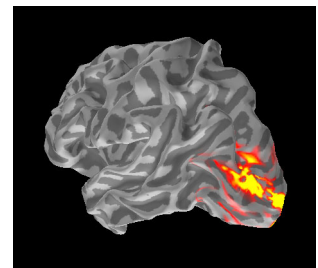
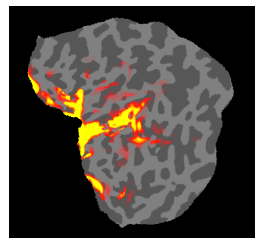
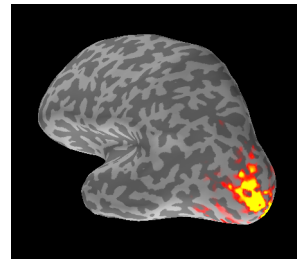
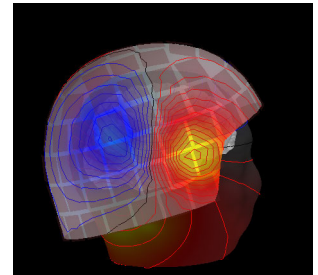


MNE software User's Guide

Version 2.7.3

November 2010



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
12th edition	MSH-MNE	2.6	March 2009
13th edition	MSH-MNE	2.7	December 2009
14th edition	MSH-MNE	2.7.1	March 2010
15th edition	MSH-MNE	2.7.2	October 2010
16th edition	MSH-MNE	2.7.3	November 2010

Contents

Chapter 1	Introduction	9
Chapter 2	Overview	11
2.1	List of components	11
2.2	File formats	16
2.3	Conventions	16
2.4	User environment	16
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	24
	Setting up the triangulation files	24
3.7	Setting up the boundary-element model	25
3.8	Setting up the MEG/EEG analysis directory	28
3.9	Preprocessing the raw data	29
	Cleaning the digital trigger channel	29
	Fixing channel information	30
	Designating bad channels	30
	Downsampling the MEG/EEG data	31
	Off-line averaging	31
3.10	Aligning the coordinate frames	31
3.11	Computing the forward solution	32
3.12	Setting up the noise-covariance matrix	35
3.13	Calculating the inverse operator decomposition	36
3.14	Analyzing the data	39
Chapter 4	Processing raw data	41
4.1	Overview	41
4.2	Command-line options	41
	Common options	41
	Interactive mode options	43
	Batch-mode options	44
4.3	The user interface	48
4.4	The File menu	48
	Open	48
	Open evoked	50

Save	50
Change working directory	51
Read projection	51
Save projection	51
Apply bad channels	51
Load events (text)	51
Load events (fif)	51
Save events (text)	52
Save events (fif)	52
Load derivations	52
Save derivations	53
Load channel selections	53
Save channel selections	53
Quit	53
4.5 The Adjust menu	53
Filter	53
Scales	54
Colors	57
Derivations	57
Selection	58
Full view layout	61
Projection	62
Compensation	62
Averaging preferences	63
4.6 The Process menu	64
Averaging	64
Estimation of a covariance matrix	65
Estimation of a covariance matrix from raw data	65
Creating a new SSP operator	65
4.7 The Windows menu	66
4.8 The Help menu	67
4.9 The raw data display	68
Browsing data	69
4.10 Events and annotations	70
Overview	70
The event list	71
Loading and saving event files	71
Defining annotated events	72
Event files	73
4.11 The tool bar	74
4.12 Topographical data displays	75
4.13 Description files for off-line averaging	76
Overall format	76
Common parameters	77
Category definition	79
4.14 Description files for covariance matrix estimation	81
Overall format	81
Common parameters	81
Covariance definitions	83
4.15 Managing averages	85

4.16	The Signal-Space Projection (SSP) method	86
	General concepts	86
	Estimation of the noise subspace	87
	EEG average electrode reference	88
4.17	Covariance matrix estimation	88
	Continuous raw data	88
	Epochs	89
	Combination of covariance matrix estimates	90
	SSP information included with covariance matrices	91
4.18	Interacting with mne_analyze	91
Chapter 5 The forward solution		93
5.1	Overview	93
5.2	MEG/EEG and MRI coordinate systems	93
5.3	The head and device coordinate systems	97
5.4	Creating a surface-based source space	98
5.5	Creating a volumetric or discrete source space	99
5.6	Creating the BEM meshes	101
	Command-line options	101
	Surface options	102
	Tessellation file format	103
	Topology checks	104
5.7	Computing the BEM geometry data	104
5.8	Coil geometry information	105
	The sensor coordinate system	105
	Calculation of the magnetic field	106
	Implemented coil geometries	107
	The coil definition file	111
	Creating the coil definition file	113
5.9	Computing the forward solution	113
	Purpose	113
	Command line options	113
	Implementation of software gradient compensation	116
	The EEG sphere model definition file	116
	EEG forward solution in the sphere model	117
	Field derivatives	117
5.10	Averaging forward solutions	118
	Purpose	118
	Command line options	118
Chapter 6 The current estimates		121
6.1	Overview	121
6.2	Minimum-norm estimates	121
	The linear inverse operator	121
	Regularization	122
	Whitening and scaling	122
	Regularization of the noise-covariance matrix	123
	Computation of the solution	124

Noise normalization	125
Predicted data	126
Cortical patch statistics	126
The orientation constraints	126
Depth weighting	127
fMRI-guided estimates	127
6.3 Effective number of averages	128
6.4 Inverse-operator decomposition	129
6.5 Producing movies and snapshots	133
General options	133
Input files	133
Times and baseline	133
Options controlling the estimates	135
Visualization options	135
Thresholding	137
Output files	138
Label processing	140
Using stc file input	141
6.6 Computing inverse from raw and evoked data	141
Command-line options	141
Implementation details	144

Chapter 7 Interactive analysis	147
---	------------

7.1 Overview	147
7.2 Command line options	147
7.3 The main window	149
7.4 The menus	150
The File menu	150
The Adjust menu	151
The View menu	152
The Labels menu	152
The Dipoles menu	153
The Help menu	154
7.5 Loading data	154
7.6 Loading epochs from a raw data file	156
7.7 Data displays	157
The topographical display	157
The sample channel display	158
Scale settings	159
7.8 The surface display	160
The surface selection dialog	161
The patch selection dialog	162
Controlling the surface display	163
Selecting vertices	164
Defining viewing orientations	165
Adjusting lighting	166
Producing output files	167
Image output modes	168
7.9 Morphing	169

7.10	The viewer	170
	Overview	170
	Viewer options	173
7.11	Magnetic field and electric potential maps	175
	Overview	175
	Technical description	175
	Field mapping preferences	177
7.12	Working with current estimates	178
	Preferences	178
	The SNR display	181
7.13	Inquiring timecourses	182
	Timecourses at vertices	182
	Timecourses at labels	182
	The timecourse manager	183
	Label timecourse files	184
	Creating new label files	185
7.14	Overlays	186
7.15	Fitting current dipoles	188
	Dipole fitting parameters	188
	The dipole fitting algorithm	190
	The dipole list	192
	Channel selections	194
7.16	Coordinate frame alignment	194
	Using a high-resolution head surface tessellations	197
	Using fiducial points identified by other software	198
7.17	Viewing continuous HPI data	199
7.18	Working with the MRI viewer	201
7.19	Working with the average brain	202
7.20	Compatibility with cliplab	203
7.21	Visualizing the head position	204
Chapter 8 Morphing and averaging		207
8.1	Overview	207
8.2	The morphing maps	207
8.3	About smoothing	208
8.4	Precomputing the morphing maps	209
8.5	Morphing label data	210
8.6	Averaging	211
	Overview	211
	The averager	212
	The description file	212
Chapter 9 Data conversion		215
9.1	Overview	215
9.2	Importing data from other MEG/EEG systems	215
	Importing 4-D Neuroimaging data	215
	Importing CTF data	216
	Importing CTF Polhemus data	218

