/MEG/sample` . With help of Section 4.8, familiarize yourself with the averaging script `audvis.ave`.

12.8.1 Using the averaging script interactively

You can invoke an averaging script in *mne_browse_raw* from *Process/Average....* Select the `audvis.ave` script from the file selection box that appears. Once averaging is complete, you can inspect the details of the averaged responses in the *Averages* window, which appears automatically. You can redisplay it from *Windows/Show averages....* The window, which appears when you select *Adjust/Manage averages..* allows you to:

1. Select which conditions (categories) are displayed.
2. Change the trace colors.
3. Inspect the averaging log.
4. Save the averaged data.
5. Delete this set of averages.

Note: If you decide to save the averages in the interactive mode, use the name `sample_audvis-ave.fif` for the result.

12.8.2 Using the averaging script in batch mode

The batch-mode version of *mne_browse_raw*, *mne_process_raw* can be used for averaging as well. Batch mode averaging can be done with the command:

```
mne_process_raw --raw sample_audvis_raw.fif \
--lowpass 40 --projoff \
--saveavetag -ave --ave audvis.ave
```

The functions of the options are:

--raw

Specifies the raw file.

--lowpass

Specifies the lowpass filter corner frequency.

--projoff

Do not apply signal-space projection and average electrode reference to the data. Regardless, the projection information is included with the data file so that it can be applied later. It is also possible to specify the `--projon` option but then there is no possibility to view unprojected data in subsequent phases of the analysis.

--saveavetag

Specifies how the averages are named. With this option, the `_raw.fif` ending is stripped of the original raw data file and the tag specified with this option (`-ave`) is added. The average file and the corresponding log file will have the extensions `.fif` and `.log`, respectively.

--ave

Specifies the averaging script.

As a result of running the averaging script a file called `sample_audvis-ave.fif` is created. It contains averages to the left and right ear auditory as well as to the left and right visual field stimuli.

12.9 Viewing the off-line average

The average file computed in the previous section can be viewed in `mne_browse_raw`.

To view the averaged signals, invoke `mne_browse_raw`:

```
cd $SAMPLE/MEG/sample
mne_browse_raw &
```

This Section gives only very basic information about the use of `mne_browse_raw` for viewing evoked-response data. Please consult Chapter 4 for more comprehensive information.

12.9.1 Loading the averages

`mne_browse_raw` loads all the available data from an average file at once:

1. Select *Open evoked...* from the *File* menu.
2. Select the average file `sample_audvis-ave.fif` file from the list and click *OK*.
3. A topographical display of the waveforms with gradiometer channels included appears.

12.9.2 Inspecting the auditory data

Select the left and right ear auditory stimulus responses for display:

1. Select *Manage averages...* from the *Adjust* menu.
2. Click off all other conditions except the auditory ones.

Set the time scale and baseline:

1. Select *Scales...* from the *Adjust* menu.
2. Switch off *Autoscale time range* and set the *Average time range* from -200 to 500 ms.
3. Switch on *Use average display baseline* and set *Average display baseline* from -200 to 0 ms.
4. Click *OK*.

You can display a subset of responses from the topodisplay by holding the shift key down and dragging with the mouse, left button down. When you drag on the response with just the left button down, the signal timing, and channel name are displayed at the bottom. If the left mouse button is down and you press shift down the time is give both in absolute units and relative to the point where shift was pressed down.

Observe the following:

1. The main deflection occurs around 100 ms over the left and right temporal areas.
2. The left-ear response (shown in yellow) is stronger on the right than on the left. The opposite is true for the the right-ear response, shown in red.

12.9.3 Inspecting the visual data

Go back to the *Manage averages...* dialog and switch all other conditions except the visual ones.

Observe the following:

1. The left and right visual field responses are quite different in spatial distribution in the occipital area.
2. There is a later response in the right parietal area, almost identical to both visual stimuli.

Tip: If you have the Neuromag software available, the averaged data can be also viewed in the Neuromag data plotter (*xplotter*). See Section B.2 for instructions on how to use the Neuromag software at the MGH Martinos Center.

12.10 Computing the noise-covariance matrix

Another piece of information derived from the raw data file is the estimate for the noise-covariance matrix, which can be computed with the command:

```
mne_process_raw --raw sample_audvis_raw.fif \
--lowpass 40 --projon \
--savecovtag -cov --cov audvis.cov
```

Using the definitions in `audvis.cov`, this command will create the noise-covariance matrix file `sample_audvis-cov.fif`. In this case the projections are set on. The projection information is then attached to the noise-covariance matrix and will be automatically loaded when the inverse-operator decomposition is computed.

Tip: You can study the contents of the covariance matrix computation description file `audvis.cov` with help of Section 4.9

12.11 MEG-MRI coordinate system alignment

12.11.1 Using `mne_analyze`

The `mne_analyze` module of the MNE is one option for the coordinate alignment. It uses a triangulated scalp surface to facilitate the alignment.

12.11.1.1 Initial alignment

Follow these steps to make an initial approximation for the coordinate alignment.

1. Go to directory `MEG/sample`.
2. Launch `mne_analyze`
3. Select *File/Load digitizer data...* and load the digitizer data from `sample_audvis_raw.fif`.
4. Load an inflated surface for subject *sample* from *File/Load surface...*
5. Bring up the viewer window from *View/Show viewer...*
6. Click *Options...* in the viewer window. Make the following selections:
 - a. Switch left and right cortical surface display off.
 - b. Make the scalp transparent.
 - c. Switch *Digitizer data* on.
4. After a while, the digitizer points will be shown. The color of the circles indicates whether the point is inside (blue) or outside (red) of the scalp. The HPI coils are shown in green and the landmark locations in light blue or light red color. The initial alignment is way off!
5. Switch the *Digitizer data* off to get the big circles out of the way.
6. Bring up the coordinate alignment window from *Adjust/Coordinate alignment...*
7. Click on the *RAP* (Right Auricular Point) button. It turns red, indicating that you should select the point from the viewer window. Click at the approximate location of this point in the viewer. The button jumps up, turns to normal color, and the MRI coordinates of the point appear in the text fields next to the button.

8. Proceed similarly for the other two landmark points: Nasion and LAP (Left Auricular Point).
9. Press *Align using fiducials*. Notice that the coordinate transformation changes from a unit transformation (no rotation, no origin translation) to a one determined by the identified landmark locations. The rotation matrix (upper left 3 x 3 part of the transformation) should have positive values close to one on the diagonal. There is a significant rotation around the x axis as indicated by elements (3,2) and (2,3) of the rotation matrix. The x and y values of the translation should be small and the z value should be negative, around -50 mm. An example of an initial coordinate transformation is shown in Figure 12.1.
10. Make the *Digitizer data* again visible from the options of the viewer window. Note that the points are now much closer to the scalp surface.

Current coordinate transformation: head -> MRI			
0.997093	0.057963	0.049446	-2.76 mm
-0.037658	0.939130	-0.341491	-7.23 mm
-0.066230	0.338636	0.938583	-49.54 mm
0.000000	0.000000	0.000000	1.00
Fiducial	xMRI / mm	yMRI / mm	zMRI / mm
LAP	-79.0	-4.3	-44.5
Nasion	3.5	94.7	12.8
RAP	72.8	10.1	54.6

Figure 12.1 Example of an initial coordinate alignment.

12.11.1.2 Refining the coordinate transformation

Before proceeding to the refinement procedure, it is useful to remove outlier digitizer points. When you rotate the image in the viewer window, you will notice that there is at least one such point over the right cheek. To discard this point:

1. Click on *Discard* in the *Adjust coordinate alignment* window.
2. Enter 10 for the distance of the points to be discarded.
3. Click done. The outlier point disappears.

The coordinate transformation can be adjusted manually with the arrow buttons in the middle part of the *Adjust coordinate alignment* dialog. These buttons move the digitizer points in the directions indicated by the amount listed next to each of the buttons.

An automatic iterative procedure, Iterative Closest Point (ICP) matching is also provided. At each iteration step

1. For each digitizer point, transformed from MEG to the MRI coordinate frame, the closest point on the triangulated surface is determined.
2. The best coordinate transformation aligning the digitizer points with the closest points on the head surface is computed.

In step 2 of the iteration, the nasion is assigned five times the weight of the other points since it can be assumed that the nasion is the easiest point to identify reliably from the surface image.

The ICP alignment can be invoked by entering the desired number of iterations next to the *ICP align* button followed by return or simply pressing the *ICP align* button. The iteration will converge in 10 to 20 steps.

Warning: The ICP alignment is a recent addition to *mne_analyze*. Use it with caution. The iteration will not converge to a reasonable solution unless an initial alignment is performed first according to Section 12.11.1.1. Outlier points should be excluded as described above. No attempt is made to compensate for the possible distance of the digitized EEG electrode locations from the scalp.

12.11.1.3 Saving the transformation

To create a MRI fif description file which incorporates the coordinate transformation click *Save MRI set* in the *Adjust coordinate alignment* dialog. This will create the MRI set file in the `$SUBJECTS_DIR/sample/mri/T1-neuromag/sets` directory, which was created by *mne_setup_mri_data*, see Section 12.5.1. The file will be called

`COR-<username>-<date>-<time>.fif`

where `<username>` is your login name.

You can also save transformation to a fif file through the *Save...* button. If the file does not exist, it will only contain the coordinate transformation. If the file exists it will be inserted to the appropriate context. An existing transformation will not be replaced unless *Overwrite existing transform* is checked in the save dialog.

Once you have saved the coordinate transformation, press *Done* and quit *mne_analyze* (*File/Quit*).

Important: If you dismiss the alignment dialog before saving the transformation, it will be lost.

12.11.2 Using MRIlab

The MEG-MRI coordinate system alignment can be also accomplished with the Neuromag tool MRIlab. This tool is only available on the HP-UX workstations at the moment. To start it, log on go to an HP-UX workstation, go to your sample data directory, and say:

```
/neuro/bin/vue/mrilab &
```

The following steps are necessary for the coordinate system alignment:

1. Load the MRI description file `COR.fif` from `subjects/sample/mri/T1-neuromag/sets` through *File/Open*.
2. Open the landmark setting dialog from *Windows/Landmarks*.
3. Click on one of the coordinate setting fields on the *Nasion* line. Click *Goto* . Select the crosshair tool and move the crosshair to the nasion. Click *Get*.
4. Proceed similarly for the left and right auricular points. Your instructor will help you with the selection of the correct points.
5. Click *OK* to set the alignment
6. Load the digitization data from the file `sample_audvis_raw.fif` or `sample_audvis-ave.fif` (the on-line evoked-response average file) in `MEG/sample` through *File/Import/Isotrak data*. Click *Make points* to show all the digitization data on the MRI slices.
7. Check that the alignment is correct by looking at the locations of the digitized points are reasonable. Adjust the landmark locations using the *Landmarks* dialog, if necessary.
8. Save the aligned file to the file suggested in the dialog coming up from *File/Save*.

12.12 The forward solution

To compute the forward solution, say:

```
cd $SAMPLE/MEG/sample
mne_do_forward_solution --mindist 5 \
--spacing oct-6 --meas sample_audvis-ave.fif
```

This produces an EEG and MEG forward solution with source space points closer than 5 mm to the inner skull surface omitted. The source space created in Section 12.5.2 will be employed. As the output from this command will indicate The forward solution will be stored in file `sample_audvis-ave-oct-6-fwd.fif`.

This command uses the three-layer BEM model `sample-5120-5120-5120-bem-sol.fif` created in Section 12.5.3. If you want to use the single-compartment BEM `sample-5120-bem-sol.fif` usable for MEG data only say:

```
cd $SAMPLE/MEG/sample
mne_do_forward_solution --mindist 5 \
--spacing oct-6 --meas sample_audvis-ave.fif \
--bem sample-5120 --megonly
```

12.13 The inverse operator decomposition

The inverse operator information, necessary for the computation of the MNEs and dSPMs is accomplished by the command:

```
mne_do_inverse_operator \
--fwd sample_audvis-ave-oct-6-fwd.fif \
--depth --loose 0.2 --meg --eeg
```

This produces a depth-weighted inverse operator decomposition with ‘loose’ orientation constraint applied. More details on the convenience script *mne_do_inverse_operator* are provided in Section 3.13.

The above command employs both EEG and MEG data. To create separate solution for EEG and MEG, run the commands:

```
mne_do_inverse_operator \
--fwd sample_audvis-ave-oct-6-fwd.fif \
--depth --loose 0.2 --meg
```

and

```
mne_do_inverse_operator \
--fwd sample_audvis-ave-oct-6-fwd.fif \
--depth --loose 0.2 --eeg
```

Note: If you were using a single-compartment BEM to compute the forward solution, you can only compute the MEG inverse operator.

12.14 Interactive analysis

The most exciting part of this exercise is to explore the data and the current estimates in *mne_analyze*. This section contains some useful steps to get you started. A lot of information about the capabilities of *mne_analyze* is given in Chapter 7. Batch-mode processing with *mne_make_movie* is discussed in Section 6.5. Cross-subject averaging is covered in Chapter 8.

Before launching *mne_analyze* it is advisable to go to the directory *MEG/sample*. The current working directory can be also changed from *mne_analyze*.

12.14.1 Getting started

Launch *mne_analyze*. Select *Help/On GLX...*, which brings up a window containing Open GL rendering context information. If first line in the information dialog that pops up says *Nondirect rendering context* instead of *Direct rendering context* you will experience slow graphics performance. To fix this, your system software, graphics adapter or both need to be updated. Consult a computer support person for further information.

12.14.2 Load surfaces

It is reasonable to start the analysis by loading the display surfaces: choose the inflated surface for subject *sample* from the dialog that appears when you select *File/Load surface...*

12.14.3 Load the data

Select *File/Open...* Select *sample_audvis-ave.fif* as your data file and select the *Left auditory* data set. Select the inverse operator *sample_audvis-ave-oct-6-meg-eeg-inv.fif* and press *OK*. After a while the signals appear in the sample waveform and topographical displays. Click on the N100m peak in the auditory response. A dSPM map appears in the main surface display.

12.14.4 Show field and potential maps

Select *Windows/Show viewer...* After a while the viewer window appears. Click on the N100m peak again. Once the field map preparation computations are complete, the magnetic field and potential maps appear. Investigate the viewer window options with help of Section 7.10.

12.14.5 Show current estimates

The options affecting the current estimates are accessible from *Adjust/Estimate parameters...* With help of Section 7.12, investigate the effects of the parameter settings.