Applying software gradient compensation	219
Importing Magnes compensation channel data	220
Creating software gradient compensation data	221
Importing KIT MEG system data	222
Importing EEG data saved in the EDF, EDF+, or BDF format	225
Overview	225
Using mne_edf2fiff	226
Post-conversion tasks	227
Importing EEG data saved in the Tufts University format	228
Importing BrainVision EEG data	229
Converting eXimia EEG data	230
9.3 Converting digitization data	230
The hpts format	231
9.4 Converting volumetric data into an MRI overlay	232
9.5 Listing source space data	233
9.6 Listing BEM mesh data	234
9.7 Converting surface data between different formats	235
command-line options	235
9.8 Converting MRI data into the fif format	238
9.9 Collecting coordinate transformations into one file	239
9.10 Converting an ncov covariance matrix file to fiff	240
9.11 Converting a lisp covariance matrix to fiff	240
9.12 The MNE data file conversion tool	241
Command-line options	241
Guide to combining options	244
Matlab data structures	245
9.13 Converting raw data to Matlab format	247
Command-line options	248
Matlab data structures	248
9.14 Converting epochs to Matlab format	250
Command-line options	251
The binary epoch data file	253
Matlab data structures	254

Chapter 10 The Matlab toolbox	257
--------------------------------------	------------

10.1 Overview	257
10.2 Some data structures	265
10.3 On-line documentation for individual routines	279

Chapter 11 Miscellaneous utilities	281
---	------------

11.1 Overview	281
11.2 Finding software versions	281
11.3 Listing contents of a fif file	281
11.4 Data file modification utilities	282
Designating bad channels: mne_mark_bad_channels	282
Fixing the encoding of the trigger channel: mne_fix_stim14	283
Updating EEG location info: mne_check_eeg_locations	283
Updating magnetometer coil types: mne_fix_mag_coil_types	284

Modifying channel names and types: mne_rename_channels	284
Modifying trigger channel data: mne_add_triggers	286
Purpose	286
Command line options	286
Removing identifying information	287
Copying the processing history	288
11.5 Creating a derivation file	288
Purpose	288
Command-line options	289
Derivation file formats	289
11.6 Creating a custom EEG layout	291
Purpose	291
11.7 Adding neighborhood information to source spaces	292
Purpose	292
Command line options	292
Computational details	293
11.8 Converting covariance data into an SSP operator	294
Purpose	294
Command line options	294
11.9 Fitting a sphere to a surface	295
Purpose	295
Command line options	295
11.10 Computing sensitivity maps	296
Purpose	296
Command line options	296
Available sensitivity maps	297
11.11 Transforming locations	298
Purpose	298
Command line options	298
11.12 Inquiring and changing baselines	299
11.13 Data simulator	300
Purpose	300
Command-line options	300
Noise simulation	302
Simulated data	302
Source waveform expressions	303
11.14 Converting parcellation data into labels	305
 Chapter 12 The sample data set	 307
12.1 Purpose	307
12.2 Overview	307
12.3 Setting up	308
12.4 Contents of the data set	309
12.5 Setting up subject-specific data	310
Structural MRIs	310
Source space	310
Boundary-element models	311
12.6 Setting up a custom EEG layout	311
12.7 Previewing the data	311

12.8	Off-line averaging	313
	Using the averaging script interactively	313
	Using the averaging script in batch mode	313
12.9	Viewing the off-line average	314
	Loading the averages	314
	Inspecting the auditory data	314
	Inspecting the visual data	315
12.10	Computing the noise-covariance matrix	315
12.11	MEG-MRI coordinate system alignment	316
	Initial alignment	316
	Refining the coordinate transformation	317
	Saving the transformation	318
12.12	The forward solution	318
12.13	The inverse operator decomposition	319
12.14	Interactive analysis	320
	Getting started	320
	Load surfaces	320
	Load the data	320
	Show field and potential maps	320
	Show current estimates	321
	Labels and timecourses	321
	Morphing	321
Chapter 13 Useful reading		323
<hr/>		
13.1	General MEG reviews	323
13.2	Cortical surface reconstruction and morphing	323
13.3	Forward modeling	323
13.4	Signal-space projections	324
13.5	Minimum-norm estimates	324
13.6	fMRI-weighted estimates	325
Appendix A Creating the BEM meshes		327
<hr/>		
A.1	Using the watershed algorithm	327
A.2	Using FLASH images	328
	Organizing MRI data into directories	328
	Creating the surface tessellations	329
	Inspecting the meshes	330
A.3	Using seglab	331
A.4	Using BrainSuite	332
Appendix B Setup at the Martinos Center		335
<hr/>		
B.1	User environment	335
B.2	Using Neuromag software	335
	Software overview	335
	Using MRILab for coordinate system alignment	336
B.3	Mature software	337
	mne_compute_mne	337

Appendix C Installation and configuration	341
C.1 System requirements	341
C.2 Installation	341
Download the software	341
Installing from a compressed tar archive	341
Installing from a Mac OSX disk image	342
Additional software	342
Testing the performance of your OpenGL graphics	342
C.3 Obtain FreeSurfer	343
C.4 How to get started	343
 Appendix D Release notes	 345
D.1 Release notes for MNE software 2.4	345
Manual	345
General software changes	345
File conversion utilities	345
mne_browse_raw	346
mne_analyze	346
mne_make_movie	346
Averaging	346
D.2 Release notes for MNE software 2.5	346
Manual	346
mne_browse_raw	347
mne_epochs2mat	347
mne_analyze	347
mne_ctf2fiff	348
mne_make_movie	348
mne_surf2bem	348
mne_forward_solution	348
mne_inverse_operator	349
mne_compute_raw_inverse	349
Time range settings	349
mne_change_baselines	349
New utilities	349
mne_show_fiff	349
mne_make_cor_set	349
mne_compensate_data	350
mne_insert_4D_comp	350
mne_ctf_dig2fiff	350
mne_kit2fiff	350
mne_make_derivations	350
BEM mesh generation	350
Matlab toolbox	350
D.3 Release notes for MNE software 2.6	351
Manual	351
Command-line options	351
Changes to existing software	351
mne_add_patch_info	351

mne_analyze	351
mne_average_forward_solutions	352
mne_browse_raw and mne_process_raw	352
mne_compute_raw_inverse	353
mne_convert_surface	353
mne_dump_triggers	353
mne_epochs2mat	354
mne_forward_solution	354
mne_list_bem	354
mne_make_cor_set	354
mne_make_movie	354
mne_make_source_space	354
mne_mdip2stc	354
mne_project_raw	354
mne_rename_channels	355
mne_setup_forward_model	355
mne_simu	355
mne_transform_points	355
Matlab toolbox	355
New utilities	355
mne_collect_transforms	355
mne_convert_dig_data	355
mne_edf2fiff	356
mne_brain_vision2fiff	356
mne_anonymize	356
mne_opengl_test	356
mne_volume_data2mri	356
mne_volume_source_space	356
mne_copy_processing_history	356
D.4 Release notes for MNE software 2.7	357
Software engineering	357
Matlab tools	357
mne_browse_raw	357
mne_analyze	358
Miscellaneous	358
D.5 Release notes for MNE software 2.7.1	358
mne_analyze	358
mne_browse_raw	359
D.6 Release notes for MNE software 2.7.2	359
mne_add_patch_info	359
Matlab toolbox	359
Miscellaneous	359
D.7 Release notes for MNE software 2.7.3	359
Miscellaneous	359

Appendix E Licence agreement 361

E.1 License agreement	361
------------------------------	------------

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 software package will 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 tools 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/share/doc/MNE-manual-<version>.pdf`. For the present manual, `<version> = 2.7`. For definition of the `MNE_ROOT` environment variable, see Section 2.4.