12.14.6 Labels and timecourses

While in directory *MEG/sample*, create a directory called *label*:

```
mkdir label
```

Using the information in Section 7.13.4 create two labels: `A-lh.label` and `A-rh.label` in the approximate location of the left and right auditory cortices. Save these labels in the newly created `label` directory.

Load all labels from the `label` directory and investigate the timecourses in these two labels as well as at individual vertices. Information on label processing can be found from Section 7.13.

12.14.7 Morphing

Goto to `$SUBJECTS_DIR` and create the directory `morph-maps`. Load the inflated surface for subject *morph* as the morphing surfaces. Try switching between the original and morphing surfaces. More information about morphing is available in Section 7.9 and in Chapter 8.

There is also a left-hemisphere occipital patch file available for subject *morph*. Load a right-hemifield visual response instead of the auditory one and investigate mapping of the current estimates on the patch.

CHAPTER 13 **Useful reading**

13.1 General MEG reviews

M. Hämäläinen, R. Hari, R. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa, "Magnetoencephalography – theory, instrumentation, and applications to noninvasive studies of the working human brain," *Reviews of Modern Physics*, vol. 65, pp. 413-497, 1993.

S. Baillet, J. C. Mosher, and R. M. Leahy, "Electromagnetic Brain Mapping," *IEEE Signal Processing Magazine*, vol. 18, pp. 14 - 30, 2001.

M. Hämäläinen and R. Hari, "Magnetoencephalographic Characterization of Dynamic Brain Activation: Basic Principles and Methods of Data Collection and Source Analysis," in *Brain mapping : the methods*, A. W. Toga and J. C. Mazziotta, Eds. Amsterdam ; Boston: Academic Press, 2002.

13.2 Cortical surface reconstruction and morphing

A. M. Dale, B. Fischl, and M. I. Sereno, "Cortical surface-based analysis. I. Segmentation and surface reconstruction," *Neuroimage*, vol. 9, pp. 179-94., 1999.

B. Fischl, M. I. Sereno, and A. M. Dale, "Cortical surface-based analysis. II: Inflation, flattening, and a surface-based coordinate system," *Neuroimage*, vol. 9, pp. 195-207., 1999.

B. Fischl, M. I. Sereno, R. B. Tootell, and A. M. Dale, "High-resolution intersubject averaging and a coordinate system for the cortical surface," *Hum Brain Mapp*, vol. 8, pp. 272-84, 1999.

13.3 Forward modelling

M. S. Hämäläinen and J. Sarvas, "Realistic conductivity geometry model of the human head for interpretation of neuromagnetic data," *IEEE Trans. Biomed. Eng.*, vol. BME-36, pp. 165-171, 1989.

B. Fischl, D. H. Salat, A. J. van der Kouwe, N. Makris, F. Segonne, B. T. Quinn, and A. M. Dale, "Sequence-independent segmentation of magnetic resonance images," *Neuroimage*, vol. 23 Suppl 1, pp. S69-84, 2004.

J. Jovicich, S. Czanner, D. Greve, E. Haley, A. van der Kouwe, R. Gollub, D. Kennedy, F. Schmitt, G. Brown, J. Macfall, B. Fischl, and A. Dale, "Reliability in multi-site structural MRI studies: effects of gradient non-linearity correction on phantom and human data," *Neuroimage*, vol. 30, pp. 436-43, 2006.

J. C. Mosher, R. M. Leahy, and P. S. Lewis, "EEG and MEG: forward solutions for inverse methods," *IEEE Trans Biomed Eng*, vol. 46, pp. 245-59, 1999.

13.4 Signal-space projections

C. D. Tesche, M. A. Uusitalo, R. J. Ilmoniemi, M. Huotilainen, M. Kajola, and O. Salonen, "Signal-space projections of MEG data characterize both distributed and well-localized neuronal sources," *Electroencephalogr Clin Neurophysiol*, vol. 95, pp. 189-200, 1995.

M. A. Uusitalo and R. J. Ilmoniemi, "Signal-space projection method for separating MEG or EEG into components," *Med Biol Eng Comput*, vol. 35, pp. 135-40, 1997.

13.5 Minimum-norm estimates

M. Hämäläinen and R. Ilmoniemi, "Interpreting magnetic fields of the brain: minimum norm estimates," Helsinki University of Technology, Espoo TKK-F-A559, 1984.

A. Dale and M. Sereno, "Improved localization of cortical activity by combining EEG and MEG with MRI cortical surface reconstruction: A linear approach," *J. Cog. Neurosci*, vol. 5, pp. 162-176, 1993.

M. S. Hämäläinen and R. J. Ilmoniemi, "Interpreting magnetic fields of the brain: minimum norm estimates," *Med Biol Eng Comput*, vol. 32, pp. 35-42., 1994.

A. M. Dale, A. K. Liu, B. R. Fischl, R. L. Buckner, J. W. Belliveau, J. D. Lewine, and E. Halgren, "Dynamic statistical parametric mapping: combining fMRI and MEG for high-resolution imaging of cortical activity," *Neuron*, vol. 26, pp. 55-67, 2000.

A. K. Liu, A. M. Dale, and J. W. Belliveau, "Monte Carlo simulation studies of EEG and MEG localization accuracy," *Hum Brain Mapp*, vol. 16, pp. 47-62, 2002.

F. H. Lin, J. W. Belliveau, A. M. Dale, and M. S. Hamalainen, "Distributed current estimates using cortical orientation constraints," *Hum Brain Mapp*, 2005.

13.6 fMRI-weighted estimates

A. M. Dale, A. K. Liu, B. R. Fischl, R. L. Buckner, J. W. Belliveau, J. D. Lewine, and E. Halgren, "Dynamic statistical parametric mapping: combining fMRI and MEG for high-resolution imaging of cortical activity," *Neuron*, vol. 26, pp. 55-67, 2000.

A. K. Liu, J. W. Belliveau, and A. M. Dale, "Spatiotemporal imaging of human brain activity using functional MRI constrained magnetoencephalography data: Monte Carlo simulations," *Proc Natl Acad Sci U S A*, vol. 95, pp. 8945-50., 1998.

F. H. Lin, T. Witzel, M. S. Hamalainen, A. M. Dale, J. W. Belliveau, and S. M. Stufflebeam, "Spectral spatiotemporal imaging of cortical oscillations and interactions in the human brain," *Neuroimage*, vol. 23, pp. 582-95, 2004.

APPENDIX A Creating the BEM meshes

A.1 Using the watershed algorithm

The watershed algorithm is part of the *FreeSurfer* software. The name of the program is *mri_watershed*. Its use in the MNE environment is facilitated by the script *mne_watershed_bem*, which assumes the following options:

--subject <subject>

Defines the name of the subject. This can be also accomplished by setting the SUBJECT environment variable.

--overwrite

Overwrite the results of previous run of *mne_watershed_bem*.

--atlas

Makes *mri_watershed* to employ atlas information to correct the segmentation.

After *mne_watershed_bem* has completed, the following files appear in the subject's *bem/watershed* directory:

<subject>_brain_surface

Contains the brain surface triangulation.

<subject>_inner_skull_surface

Contains the inner skull triangulation.

<subject>_outer_skull_surface

Contains the outer skull triangulation.

<subject>_outer_skin_surface

Contains the scalp triangulation.