We want to thank all MNE Software users at the Martinos Center and in other institutions 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 *Center for Functional Neuroimaging Technologies* P41RR14075-06, the NIH grants 1R01EB009048-01, R01 EB006385-A101, 1R01 HD40712-A1, 1R01 NS44319-01, and 2R01 NS37462-05, ell as by Department of Energy under Award Number DE-FG02-99ER62764 to The MIND Institute.

CHAPTER 2 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 9.12.
<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 5.9.
<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 5.4.

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>	A convenience script to create the <i>fif</i> files describing the anatomical MRI data, see Section 3.4.
<i>mne_setup_source_space</i>	A 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 neighborhood 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 11.14.
<i>mne_anonymize</i>	Remove subject-specific information from a fif data file, see Section 11.4.7.
<i>mne_average_forward_solutions</i>	Calculate an average of forward solutions, see Section 5.10.
<i>mne_brain_vision2fiff</i>	Convert EEG data from BrainVision format to fif format, see Section 9.2.10.

Table 2.2 Utility programs.

Name	Purpose
<i>mne_change_baselines</i>	Change the dc offsets according to specifications given in a text file, see Section 11.12.
<i>mne_change_nave</i>	Change the number of averages in an evoked-response data file. This is often necessary if the file was derived from several files.
<i>mne_check_eeg_locations</i>	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.
<i>mne_check_surface</i>	Check the validity of a FreeSurfer surface file or one of the surfaces within a BEM file. This program simply checks for topological errors in surface files.
<i>mne_collect_transforms</i>	Collect coordinate transformations from several sources into a single fif file, see Section 9.9.
<i>mne_compensate_data</i>	Change the applied software gradient compensation in an evoked-response data file, see Section 9.2.4.
<i>mne_convert_lspcov</i>	Convert the LISP format noise covariance matrix output by graph into fif, see Section 9.11.
<i>mne_convert_ncov</i>	Convert the ncov format noise covariance file to fif, see Section 9.10.
<i>mne_convert_surface</i>	Convert FreeSurfer and text format surface files into Matlab mat files, see Section 9.7.
<i>mne_cov2proj</i>	Pick eigenvectors from a covariance matrix and create a signal-space projection (SSP) file out of them, see Section 11.9.
<i>mne_create_comp_data</i>	Create a fif file containing software gradient compensation information from a text file, see Section 9.2.6.
<i>mne_ctf2fiff</i>	Convert a CTF ds folder into a fif file, see Section 9.2.2.
<i>mne_ctf_dig2fiff</i>	Convert text format digitization data to fif format, see Section 9.2.3.
<i>mne_dicom_essentials</i>	List essential information from a DICOM file. This utility is used by the script <i>mne_organize_dicom</i> , see Section A.2.1.
<i>mne_edf2fiff</i>	Convert EEG data from the EDF/EDF+/BDF formats to the fif format, see Section 9.2.8.
<i>mne_epochs2mat</i>	Apply bandpass filter to raw data and extract epochs for subsequent processing in Matlab, see Section 9.14.

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_eximia2fiff</code>	Convert EEG data from the Nexstim eXimia system to fif format, see Section 9.2.11.
<code>mne_fit_sphere_to_surf</code>	Fit a sphere to a surface given in either fif or FreeSurfer format, see Section 11.9.
<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 9.6.
<code>mne_list_coil_def</code>	Create the coil description file. This is run automatically at when the software is set up, see Section 5.8.5.
<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 9.5.
<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 from COR or mgh/mgz format MRI data, see Section 3.4. The <code>mne_make_cor_set</code> utility is described in Section 9.8.
<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.

Table 2.2 Utility programs.

Name	Purpose
<i>mne_mark_bad_channels</i>	Update the list of unusable channels in a data file, see Section 11.4.1.
<i>mne_morph_labels</i>	Morph label file definitions between subjects, see Section 8.5.
<i>mne_organize_dicom</i>	Organized DICOM MRI image files into directories, see Section A.2.1.
<i>mne_prepare_bem_model</i>	Perform the geometry calculations for BEM forward solutions, see Section 5.7.
<i>mne_process_stc</i>	Manipulate stc files.
<i>mne_raw2mat</i>	Convert raw data into a Matlab file, see Section 9.13.
<i>mne_rename_channels</i>	Change the names and types of channels in a fif file, see Section 11.4.5.
<i>mne_sensitivity_map</i>	Compute a sensitivity map and output the result in a w-file, see Section 11.10.
<i>mne_sensor_locations</i>	Create a file containing the sensor locations in text format.
<i>mne_show_fiff</i>	List contents of a fif file, see Section 11.3
<i>mne_simu</i>	Simulate MEG and EEG data, see Section 11.13.
<i>mne_smooth</i>	Smooth a w or stc file.
<i>mne_surf2bem</i>	Create a <i>fif</i> file describing the triangulated compartment boundaries for the boundary-element model (BEM), see Section 5.6.
<i>mne_toggle_skips</i>	Change data skip tags in a raw file into ignored skips or vice versa.
<i>mne_transform_points</i>	Transform between MRI and MEG head coordinate frames, see Section 11.11.
<i>mne_tufts2fiff</i>	Convert EEG data from the Tufts University format to fif format, see Section 9.2.9.
<i>mne_view_manual</i>	Starts a PDF reader to show this manual from its standard location.
<i>mne_volume_data2mri</i>	Convert volumetric data defined in a source space created with <i>mne_volume_source_space</i> into an MRI overlay, see Section 9.4.
<i>mne_volume_source_space</i>	Make a volumetric source space, see Section 5.5.
<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 MNE software employs the `fif` file format whenever possible. New tags have been added to incorporate information specific to the calculation of cortically contained source estimates. 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 and thus omit the backslashes. The order of options is irrelevant. Entries to be typed literally are shown like `this`. *Italicized* text indicates conceptual entries. For example, `<dir>` indicates a directory name.

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 double-dash (`--`) option convention, *i.e.*, the option names are preceded by two dashes.

Most of the programs have two common options to obtain general information:

--help

Prints concise usage information.

--version

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

2.4 User environment

The system-dependent location of the MNE Software will be here referred to by the environment variable `MNE_ROOT`. There are two scripts for setting up user environment so that the software can be used conveniently:

`$MNE_ROOT/bin/mne_setup_sh`

and


```
$MNE_ROOT/bin/mne_setup
```

compatible with the POSIX and csh/tcsh shells, respectively. Since the scripts set environment variables they should be ‘sourced’ to the present shell. You can find which type of a shell you are using by saying

```
echo $SHELL
```

If the output indicates a POSIX shell (bash or sh) you should issue the three commands:

```
export MNE_ROOT=<MNE>
export MATLAB_ROOT=<Matlab>
. $MNE_ROOT/bin/mne_setup_sh
```

with <MNE> replaced by the directory where you have installed the MNE software and <Matlab> is the directory where Matlab is installed. If you do not have Matlab, leave MATLAB_ROOT undefined. If Matlab is not available, the utilities *mne_convert_mne_data*, *mne_epochs2mat*, *mne_raw2mat*, and *mne_simu* will not work.