All of these surfaces are in the FreeSurfer format. In addition, there will be a directory called *bem/watershed/ws* which contains the brain MRI volume. Furthermore, *mne_watershed_bem* script converts the scalp surface to *fif* format and saves the result to *bem/<subject>-head.fif*. The *mne_analyze* tool described Chapter 7 looks for this file the visualizations involving the scalp surface.

A.2 Using FLASH images

This method depends on the availability of MRI data acquired with a multi-echo FLASH sequence at two flip angles (5 and 30 degrees). These data can be acquired separately from the MPRAGE data employed in FreeSurfer cortical reconstructions but it is strongly recommended that they are collected at the same time with the MPRAGEs or at least with the same scanner. For easy co-registration, the images should have FOV, matrix, slice thickness, gap, and slice orientation as the MPRAGE data. For information on suitable pulse sequences, see reference [B. Fischl *et al.* and J. Jovicich *et al.*, 2006] in Section 13.3. At the Martinos Center, use of the 1.5-T Avanto scanner (Bay 2) is recommended for best results.

Creation of the BEM meshes using this method involves the following steps:

1. Organizing the MRI data. This is facilitated by the script *mne_organize_dicom*, see Section A.2.1.
2. Creating a synthetic 5-degree flip angle FLASH volume, register it with the MPRAGE data, and run the segmentation and meshing program. This step is accomplished by running the script *mne_flash_bem*, see Section A.2.2.
3. Inspecting the meshes with tkmedit, see Section A.2.3.

Note: The following sections assume that you have run the appropriate setup scripts to make both MNE and FreeSurfer software available.

A.2.1 Organizing MRI data into directories

Since all images comprising the multi-echo FLASH data are contained in a single series, it is necessary to organize the images according to the echoes before proceeding to the BEM surface reconstruction. This is accomplished by the *mne_organize_dicom* script, which creates a directory tree with symbolic links to the original DICOM image files. To run *mne_organize_dicom*, proceed as follows:

1. Copy all of your images or create symbolic links to them in a single directory. The images must be in DICOM format. We will refer to this directory as *<source>*.
2. Create another directory to hold the output of *mne_organize_dicom*. We will refer to this directory as *<dest>*.
3. Change the working directory to *<dest>*.
4. Say *mne_organize_dicom <source>*. Depending on the total number of images in *<source>* this script may take quite a while to run. Progress is indicated by listing the number of images processed at 50-image intervals.

As a result, *<dest>* will contain several directories named *<three-digit number>_<protocol_name>* corresponding to the different series of images acquired. Spaces and parenthesis in protocol names will be replaced by underscores. Under each of these directories there are one or more directories named *<three-digit>* number corresponding to one or more subsets of images in this series (protocol). The only subset division scheme implemented in *mne_organize_dicom* is that according to different echoes, typically found in multi-echo FLASH data. These second level directories will contain symbolic links pointing to the original image data.

Note: *mne_organize_dicom* was developed specifically for Siemens DICOM data. Its correct behavior with DICOM files originating from other MRI scanners has not been verified at this time.

Tip: Since *mne_organize_dicom* processes all images, not only the FLASH data, it may be a useful preprocessing step before FreeSurfer reconstruction process as well.

A.2.2 Creating the surface tessellations

The BEM surface segmentation and tessellation is automated with the script *mne_flash_bem*. It assumes that a FreeSurfer reconstruction for this subject is already in place. To run this utility:

1. Run *mne_organize_dicom* as described above.
2. Change to the *<dest>* directory where *mne_organize_dicom* created the image directory structure.
3. Create symbolic links from the directories containing the 5-degree and 30-degree FLASH series to *flash05* and *flash30*, respectively:
 - `ln -s <FLASH 5 series dir> flash05`
 - `ln -s <FLASH 30 series dir> flash30`
4. Set the *SUBJECTS_DIR* and *SUBJECT* environment variables
5. Run *mne_flash_bem*.

It may take a while for *mne_flash_bem* to complete. It uses the FreeSurfer directory structure under *\$SUBJECTS_DIR/\$SUBJECT*. The script encapsulates the following processing steps:

1. It create an mgz file corresponding to each of the eight echoes in each of the FLASH sequences in *mri/flash*. The files will be called *mef<flip-angle>_<echo-number>.mgz*.
2. It creates parameter maps in *mri/flash/parameter_maps* using *mri_ms_fitparms*.
3. It creates a synthetic 5-degree flip angle volume in *mri/flash/parameter_maps/flash5.mgz* using *mri_synthesize*.
4. Using *fsl_rigid_register*, it creates a registered 5-degree flip angle volume *mri/flash/parameter_maps/flash5_reg.mgz* by registering *mri/flash/parameter_maps/flash5.mgz* to the *T1* volume under *mri*.

5. Using *mri_convert*, it converts the *flash5_reg* volume to COR format under *mri/flash5*. If necessary, the *T1* and *brain* volumes are also converted into the COR format.
6. It runs *mri_make_bem_surfaces* to create the BEM surface tessellations.
7. It creates the directory *bem/flash*, moves the tri-format triangulations there and creates the corresponding FreeSurfer surface files in the same directory.
8. The COR format volumes created by *mne_flash_bem* are removed.

A.2.3 Inspecting the meshes

It is advisable to check the validity of the BEM meshes before using them. This can be done with help of *tkmedit* either before or after executing *mne_setup_forward_model*, see Section 3.7.

A.3 Using seglab

The brain segmentation provided by FreeSurfer in the directory *mri/brain* can be employed to create the inner skull surface triangulation with help of *seglab*, the Neuromag MRI segmentation tool. The description below assumes that the user is familiar with the *seglab* tool. If necessary, consult the *seglab* manual, Neuromag P/N NM20420A-A.

The data set *mri/brain* typically contains tissues within or outside the skull, in particular around the eyes. These must be removed manually before the inner skull triangulation is created. The editing and triangulation can be accomplished as outlined below

1. Set up the MRIs for Neuromag software access

Run the *mne_setup_mri* too as described in Section 3.4. As a result, the directories *mri/T1-neuromag* and *mri/brain-neuromag* are set up.

2. Load the MRI data

Open the file *mri/brain-neuromag/sets/COR.fif* and adjust the scaling of the data.

3. Preparatory steps

Set the minimum data value to 1 using the *min3D* operator. Make a backup of the data with the *backup3D* operator.

4. Manual editing

The *maskDraw3D* operation is recommended for manual editing. To use it, first employ the *grow3D* operator with threshold interval 2...255 and the seed point inside the brain. Then do the editing in the slicer window as described in Section 5.4.2 of the *seglab* man-

ual. Note that it is enough to remove the connectivity to the extracerebral tissues rather than erasing them completely.

5. Grow again and mask

Once manual editing is complete, employ the *grow3D* operator again and do *mask3D* with the backup data to see whether the result is satisfactory. If not, undo *mask3D* and continue manual editing. Otherwise, undo *mask3D* and proceed to the next step.

6. Dilation

It is advisable to make the inner skull surface slightly bigger than the brain envelope obtained in the previous step. Therefore, apply the *dilate3D* operation once or twice. Use the values 1 for nbours and 26 for nhoud in the first dilation and 1 and 18 in the second one, respectively.

7. Triangulation

Triangulate the resulting object with the *triangulate3D* operator. Use a sidelength of 5 to 6 mm. Check that the triangulation looks reasonable in the 3D viewing window.