For csh/tcsh the corresponding commands are:

```
setenv MNE_ROOT <MNE>
setenv MATLAB_ROOT <Matlab>
source $MNE_ROOT/bin/mne_setup
```

For BEM mesh generation using the watershed algorithm or on the basis of multi-echo FLASH MRI data (see Appendix A) and for accessing the *tkmedit* program from *mne_analyze*, see Section 7.18, 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.

The environment variables relevant to the MNE software are listed in 2.3

Name of the variable	Description
MNE_ROOT	Location of the MNE software, see above
FREESURFER_HOME	Location of the FreeSurfer software. Needed during FreeSurfer reconstruction and if the FreeSurfer MRI viewer is used with <i>mne_analyze</i> , see Section 7.18.
SUBJECTS_DIR	Location of the MRI data

Table 2.3 Environment variables

Name of the variable	Description
SUBJECT	Name of the current subject
MNE_TRIGGER_CH_NAME	Name of the trigger channel in raw data, see Section 4.2.1.
MNE_TRIGGER_CH_MASK	Mask to be applied to the trigger channel values, see Section 4.2.1.

Table 2.3 Environment variables

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.

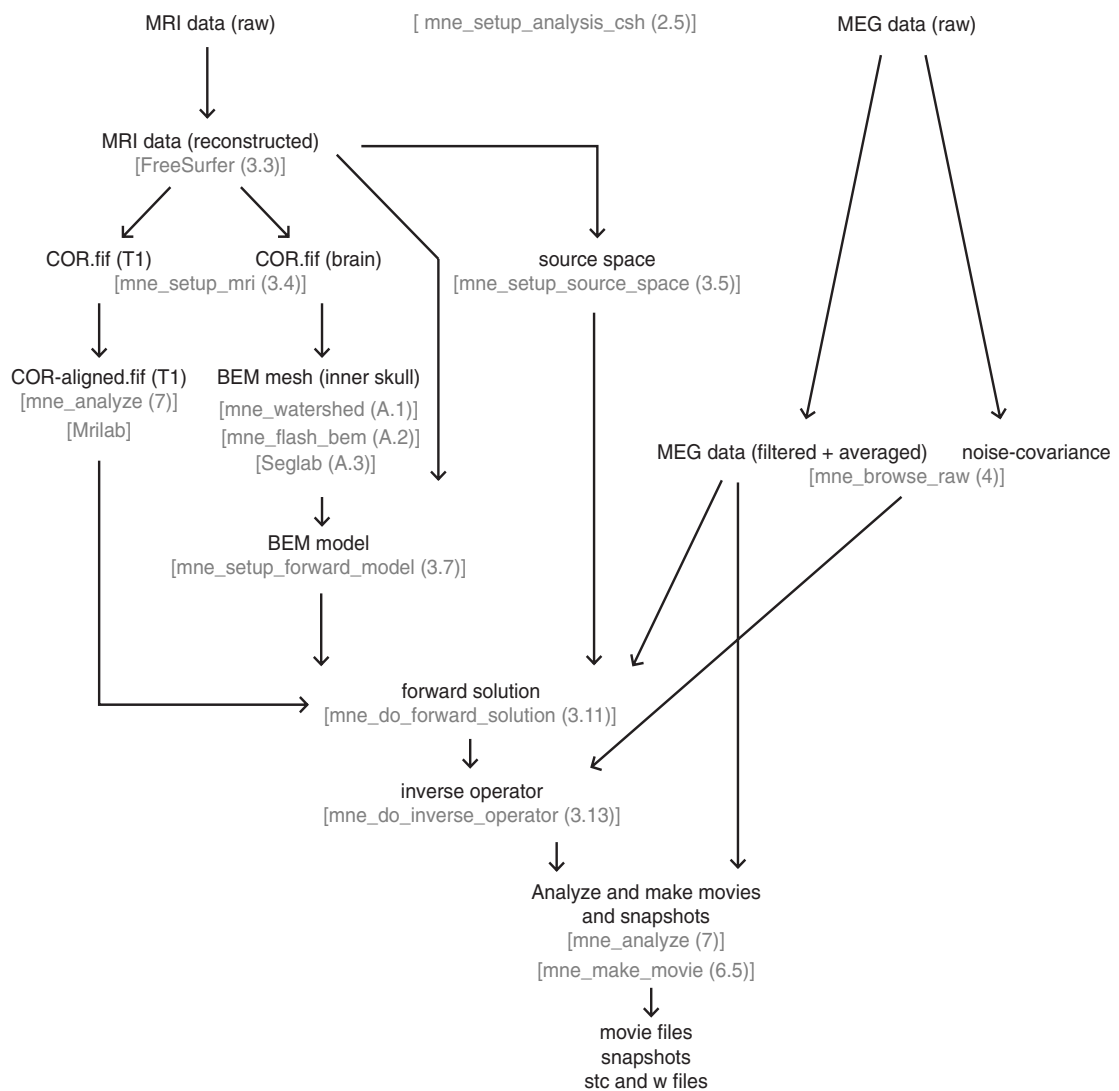


Figure 3.1 Workflow of the MNE software. References in parenthesis indicate sections and chapters of this manual.

3.2 Selecting the subject

Before starting the data 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 common to many MNE programs and scripts. 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`.

It is also recommended that the *FreeSurfer* environment is set up before using the MNE software.

3.3 Cortical surface reconstruction with FreeSurfer

The first processing stage is the creation of various surface reconstructions with *FreeSurfer*. The recommended *FreeSurfer* workflow is summarized on the FreeSurfer wiki pages: <https://surfer.nmr.mgh.harvard.edu/fswiki/RecommendedReconstruction>. Please refer to the *FreeSurfer* wiki pages (<https://surfer.nmr.mgh.harvard.edu/fswiki/>) and other *FreeSurfer* documentation for more information.

Important: Only the latest (4.0.X and later) *FreeSurfer* distributions contain a version of *tkmedit* which is compatible with *mne_analyze*, see Section 7.18.

3.4 Setting up the anatomical MR images for MRILab

If you have the Neuromag software installed, the Neuromag MRI viewer, MRILab, can be used to access the MRI slice data created by *FreeSurfer*. In addition, the Neuromag MRI directories can be used for storing the MEG/MRI coordinate transformations created with *mne_analyze*, see Section 7.16. During the computation of the forward solution, *mne_do_forward_solution* searches for the MEG/MRI coordinate in the Neuromag MRI directories, see Section 3.11. The `fif` files created by *mne_setup_mrit* can be loaded into Matlab with the *fiff_read_mri* function, see Chapter 10.

These functions require 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>-neuromag/slices` and `mri/<name>-neuromag/sets`. If the input data set is in COR format, `mne_setup_mri` makes symbolic links from the COR files in the directory `mri/<name>` into `mri/<name>-neuromag/slices`, and creates a corresponding `fif` file `COR.fif` in `mri/<name>-neuromag/sets`. This “description file” contains references to the actual MRI slices.

If the input MRI data are stored in the newer `mgz` format, the file created in the `mri/<name>-neuromag/sets` directory will include the MRI pixel data as well. If available, the coordinate transformations to allow conversion between the MRI (surface RAS) coordinates and MNI and FreeSurfer Talairach coordinates are copied to the MRI description file. `mne_setup_mri` invokes `mne_make_cor_set`, described in Section 9.8 to convert the data.

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.

Tip: If the name specified with the `--mri` option contains a slash, the MRI data are accessed from the directory specified and the `SUBJECT` and `SUBJECTS_DIR` environment variables as well as the `--subject` option are ignored.

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 convenience script `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.

--morph <name>

Name of a subject in SUBJECTS_DIR. If this option is present, the source space will be first constructed for the subject defined by the --subject option or the SUBJECT environment variable and then morphed to this subject. This option is useful if you want to create a source spaces for several subjects and want to directly compare the data across subjects at the source space vertices without any morphing procedure afterwards. The drawback of this approach is that the spacing between source locations in the “morph” subject is not going to be as uniform as it would be without morphing.

--spacing <spacing/mm>

Specifies the grid spacing for the source space in mm. If not set, a default spacing of 7 mm is used. Either the default or a 5-mm spacing is recommended.

--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 ($\langle number \rangle > 0$) or an octahedron ($\langle number \rangle < 0$). 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 $\langle number \rangle$ 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 MRILab compatible ascii format.
3. `<subject>-<spacing>-lh.dip` and `<subject>-<spacing>-rh.dip` containing the source space points in MRILab 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.

Tip: After the geometry is set up it is possible 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, a 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 triangulation data in text. Before proceeding to the creation of the boundary element model, standard files (or symbolic links created with the `ln -s` command) 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: 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.

Tip: The triangulation files can include name of the subject as a prefix `<subject name>-.e.g., duck-inner_skull.surf`.

Tip: The `mne_convert_surface` utility described in Section 9.7 can be used to convert text format triangulation files into the *FreeSurfer* surface format.

Important: “Aliases” created with the Mac OSX finder are not equivalent to symbolic links and do not work as such for the UNIX shells and MNE programs.

3.7 Setting up the boundary-element model

This stage sets up the subject-dependent data for computing the forward solutions:

1. The `fif` format boundary-element model geometry file is created. This step also checks that the input surfaces are complete and that they are topologically correct, *i.e.*, that the surfaces do not intersect and that the surfaces are correctly ordered (outer skull surface inside the scalp and inner skull surface inside the outer skull). 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. Text files containing the boundary surface vertex coordinates are created.
3. The geometry-dependent BEM solution data are computed. This step can be optionally omitted. This step takes several minutes to complete.

This step assigns the conductivity values to the BEM compartments. For the scalp and the brain compartments, the default is 0.3 S/m. The default skull conductivity is 50 times smaller, *i.e.*, 0.006 S/m. Recent publications, see Section 13.3, report a range of skull conductivity ratios ranging from 1:15 (Oostendorp *et al.*, 2000) to 1:25 - 1:50 (Slew *et al.*, 2009, Gonçalves *et al.*, 2003). The MNE default ratio 1:50 is based on the typical values reported in (Gonçalves *et al.*, 2003), since their approach is based on comparison of SEF/SEP measurements in a BEM model. The vari-

ability across publications may depend on individual variations but, more importantly, on the precision of the skull compartment segmentation.

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 results in a long computation time even in a workstation with generous amounts of memory. 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 0.006 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.

--outershift <value/mm>

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

--scalpshift <value/mm>

Shift the scalp 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).

As a result of running the `mne_setup_foward_model` script, 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>-<ntri>.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. `<subject>-<surface name>-<ntri>.surf` files are created for each of the surfaces present in the BEM model. These can be loaded to *tkmedit* to check the location of the surfaces.
 4. 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 subsequently 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/EEG analysis directory

The remaining steps require that the actual MEG/EEG data are available. It is recommended that a new directory is created for the MEG/EEG data processing. The raw data files collected should not be copied there but rather referred to with symbolic links created with the `ln -s` command. 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. If *mne_browse_raw* or *mne_process_raw* is employed to calculate the offline averages and covariance matrices, this step is unnecessary since the data are filtered on the fly. For information on these programs, please consult Chapter 4.
6. For evoked-response analysis, the data has to be re-averaged off line, see Section 3.9.5.

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. It is advisable to lowpass filter and downsample the large raw data files often emerging in cognitive and patient studies to speed up subsequent processing. This can be accomplished with the *mne_process_raw* and *mne_browse_raw* software modules. For details, see Sections 4.2.3 and 4.4.3.

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

3.9.5 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.

3.10 Aligning the coordinate frames

The calculation of the forward solution requires knowledge of the relative location and orientation of the MEG/EEG and MRI coordinate systems. The MEG/EEG head coordinate system is defined in Section 5.3. The conversion tools included in the MNE software take care of the idiosyncrasies 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 5.3, 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/EEG 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. 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 most recent BEM file 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-neuro-mag/sets`. The search order for MEG/MRI coordinate transformations is discussed below.

--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. The search order for MEG/MRI coordinate transformations is discussed below.

--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.

--destdir <directory>

Optionally specifies a directory where the forward solution will be stored.

--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.

--eegonly

Omit MEG forward calculations.

--all

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

--overwrite

Overwrite the possibly existing forward model file.

--help

Show usage information for the script.

The MEG/MRI transformation is determined by the following search sequence:

1. If the **--mri** option was present, the file is looked for literally as specified, in the directory of the measurement file specified with the **--meas** option, and in the directory `$SUBJECTS_DIR/$SUBJECT/mri/T1-neuromag/sets`. If the file is not found, the script exits with an error message.
2. If the **--trans** option was present, the file is looked up literally as specified. If the file is not found, the script exists with an error message.
3. If neither **--mri** nor **--trans** option was not present, the following default search sequence is engaged:
 - a. The `.fif` ending in the measurement file name is replaced by `-trans.fif`. If this file is present, it will be used.
 - b. The newest file whose name ends with `-trans.fif` in the directory of the measurement file is looked up. If such a file is present, it will be used.
 - c. The newest file whose name starts with `COR-` in directory `$SUBJECTS_DIR/$SUBJECT/mri/T1-neuromag/sets` is looked up. If such a file is present, it will be used.
 - d. If all the above searches fail, the script exits with an error message.

This search sequence is designed to work well with the MEG/MRI transformation files output by *mne_analyze*, see Section 7.16. It is recommended that *-trans.fif* file saved with the *Save default* and *Save...* options in the *mne_analyze* alignment dialog are used because then the `$SUBJECTS_DIR/$SUBJECT` directory will be composed of files which are dependent on the subjects's anatomy only, not on the MEG/EEG data to be analyzed.

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 an 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 epi-

leptic 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.

--eeg

Employ EEG data in the inverse calculation. If neither **--meg** nor **--eeg** is set only MEG channels 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 0.1...0.6.

--depth

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

--bad <name>

Specifies 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.

--noisecov <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. For backward compatibility, **-senscov** can be used as a synonym for **--noisecov**.

--noiserank <value>

Specifies the rank of the noise covariance matrix explicitly rather than trying to reduce it automatically. This option is seldom needed,

--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, an *a priori* weighting can be applied to the source covariance matrix. The source of the weighting is usually fMRI but may be also some other data, provided that the weighting 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*, *tksurf* 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 weight files, respectively. The application of the weighting 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 is 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.

--destdir <directory>

Optionally specifies a directory where the inverse operator will be stored.

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, the MNE software includes bad-channel designation files which can be used 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/share/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 the 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. Using the MNE Matlab toolbox, it is possible to perform many of the above operations in Matlab using your own Matlab code based on the MNE Matlab toolbox. For more information on the MNE Matlab toolbox, see Chapter 10.
7. 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 raw data processor *mne_browse_raw* is designed for simple raw data viewing and processing operations. In addition, the program is capable of off-line averaging and estimation of covariance matrices. *mne_browse_raw* can be also used to view averaged data in the topographical layout. Finally, *mne_browse_raw* can communicate with *mne_analyze* described in Chapter 7 to calculate current estimates from raw data interactively.

mne_browse_raw has also an alias, *mne_process_raw*. If *mne_process_raw* is invoked, no user interface appears. Instead, command line options are used to specify the filtering parameters as well as averaging and covariance-matrix estimation command files for batch processing. This chapter discusses both *mne_browse_raw* and *mne_process_raw*.