8. Save the triangulation

Save the triangulated surface as a mesh into *bem/inner_skull.tri*. Select unit of measure as millimeters and employ the MRI coordinate system.

A.4 Other options

The *BrainSuite* software running under the Windows operating system can also be used for BEM mesh generation. This software, written by David W. Shattuck, is distributed as a collaborative project between the Laboratory of Neuro Imaging at the University of California Los Angeles (Director: Dr. Arthur W. Toga) and the Biomedical Imaging Research Group at the University of Southern California (Director: Dr. Richard M. Leahy). For further information, see <http://brainsuite.usc.edu/>. The conversion of BrainSuite tessellation files to MNE software compatible formats is covered in Section 5.4.6.

Note: If *BrainSuite* is employed in the segmentation, the *dfs* format surface files have to be converted into the *FreeSurfer* format with the utility *mne_convert_dfs*, see Section 5.4.6. Since corneate transformation is usually involved in this conversion, the correctness of the surfaces should be always checked with *tkmedit* or *MRILab*.

APPENDIX B Setup at the Martinos Center

This Appendix contains information which specific to the Martinos Center setup.

B.1 User environment

In the Martinos Center computer network, the present location of `<MEGsw>` is `/space/orsay/8/megdev/megsw`. A constantly changing development version is available at `/space/orsay/8/megdev/megsw-dev`. The previous release of the software is available at `/space/orsay/8/megdev/megsw-prev`. Most users at the Martinos Center use `tcsh` as their login shell.

B.2 Using Neuromag software

B.2.1 Using Neuromag software on the HP workstations

The complete set of Neuromag software is available on the HP-UX workstations. The programs can be accessed either locally at the workstation using the CDE Neuromag toolbox icons or from the command line, see Table B.1. The corresponding manuals are listed in Section B.2.

Module	Icon	Command line
xfit	SourceModelling	/neuro/bin/vue/xfit
xplotter	Plotting	/neuro/bin/vue/xplotter
graph	Graph	/neuro/bin/vue/graph
mrilab	MEG-MRI-integration	/neuro/bin/vue/mrilab
seglab	MRI-segmentation	/neuro/bin/vue/seglab
cliplab	GraphicsClipboard	/neuro/bin/vue/cliplab

Table B.1 Principal Neuromag software modules available on the HP workstations.

Module	Icon	pdf
xfit	SourceModelling-manual	XFit-5.4.pdf
xplotter	Plotting-manual	Xplotter.pdf
graph	Graph-users-guide Graph-reference-manual	GraphUsersGuide-2.93.pdf GraphReference-2.93.pdf
mrilab	MEG-MRI-integration-manual	Mrilab.pdf
seglab	MRI-segmentation-manual	Seglab.pdf
cliplab	GraphicsClipboard-manual	Cliplab.pdf

Table B.2 Neuromag software manuals available on the HP workstations.
All pdfs are located in /neuro/manuals.

B.2.2 Using Neuromag software on LINUX workstations

At present, alpha-test versions of xfit and xplotter are available on the LINUX workstations. To access them, you first need to issue the command:

```
setenv NEUROMAG_ROOT \
/space/orsay/8/megdev/megsw-neuromag
```

or in POSIX shell

```
export \
NEUROMAG_ROOT=/space/orsay/8/megdev/megsw-neuro-
mag
```

Thereafter, xplotter and xfit are accessible with the commands

```
$NEUROMAG_ROOT/bin/X11/xplotter
```

and

```
$NEUROMAG_ROOT/bin/X11/xfit
```

respectively.

More Neuromag software will be soon introduced for the LINUX platform.

B.3 Mature software

This Section contains documentation for software components, which are still available in the MNE software but have been replaced by new programs. In addition some now obsolete procedures using the Neuromag *graph* software are described here.

B.3.1 `mne_compute_mne`

This chapter contains information about the options accepted by the program `mne_compute_mne`, which is gradually becoming obsolete. All of its functions will be eventually included to `mne_make_movie`, see Section 6.5. At this time, `mne_compute_mne` is still needed to produce time-collapsed w files unless you are willing to write a Matlab script of your own for this purpose.

`mne_compute_mne` accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--inv <name>

Load the inverse operator decomposition from here.

--meas <name>

Load the MEG or EEG data from this file.

--set <number>

The data set (condition) number to load. The list of data sets can be seen, e.g., in `mne_analyze`, `mne_browse_raw`, and `xplotter`.

--bmin <time/ms>

Specifies the starting time of the baseline. In order to activate baseline correction, both `--bmin` and `--bmax` options must be present.

--bmax <time/ms>

Specifies the finishing time of the baseline.

--nave <value>

Specifies the number of averaged epochs in the input data. If the input data file is one produced by `mne_process_raw` or `mne_browse_raw`, the number of averages is correct in the file. However, if subtractions or some more complicated combinations of simple averages are produced, e.g., by using the `xplotter` software,

the number of averages should be manually adjusted. This is accomplished either by employing this flag or by adjusting the number of averages in the data file with help of `mne_change_nave`.

--snr <value>

An estimate for the amplitude SNR. The regularization parameter will be set as $\lambda = 1/\text{SNR}$. If the SNR option is absent, the regularization parameter will be estimated from the data. The regularization parameter will be then time dependent.

--snronly

Only estimate SNR and output the result into a file called SNR. Each line of the file contains three values: the time point in ms, the estimated SNR + 1, and the regularization parameter estimated from the data at this time point.

--abs

Calculate the absolute value of the current and the dSPM for fixed-orientation data.

--spm

Calculate the dSPM instead of the expected current value.

--chi2

Calculate an approximate χ^2_3 statistic instead of the F statistic. This is simply accomplished by multiplying the F statistic by three.

--sqrtF

Take the square root of the χ^2_3 or F statistic before outputting the stc file.

--collapse

Make all frames in the stc file (or the wfile) identical. The value at each source location is the maximum value of the output quantity at this location over the analysis period. This option is convenient for determining the correct thresholds for the rendering of the final brain-activity movies.

--collapse1

Make all frames in the stc file (or the wfile) identical. The value at each source location is the L_1 norm of the output quantity at this location over the analysis period.

--collapse2

Make all frames in the stc file (or the wfile) identical. The value at each source location is the L_2 norm of the output quantity at this location over the analysis period.

--SIcurrents

Output true current values in SI units (A/m). By default, the currents are scaled so that the maximum current value is set to 50 (Am).

--out <name>

Specifies the output file name. This is the ‘stem’ of the output file name. The actual name is derived by removing anything upto and including the last period from the end of <name>. According to the hemisphere, -lh or -rh is then appended. Finally, .stc or .w is added, depending on the output file type.

--wfiles

Use binary w-files in the output whenever possible. The noise-normalization factors can be always output in this format. The current estimates and dSPMs can be output as wfiles if one of the collapse options is selected.

--pred <name>

Save the predicted data into this file. This is a fif file containing the predicted data waveforms, see Section 6.2.7.

--outputnorm <name>

Output noise-normalization factors to this file.

--invnorm

Output inverse noise-normalization factors to the file defined by the --outputnorm option.