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 interactive (*mne_browse_raw*) and batch (*mne_process_raw*) modes are listed.

4.2.1 Common options

--version

Show the program version and compilation date.

--help

List the command-line options.

--cd <dir>

Change to this directory before starting.

--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 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). The same compensation will be applied to all data files loaded by *mne_process_raw*. For *mne_browse_raw*, this applies only to the data file loaded by specifying the **--raw** option. For interactive data loading, the software gradient compensation is specified in the corresponding file selection dialog, see Section 4.4.1.

--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, the data will not be highpass filtered. It is best to 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 apart from that used during the acquisition will be 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 composite digital trigger channel. The default value is 'STI 014'. Underscores in the channel name will be replaced by spaces.

--digtrigmask <number>

Mask to be applied to the trigger channel values before considering them. This option is useful if one wants to set some bits in a don't care state. For example, some finger response pads keep the trigger lines high if not in use, *i.e.*, a finger is not in place. Yet, it is convenient to keep these devices permanently connected to the acquisition system. The number can be given in decimal or hexadecimal format (beginning with 0x or 0X). For example, the value 255 (0xFF) means that only the lowest order byte (usually trigger lines 1 - 8 or bits 0 - 7) will be considered.

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

Note: Strictly speaking, trigger mask value zero would mean that all trigger inputs are ignored. However, for convenience, setting the mask to zero or not setting it at all has the same effect as 0xFFFFFFFF, *i.e.*, all bits set.

Tip: The digital trigger channel can also be set with the MNE_TRIGGER_CH_NAME environment variable. Underscores in the variable value will *not* be replaced with spaces by *mne_browse_raw* or *mne_process_raw*. Using the **--digtrig** option supersedes the MNE_TRIGGER_CH_NAME environment variable.

Tip: The digital trigger channel mask can also be set with the MNE_TRIGGER_CH_MASK environment variable. Using the **--digtrigmask** option supersedes the MNE_TRIGGER_CH_MASK environment variable.

4.2.2 Interactive mode options

These options apply to the interactive (*mne_browse_raw*) version only.

--allowmaxshield

Allow loading of unprocessed Elekta-Neuromag data with Max-Shield on. These kind of data should never be used for source localization without further processing with Elekta-Neuromag software.

--deriv <name>

Specifies the name of a derivation file. This overrides the use of a standard derivation file, see Section 4.4.12.

--sel <name>

Specifies the channel selection file to be used. This overrides the use of the standard channel selection files, see Section 4.5.5.

4.2.3 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.

--makeproj

Estimate the noise subspace from the data and create a new signal-space projection operator instead of using one attached to the data file or those specified with the **--proj** option. The following eight options define the parameters of the noise subspace estimation. More information on the signal-space projection can be found in Section 4.16.

--projevent <no>

Specifies the events which identify the time points of interest for projector calculation. When this option is present, **--projtmin** and **--projtmax** are relative to the time point of the event rather than the whole raw data file.

--projtmin <time/s>

Specify the beginning time for the calculation of the covariance matrix which serves as the basis for the new SSP operator. This option is required with `--projevent` and defaults to the beginning of the raw data file otherwise. This option is effective only if `--makeproj` or `--saveprojtag` options are present.

--projtmax <time/s>

Specify the ending time for the calculation of the covariance matrix which serves as the basis for the new SSP operator. This option is required with `--projevent` and defaults to the end of the raw data file otherwise. This option is effective only if `--makeproj` or `--saveprojtag` options are present.

--projngrad <number>

Number of SSP components to include for planar gradiometers (default = 5). This value is system dependent. For example, in a well-shielded quiet environment, no planar gradiometer projections are usually needed.

--projnmag <number>

Number of SSP components to include for magnetometers / axial gradiometers (default = 8). This value is system dependent. For example, in a well-shielded quiet environment, 3 – 4 components are needed while in a noisy environment with light shielding even more than 8 components may be necessary.

--projgradrej <value/ fT/cm>

Rejection limit for planar gradiometers in the estimation of the covariance matrix from which the new SSP operator is derived. The default value is 2000 fT/cm. Again, this value is system dependent.

--projmagrej <value/ fT>

Rejection limit for planar gradiometers in the estimation of the covariance matrix from which the new SSP operator is derived. The default value is 3000 fT. Again, this value is system dependent.

--saveprojtag <tag>

This option defines the names of files to hold the SSP operator. If this option is present the `--makeproj` option is implied. The SSP operator file name is formed by removing the trailing `.fif` or `_raw.fif` from the raw data file name by appending `<tag>.fif` to this stem. Recommended value for `<tag>` is `-proj`.

--saveprojaug

Specify this option if you want to use the projection operator file output in the Elekta-Neuromag *Signal processor (graph)* software.

--eventsout <name>

List the digital trigger channel events to the specified file. By default, only transitions from zero to a non-zero value are listed. If multiple raw data files are specified, an equal number of **--event-sout** options should be present. If the file name ends with **.fif**, the output will be in fif format, otherwise a text event file will be output.

--allevents

List all transitions to file specified with the **--eventsout** option.

--events <name>

Specifies the name of a fif or text format event file (see Section 4.10.5) to be associated with a raw data file to be processed. If multiple raw data files are specified, the number of **--events** options can be smaller or equal to the number of raw data files. If it is equal, the event filenames will be associated with the raw data files in the order given. If it is smaller, the remaining raw data files for which an event file is not specified will *not* have an event file associated with them. The event file format is recognized from the file name: if it ends with **.fif**, the file is assumed to be in fif format, otherwise a text file is expected.

--ave <name>

Specifies the name of an off-line averaging description file. For details of the format of this file, please consult Section 4.13. 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 <tag>

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 **<tag>.fif** or **<tag>.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 estimation. For details of the format of this file, please see Section 4.14. 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 <tag>

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

--savehere

If the `--saveavetag` and `--savecovtag` options are used to generate the file output file names, the resulting files will go to the same directory as raw data by default. With this option the output files will be generated in the current working directory instead.

--gcov <name>

If multiple raw data files are specified as input and covariance matrix estimation is requested, the grand average over all data files will be saved to `<name>`. The details of the covariance matrix estimation are given in Section 4.17.

--save <name>

Save a filtered and optionally down-sampled version of the data file to `<name>`. If multiple raw data files are specified, an equal number of `--save` options should be present. If `<filename>` ends with `.fif` or `_raw.fif`, these endings are deleted. After these modifications, `_raw.fif` is inserted after the remaining part of the file name. If the file is split into multiple parts (see `--split` option below), the additional parts will be called `<name>--<number>_raw.fif`

--split <size/MB>

Specifies the maximum size of the raw data files saved with the `--save` option. By default, the output is split into files which are just below 2 GB so that the fif file maximum size is not exceed.