--dip <name>

Specifies a dipole distribution snapshot file. This is a file containing the current distribution at a time specified with the --diptime option. The file format is the ASCII dip file format produced by the Neuromag source modelling software (xfit). Therefore, the file can be loaded to the Neuromag MRILab MRI viewer to display the actual current distribution. This option is only effective if the --spm option is absent.

--diptime <time/ms>

Time for the dipole snapshot, see --dip option above.

--label <name>

Label to process. The label files are produced by tksurfer and specify regions of interests (ROIs). A label file name should end with -lh.label for left-hemisphere ROIs and with -rh.label for right-hemisphere ones. The corresponding output files are tagged with -lh-<data type>.amp and -rh-<data type>.amp, respectively. <data type> equals MNE for expected current data and spm for dSPM data. Each line of the output file contains the waveform of the output quantity at one of the source locations falling inside the ROI.

Note: The `--tmin` and `--tmax` options which existed in previous versions of `mne_compute_mne` have been removed. `mne_compute_mne` can now process only the entire averaged epoch.

B.3.2 Using graph for signal processing

B.3.2.1 Filtering

There is a customized setup for filtering with graph. To use it, proceed as follows:

1. Start the newest version of *graph* from one of the HP workstations by saying: `/neuro/bin/vue/graph`. Note that *graph* can be used remotely from the LINUX workstations by doing an `ssh` to a HP workstation first.
2. Load the setup `/neuro/lisp/local/graph-2.93/setup/vv_filter_raw.setup` into *graph*. This script assumes that there are 306 MEG channels and 64 EEG/EOG/ECG/EMG channels in the data.
3. Set the bandwidth in the MEG (and EEG) filters equal to the bandwidth you are going to use when the data are processed.
4. Select save filtered data from the User menu and select a destination file for the filtered result. A name of the form `<name>_filt_raw.fif` is recommended.
5. Wait patiently for the filtering to complete.

Important: Graph cannot be used to downsample the data because we want to include the digital trigger channel, which requires special treatment.

Tip: The output of the widget *raw-filter* is saved to the filtered file. If you have a channel setup different from the default one a modification of the setup is required. For example, if there are 128 EEG channels, you will have to re-establish the links from buffer all the way to the eeg filter and say `(select-to 'raw-out (meg 0 - 305) (eeg 0 -127) (stim 0))`.

B.3.2.2 Computing a noise-covariance matrix from raw data

The Neuromag *graph* software available on the HP workstations can be employed to calculate a noise-covariance matrix estimate from continuous data. The following steps are necessary:

1. Start the newest version of *graph* from one of the HP workstations by saying: `/neuro/bin/vue/graph`. Note that *graph* can be used remotely from the LINUX workstations by doing an `ssh` to a HP workstation first.
2. Load your favorite setup into *graph*.

3. Say (`require 'sensor-cov`) on the LISP command line.
4. Set the bandwidth in the MEG (and EEG) filters equal to the bandwidth you are going to use when the data are processed.
5. Say (`sensor-cov-on-widget '<widget> <start> <end>'`) on the LISP command line to calculate the noise-covariance matrix estimate. Here `<widget>` is the name of the widget containing the MEG (and EEG) channels after filtering while `<start>` and `<end>` specify the time interval (in seconds) employed to calculate the estimate. An interval at least 60 seconds long is recommended. You will be prompted for the noise-covariance matrix file. Both an fiff version (`<name>-cov.fif`) and a text version (`<name>-cov.txt`) is produced. The former will be used in the MNE computations while the latter can be loaded directly into MATLAB for inspection.

APPENDIX C Installation and configuration

The MNE software is distributed as a compressed tar archive. This Appendix describes the steps needed to install the software.

C.1 Installation

C.1.1 Create a directory to hold the software

Select a location for the MNE software. This most probably be an NFS file system available to all users in your network. Create the directory with the command:

```
mkdir <MEGswdir>
```

C.1.2 Unpack the archive

Go to the selected directory and unpack the tar archive:

```
cd <MEGswdir>  
tar zxvf <name of the tar archive>
```

C.2 Configuration

C.2.1 Set up a link to Matlab

The Matlab conversion routines make use of libraries distributed with Matlab. Therefore, a link to the Matlab installation root directory must be provided to the MNE software. If you have Matlab available, create the a symbolic link:

```
cd <MEGswdir>  
ln -s <MATLAB root directory> matlab
```

If you do not have Matlab available, create a dummy directory:

```
cd <MEGswdir>
mkdir matlab
```

Should you later acquire Matlab, remove the dummy directory and replace it with the proper symbolic link:

```
cd <MEGswdir>
rmdir matlab
ln -s <MATLAB root directory> matlab
```

Note: Without Matlab present, the utilities *mne_convert_mne_data*, *mne_convert_surface*, *mne_raw2mat*, and *mne_simu* will not work.

C.2.2 Run the configuration script

The configuration of MNE software is accomplished by running a single configuration script:

```
<MEGswdir>/mne/bin/admin/mne_setup_mne
```

Important: The *<MEGswdir>* must be the absolute pathname (starting with /) of the location of the MNE software as it will appear to the users of the server where the software was installed.

C.2.3 Modify the login scripts

Before using the MNE software each user must ‘source’ the script

```
<MEGswdir>/mne/setup/mne/mne_setup_analysis_csh
```

or

```
<MEGswdir>/mne/setup/mne/mne_setup_analysis_sh
```

in the shell. This task can be facilitated by setting up suitable aliases in the shell login scripts as described in Section 2.5.

C.2.4 Add open source software

The Mac OSX version of the MNE software relies on a few open source library packages. These packages can be installed with fink. To install fink on your computer, goto <http://sourceforge.net/> and look for Fink. There is also a graphical user interface called FinkCommander. The latter is convenient if you want to install open source packages for other purposes. For MNE, however, it is sufficient to install Fink.

The necessary packages can be easily accomplished by the script `mne_install_packages`, which super-user privileges. To run the script, say:

```
sudo <MEGswdir>/mne/bin/admin/mne_install_packages
```

If you accidentally run this script on a platform other than Mac OSX, nothing happens.

C.2.5 Optional: Remove unnecessary binaries

At present, the MNE software distribution contains binaries and shared libraries for LINUX and Mac OSX platforms. If you are always going to use your installed software on only one of these platforms, you can remove the binary code which is not going to be in use:

1. Source the correct setup script to your shell to get the `MNE_ROOT` environment variable set correctly.
2. Say `$MNE_ROOT/bin/admin/mne_remove_platform <platform>`, where `<platform>` is either `darwin` or `linux` depending on whether you want to remove the Mac OSX or LINUX binaries and libraries.

C.3 Licensing

Key modules of the MNE software require a valid license to run. The proper location of the license file is `<MEGswdir>/mne/setup/mne/LICENSES`. The licenses are free and thus the main purpose of the license is to allow the developers of MNE software to be aware of the number of current users of the software and their geographical locations. Furthermore, when obtaining a license, the licensee agrees to the MNE License Agreement included as Appendix E of this manual.

After you have filled out the registration form on our website (<http://www.nmr.mgh.harvard.edu/martinos/userInfo/data/sofMNE.php>), you will shortly receive an a confirmation email with information of the download location of the software as well as the license file as an attachment.. The name of the file is `LICENSES`. Save it to `<MEGswdir>/mne/setup/mne/LICENSES`.

The license file has a finite lifespan. If you have registered as a user of the MNE software, you will automatically receive an email at with a new license file at least two weeks before the expiration date of the current licens.