--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

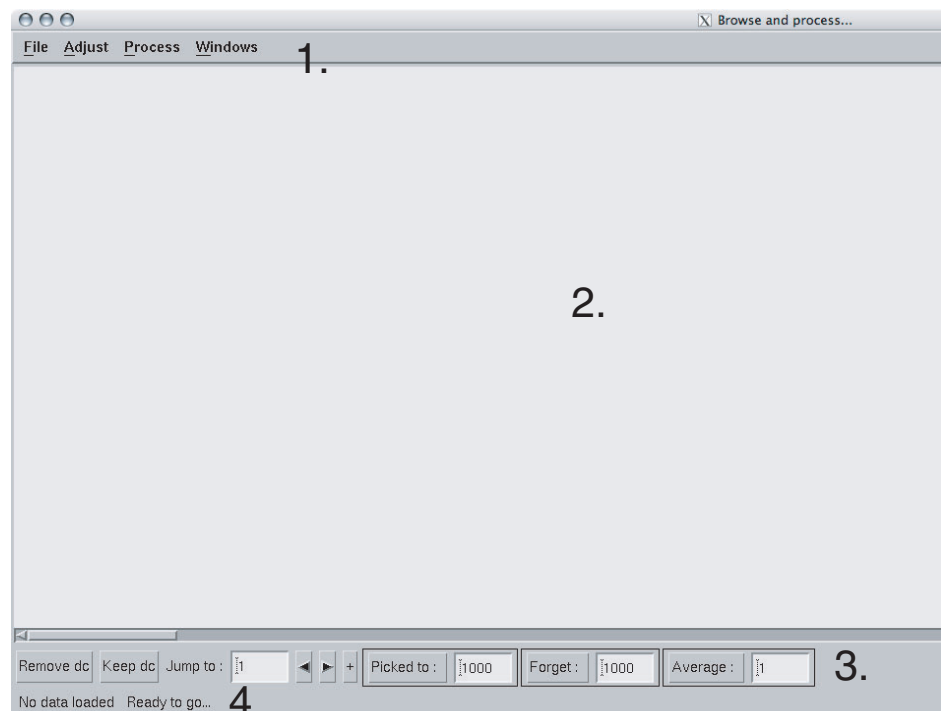


Figure 4.1 The user interface of *mne_browse_raw*

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 The *File* menu

4.4.1 Open

Selecting *Open* from file menu pops up the dialog shown in Figure 4.2.

The *Raw files* and *Maxfilter output* buttons change the file name filter to include names which end with `_raw.fif` or `sss.fif`, respectively, to facilitate selection of original raw files or those processed with the Neuro-mag *Maxfilter*TM software

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.

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.

When a raw data file is opened, the digital trigger channel 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 extension>-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>`. The fif format event files can be read and written with the *mne_read_events* and *mne_write_events* functions in the MNE Matlab toolbox, see Chapter 10.

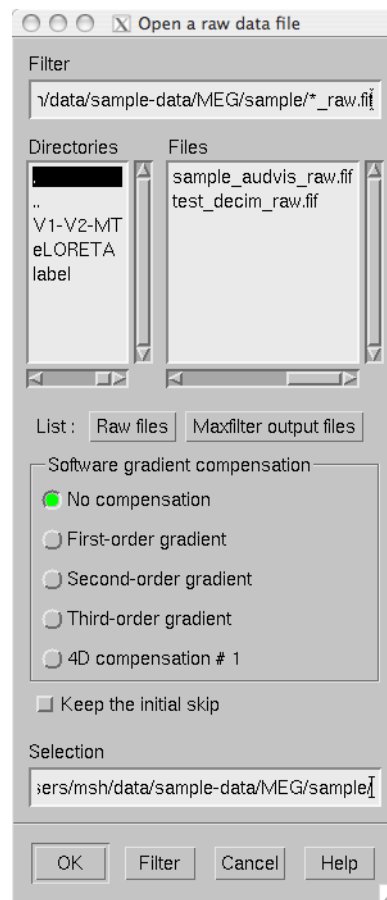


Figure 4.2 The *Open* dialog.

4.4.2 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 average view window, see Section 4.12. The data loaded are affected by the scale settings, see, Section 4.5.2, the filter, see Section 4.5.1, and the options selected in the *Manage averages* dialog, see Section 4.15.

4.4.3 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 down-sampling factor. The sampling frequency after down-sampling must be at least three times the low-pass filter corner frequency. The output will be split into files which are just below 2 GB so that the fif file maximum size is not exceed.

If `<filename>` ends with `.fif` or `_raw.fif`, these endings are deleted. After these modifications, `_raw.fif` is inserted after the remaining part of the file name. If the file is split into multiple parts, the additional parts will be called `<name>--<number>_raw.fif`. For downsampling and saving options in `mne_process_raw`, see Section 4.2.3.

4.4.4 Change working directory

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

4.4.5 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.

4.4.6 Save projection

The *Save projection...* item in the *File* menu pops up a dialog to save the present projection operator into a file. Normally, the EEG average reference projection is not included. If you want to include it, mark the *Include EEG average reference* option. If your MEG projection includes items for both magnetometers and gradiometers and you want to use the projection operator file output from here in the Neuromag *Signal processor (graph)* software, mark the *Enforce compatibility with graph* option.

4.4.7 Apply bad channels

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

4.4.8 Load events (text)

Reads a text format event file. For more information on events, see Section 4.10.

4.4.9 Load events (fif)

Reads a fif format event file. For more information on events, see Section 4.10.

4.4.10 Save events (text)

Brings up 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 estimation as an input file to specify the time points of events, see Section 4.10.5. For more information on events, see Section 4.10.

4.4.11 Save events (fif)

Save the events in fif format. These binary event files can be read and written with the *mne_read_events* and *mne_write_events* functions in the MNE Matlab toolbox, see Chapter 10. For more information on events, see Section 4.10.

4.4.12 Load derivations

This menu choice allows loading of channel derivation data files created with the *mne_make_derivations* utility, see Section 11.5, or using the interactive derivations editor in *mne_browse_raw*, see Section 4.5.4. 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 weights, 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 4.5.5. At present, derived channels cannot be displayed in topographical data displays. Derived channels are not included in averages or noise covariance matrix estimation.

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 unless the `--deriv` option has been used to specify a nonstandard derivation file, see Section 4.2.2.

4.4.13 Save derivations

Saves the current derivations into a file.

4.4.14 Load channel selections

This choice loads a new set of channel selections. The default directory for the selections is \$HOME/.mne. If this directory does not exist, it will be created before bringing up the file selection dialog to load the selections.

4.4.15 Save channel selections

This 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 4.5.5. The default directory for the selections is \$HOME/.mne. If this directory does not exist, it will be created before bringing up the file selection dialog to save the selections. Note that all currently existing selections will be saved, not just those added to the ones initially loaded.

4.4.16 Quit

Exits the program without questions asked.

4.5 The *Adjust* menu

4.5.1 Filter

Selecting *Filter...* from the *Adjust* menu pops up the dialog shown in Figure 4.3.

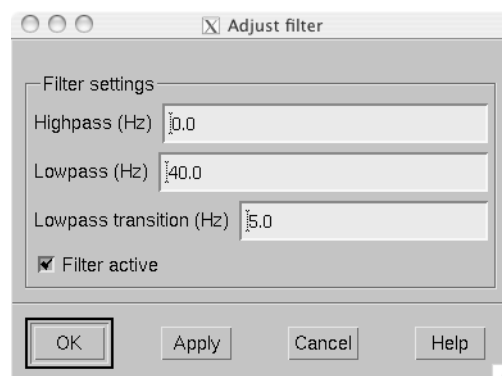


Figure 4.3 The filter adjustment dialog.

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. When a filter is in effect, the value of the first sample in the file is subtracted from the data to correct for an initial dc offset. This procedure also eliminates any filter artifacts in the beginning of the data.

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.

4.5.2 Scales

Selecting *Scales...* from the *Adjust* menu pops up the dialog shown in Figure 4.4.