C.4 Obtain FreeSurfer

The MNE software relies on the FreeSurfer software for cortical surface reconstruction and other MRI-related tasks. Please consult the FreeSurfer home page site at <http://surfer.nmr.mgh.harvard.edu/>.

C.5 How to get started

After you have installed the software, a good place to start is to look at the manual:

- Source the correct setup script, see Section C.2.3, above and
- Say `mne_view_manual`.

Chapters of interest for a novice user include:

- Chapters 1 and 2 contain introduction to the software and setup instructions.
- Chapter 3 is an overview of the necessary steps to compute the cortically constrained minimum-norm solutions.
- Chapter 12 is a hands-on exercise demonstrating analysis of the sample data set.
- Chapter 13 contains a list of useful references for understanding the methods implemented in the MNE software.

APPENDIX D Release notes

This appendix contains a brief description of the changes in MNE software in each major release.

D.1 Release notes for MNE software 2.4

D.1.1 Manual

The manual has been significantly expanded and reorganized. Chapter 7 describing *mne_analyze* has been added. Chapter 12 contains instructions for analyzing the sample data set provided with the software. Useful background material is listed in Chapter 13. Almost all utility programs are now covered in the manual.

D.1.2 General software changes

The following overall changes have been made:

1. A forward solution library independent of Neuromag software was written.
2. The MEG sensor information is now imported from the coil definition file instead of being hardcoded in the software. For details, see Section 5.6.4.
3. CTF and 4D Neuroimaging sensors are now supported.
4. The number of Neuromag-based utilities was minimized.
5. The LINUX port of Neuromag software modules was completely separated from the MNE software and now resides under a separate directory tree.
6. Support for topologically connected source spaces was added, see Section 3.5.
7. A lot of bugs were fixed.

D.1.3 File conversion utilities

The following import utilities were added:

1. *mne_ctf2fiff* to convert CTF data to the fif format, see Section 9.2.2.
2. *mne_tufts2fiff* to convert EEG data from Tufts university to fif format, see Section 9.2.7.

The output of the Matlab conversion utilities was changed to use structures. For details, see Sections 9.7, 9.8, and 9.9.

Matlab tools to import and export w and stc files were added.

D.1.4 *mne_browse_raw*

Output of decimated and filtered data is now available. *mne_analyze* now fully supports 32-bit integer data found in CTF and new Neuromag raw data files.

D.1.5 *mne_analyze*

The following changes have been made in *mne_analyze*:

1. Curved and flat surface patches are now supported.
2. An iterative coordinate alignment procedure was added, see Sections 7.16 and Section 12.11.1.
3. Utility to view continuous HPI information was added, see Section 7.17.
4. Several small changes and bug fixes were done.

D.1.6 *mne_make_movie*

The only major change in *mne_make_movie* is the addition of support for curved and surface patches.

D.1.7 Averaging

The highly inefficient program *mne_grand_average* has been removed from the distribution and replaced with the combined use of *mne_make_movie* and a new averaging program *mne_average_estimates*, see Section 8.6.

D.2 Release notes for MNE software 2.5

D.2.1 Manual

The MNE Matlab toolbox is now covered in a separate chapter. Change bars are employed to indicate changes in the chapters that existed in the

previous version of the manual. Note that Chapter 10 describing the Matlab toolbox is totally new and change bars have not been used there. Furthermore, Appendix B now contains all the information specific to the Martinos Center.

D.2.2 `mne_browse_raw`

There are several improvements in the raw data processor *mne_browse_raw/mne_process_raw*:

- Possibility to delete and add channel selections interactively has been added. A nonstandard channel selection file can be now specified on the command line.
- Handling of CTF software gradient compensation has been added.
- The vertical scale of the digital trigger channel is now automatically set to accommodate the largest trigger value.
- It is now possible to load evoked-response data sets from files. Time scales of the evoked-response data and data averaged in *mne_browse_raw* can be now set from the scales dialog. Section 12.9 in Chapter 12 has been updated to employ *mne_browse_raw* in viewing the averages computed from the sample raw data set.
- It is now possible to create new SSP operators in *mne_browse_raw*, see Section 4.4.3.4.
- Listing of amplitude values have been added to both the strip-chart and topographical displays.
- Text format event files can now be loaded for easy inspection of rejected epochs, for example.
- Handling of derived channels has been added, see Sections 4.4.1.10 and 11.5.
- SSS information is now transferred to the covariance matrix output files.
- Neuromag processing history is included with the output files.

D.2.3 `mne_epochs2mat`

This new utility extracts epochs from a raw data file, applies a bandpass filter to them and outputs them in a format convenient for processing in Matlab, see Section 9.10.

D.2.4 `mne_analyze`

The following new features have been added:

1. Processing of raw data segment and easy switching between multiple evoked data sets (not in the manual yet).
2. Sketchy surface display mode for source spaces with selection triangulation information created with the `--ico` option to *mne_setup_source_space*.
3. Rotation of the coordinate frame in the coordinate system alignment dialog, see Section 7.16;
4. Several new graphics output file formats as well as automatic and snapshot output modes, see Section 7.8.7.
5. It is now possible to inquire timecourses from stc overlays. Both labels and surface picking are supported.
6. Added an option to include surface vertex numbers to the timecourse output, see Section 7.13.3.1.
7. Overlays matching the scalp surface can now be loaded, see Section 7.14.
8. The dipole display dialog has now control over the dipole display properties. Multiple dipoles can be now displayed, see Section 7.15.
9. Time stepping with cursor keys has been added.
10. Dynamic cursors have been added to the full view display.
11. The viewer display now automatically rotates to facilitate fiducial picking from the head surface.

D.2.5 *mne_ctf2fiff*

Correct errors in compensation channel information and compensation data output. The transformation between the CTF and Neuromag coordinate frames is now included in the output file.

D.2.6 *mne_make_movie*

Added the `--labelverts` option, see Section 6.5.8.

D.2.7 *mne_surf2bem*

Added the `--shift` option to move surface vertices outwards. Fixed some loopholes in topology checks. Also added the `--innershift` option to *mne_setup_forward_model*. For more information, see Sections 3.7 and 5.4.

D.2.8 *mne_forward_solution*

Added code to compute forward solutions for CTF data with software gradient compensation on.

D.2.9 `mne_inverse_operator`

The following changes have been made in *mne_inverse_operator*:

- Added options to regularize the noise-covariance matrix.
- Added correct handling of the rank-deficient covariance matrix resulting from the use of SSS.
- Additional projections cannot be specified if the noise covariance matrix was computed with projections on.
- Bad channels can be added only in special circumstances if the noise covariance matrix was computed with projections on.

D.2.10 `mne_compute_raw_inverse`

This utility is now documented in Section 6.6. The utility *mne_make_raw_inverse_operator* has been removed from the software.

D.2.11 Time range settings

The tools *mne_compute_raw_inverse*, *mne_convert_mne_data*, and *mne_compute_mne* no longer have command-line options to restrict the time range of evoked data input.

D.2.12 `mne_change_baselines`

It is now possible to process all data sets in a file at once. All processed data are stored in a single output file.

D.2.13 New utilities

D.2.13.1 `mne_show_fiff`

Replacement for the Neuromag utility *show_fiff*. This utility conforms to the standard command-line option conventions in MNE software. For details, see Section 11.3.

D.2.13.2 `mne_make_cor_set`

Replaces the functionality of the Neuromag utility *create_mri_set_simple* to create a fiff format description file for the FreeSurfer MRI data. This utility is called by the *mne_setup_mri* script.

D.2.13.3 mne_compensate_data

This utility applies or removes CTF software gradient compensation from evoked-response data, see Section 9.2.4.

D.2.13.4 mne_insert_4D_comp

This utility merges 4D Magnes compensation data from a text file and the main helmet sensor data from a fif file and creates a new fif file Section 9.2.5.

D.2.13.5 mne_ctf_dig2fiff

This utility reads a text format Polhemus data file, transforms the data into the Neuromag head coordinate system, and outputs the data in fif or hpts format.

D.2.13.6 mne_kit2fiff

The purpose of this new utility is to import data from the KIT MEG system, see Section 9.2.8.

D.2.13.7 mne_make_derivations

This new utility will take derivation data from a text file and convert it to fif format for use with *mne_browse_raw*, see Section 11.5.

D.2.14 BEM mesh generation

All information concerning BEM mesh generation has been moved to Appendix A. Utilities for BEM mesh generation using FLASH images have been added, see Section A.2.

D.2.15 Matlab toolbox

The MNE Matlab toolbox has been significantly enhanced. New features include:

- Basic routines for reading and writing fif files.
- High-level functions to read and write evoked-response fif data.
- High-level functions to read raw data.
- High-level routines to read source space information, covariance matrices, forward solutions, and inverse operator decompositions directly from fif files.

The Matlab toolbox is documented in Chapter 10.

The *mne_div_w* utility has been removed because it is now easy to perform its function and much more using the Matlab Toolbox.

APPENDIX E Licence agreement

This appendix includes the terms of the MNE software End-User License Agreement (EULA).

E.1 License agreement

THE GENERAL HOSPITAL CORPORATION

ACADEMIC RESEARCH USE

SOFTWARE LICENSE AGREEMENT FOR BINARY CODE

By downloading and/or using the MNE software which is the subject of this Agreement (the “Software”), you hereby accept and agree to all of the terms and conditions of this Agreement. As used in this Agreement, “you” means the individual who clicks the “I accept” button required as a condition of downloading the Software and the not-for-profit or governmental institution or entity which employs or is otherwise affiliated with such individual at the time of such download (the “Institution”).

1. *License Grant.* Subject to all of the terms and conditions of this Agreement, [The General Hospital Corporation, d/b/a Massachusetts General Hospital] [The Brigham and Women's Hospital, Inc.] (“Licensor”) hereby grants you a non-exclusive, non-transferable, non-sublicensable license under Licensor's rights in the Software to copy and use the binary code of the Software solely for research and educational purposes under your direction at the Institution (“Research and Educational Purposes,” which term shall include company sponsored research conducted by you in accordance with Institution's policies).
2. *No Transfer.* You may not sell, license, distribute, rent, lease, offer on an ASP or service bureau basis, grant a security interest in or otherwise transfer the Software to any third party or use the Software for any commercial purpose.
3. *Installation and Maintenance.* You are solely responsible for installing and maintaining the Software and for testing the Software for proper operation. Licensor shall have no obligation to provide any support, maintenance, corrections, debugging, improvements, modifications, upgrades or updates of the Software or notice of any of the forgoing, or

- otherwise assist Licensee in installing, configuring, integrating, understanding or using the Software.
4. *Attributions and Acknowledgments.* You must preserve and maintain all applicable attributions, copyright notices and licenses included in or applicable to the Software. You agree to provide acknowledgement of Licensor and its designated professional staff who participated in the creation of the Software in publications or presentations in accordance with standard academic practice, provided that you may not otherwise use Licensor's name, logos or trademarks or the name of any individual associated with Licensor, or of any funding agency, in any advertising, promotional or sales material or publicity or in any document employed to obtain funds or financing, or to endorse or promote any research results or products related to or arising from the Software, without the prior written consent of a person authorized to make such consent.
 5. *Third Party Software.* This Agreement does not grant any rights with respect to any third party software, except those rights that Licensor has been authorized by a third party to grant to you, and accordingly you are solely responsible for obtaining any permissions from third parties which are necessary to use and copy the Software.
 6. *Compliance with Law.* You must comply with all applicable governmental laws, regulations and orders, including without limitation those relating to export and import control, in exercising your rights under this Agreement.
 7. *Termination.* You may terminate this Agreement at any time by destroying all copies of the Software. Licensor may terminate this Agreement at any time by providing notice to you of such termination. Any use or copying of the Software by you which is inconsistent with the terms and conditions of this Agreement shall automatically render this Agreement null and void and terminate the license granted hereunder. Upon any termination of this Agreement you must stop using the Software and return or destroy all copies of the Software, including any portion thereof.
 8. *DISCLAIMERS.* YOU ACKNOWLEDGE THAT THE SOFTWARE HAS BEEN DESIGNED FOR RESEARCH PURPOSES ONLY AND HAS NOT BEEN REVIEWED OR APPROVED BY THE FOOD AND DRUG ADMINISTRATION OR BY ANY OTHER AGENCY, AND YOU FURTHER ACKNOWLEDGE THAT CLINICAL APPLICATIONS ARE NEITHER RECOMMENDED NOR ADVISED. The Software is provided "AS IS", is experimental, may contain errors and is subject to further development and revision. Licensor does not guarantee the accuracy of the Software, or of any results or data. LICENSOR SPECIFICALLY DISCLAIMS ALL EXPRESS AND IMPLIED WARRANTIES OF ANY KIND INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT
 9. *LIMITATION OF LIABILITY.* IN NO EVENT SHALL LICENSOR OR ANY OF ITS TRUSTEES, DIRECTORS, OFFICERS, MEDICAL OR PROFESSIONAL STAFF, EMPLOYEES, STUDENTS OR AGENTS ("LICENSOR'S PERSONNEL") BE LIABLE TO ANY

PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY ARISING IN ANY WAY RELATED TO THE SOFTWARE, EVEN IF LICENSOR OR ANY OF LICENSOR'S PERSONNEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCEPT TO THE EXTENT PROHIBITED BY LAW OR REGULATION, YOU ASSUME ALL RISK AND LIABILITY FOR YOUR USE AND COPYING OF THE SOFTWARE, AND AGREE TO INDEMNIFY AND HOLD HARMLESS LICENSOR AND EACH OF LICENSOR'S PERSONNEL FROM AND AGAINST ANY AND ALL CLAIMS, SUITS, ACTIONS, DEMANDS AND JUDGMENTS ARISING THEREFROM

10. *U.S. Government Rights.* For Software supported by federal funding, the license granted under this Agreement is subject to the rights, conditions and limitations imposed by U.S. law including without limitation 35 U.S.C. § 202 et seq. and regulations pertaining thereto.
11. *General.* This Agreement constitutes the entire understanding between you and Licensor with respect to the subject matter hereof, and supercedes any prior or contemporaneous oral or written agreements with respect thereto. Sections 2, 4, 6, 8, 9, 11 shall survive any termination of this Agreement. This Agreement may be modified or amended only in a writing signed by duly authorized representatives of both Parties hereto. The invalidity or unenforceability of any provision of this Agreement shall not affect any other provision hereof. This Agreement and the license granted hereunder may not be assigned