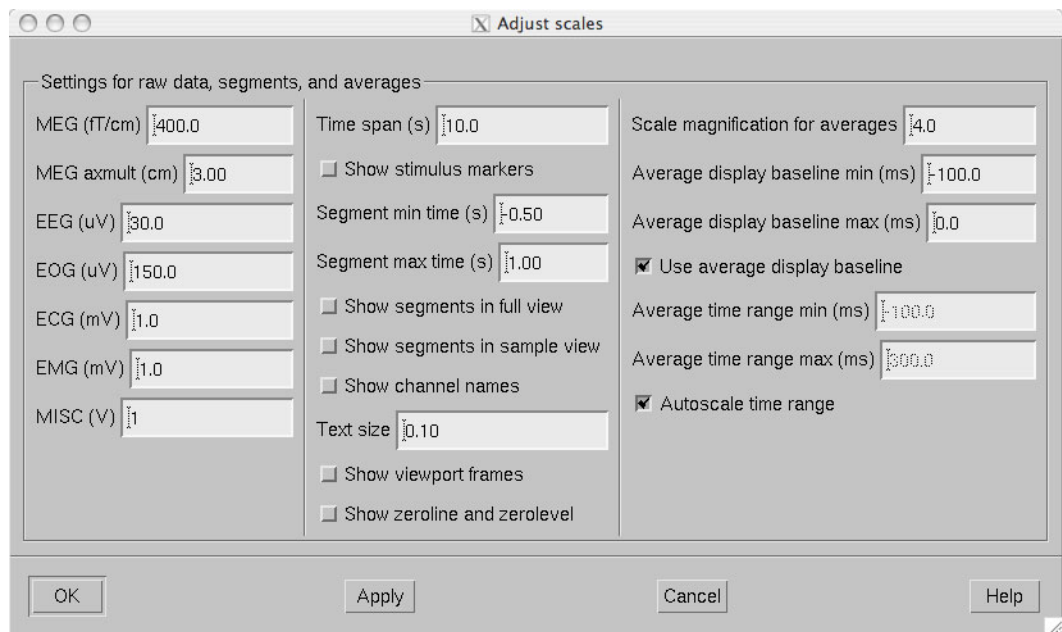


Figure 4.4 The *Scales* dialog.

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 stimulus 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.

Show channel names

Show the names of the channels in the topographical displays.

Text size

Size of the channel number text as a fraction of the height of each viewport.

Show viewport frames

Show the boundaries of the viewports in the topographical displays.

Show zeroline and zerolevel

Show the zero level, *i.e.*, the baseline level in the topographical displays. In addition, the zero time point is indicated in the average views if it falls to the time range, *i.e.*, if the minimum of the time scale is negative and the maximum is positive.

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.

4.5.3 Colors

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

4.5.4 Derivations

Brings up the interactive derivations editor. This editor can be used to add or modify derived channels, *i.e.*, linear combinations of signals actually recorded. Channel derivations can be also created and modified using the *mne_make_derivations* tool, see Section 11.5. The interactive editor contains two main areas:

1. Interactive tools for specifying a channel linear combination. This tool is limited to combining up to five channels in each of the derivations. Clicking *Add* after defining the name of the new derivation, the weights of the component channels and their names, adds the corresponding arithmetic expression to the text area.
2. Text area which contains the currently defined derivations as arithmetic expressions in a format identical to that used by *mne_make_derivations*. These expressions can be manually edited before accepting the new set of derivations. Initially, the text area will contain the derivations already defined.

The *Define* button interprets the arithmetic expressions in the text area as new derivations and closes the dialog. The *Cancel* button closes the dialog without any change in the derivations.

Recommended workflow for defining derived EEG channels and associated selections interactively involves the following steps:

1. If desired, EEG channels can be relabeled with descriptive names using the *mne_rename_channels* utility, see Section 11.4.5. It is strongly recommended that you keep a copy of the channel alias file used by *mne_rename_channels*. If necessary, you can then easily return to the original channel names by running *mne_rename_channels* again with the `--revert` option.
2. Load the data file into *mne_browse_raw* and use the interactive derivations editor to create the desired derived channels. These are usually differences between the signals in two EEG electrodes.
3. Save the derivations from the file menu.
4. If desired, move the derivations file to the standard location (`$HOME/.mne/mne_browse_raw-deriv.fif`).
5. Create new channel selections employing the original and derived channels using the channel selection tool described in Section 4.5.5.
6. Save the new channel selections from the file menu.
7. If desired, change the order of the channels in the selections in the selection file by editing it in a text editor and move it to the standard location `$HOME/.mne/mne_browse_raw.sel`.

4.5.5 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 `$MNE_ROOT/share/mne/mne_browse_raw/mne_browse_raw.sel`. This default channel selection file can be modified by copying the file into `$HOME/.mne/mne_browse_raw.sel`. The format of this text file should be self explanatory.

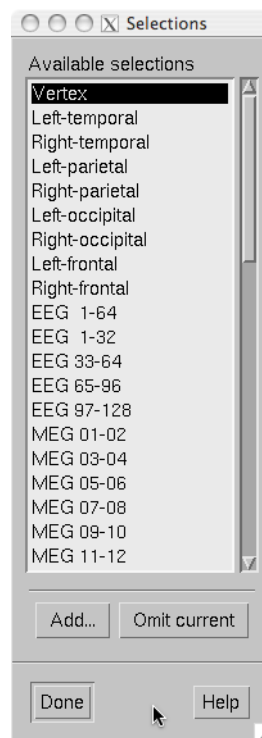


Figure 4.5 The channel selection dialog.

The channel selection dialog is shown in Figure 4.5. The number of items in the selection list depends on the contents of your selection file. If the list has the keyboard focus you can easily move from one selection to another with the up and down arrow keys.

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

Add...

Brings up the selection dialog shown in Figure 4.6 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 4.4.15.



Figure 4.6 Dialog to create a new channel selection.

The components of the selection creation dialog shown in Figure 4.6 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 4.4.12.

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 deselects 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 *Regex*:

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

Note: The interactive tool for creating the channel selections does not allow you to change the order of the selected channels from that given by the list of channels. However, the ordering can be easily changed by manually editing the channel selection file in a text editor.

4.5.6 Full view layout

Shows a selection of available layouts for the topographical views (full view and average display). The system-wide layout files reside in `$MNE_ROOT/share/mne/mne_analyze/lout`. In addition any layout files residing in `$HOME/.mne/lout` are listed. The default layout is Vectorview-grad. If there is a layout file in the user's private layout directory ending with `-default.lout`, that layout will be used as the default instead. The *Default* button returns to the default layout.

The format of the layout files is:

```

<plot area limits>
<viewport definition #1>
...
<viewport definition #N>

```

The *<plot area limits>* define the size of the plot area (x_{\min} x_{\max} y_{\min} y_{\max}) which should accommodate all view ports. When the layout is used, the plot area will preserve its aspect ratio; if the plot window has a different aspect ratio, there will be empty space on the sides.

The viewports define the locations of the individual channels in the plot. Each viewport definition consists of

```

<number>  $x_0$   $y_0$  <width> <height> <name>[:<name>]...

```

where number is a viewport number (not used by the MNE software), x_0 and y_0 are the coordinates of the lower-left corner of the viewport, <width> and <height> are the viewport dimensions, and <name> is a name of a channel. Multiple channel names can be specified by separating them with a colon.

When a measurement channel name is matched to a layout channel name, all spaces are removed from the channel names and the both the layout channel name and the data channel name are converted to lower case. In addition anything including and after a hyphen (-) is omitted. The latter convention facilitates using CTF MEG system data, which has the serial number of the system appended to the channel name with a dash. Removal of the spaces is important for the Neuromag Vectorview data because newer systems do not have spaces in the channel names like the original Vectorview systems did.

Tip: The *mne_make_eeg_layout* utility can be employed to create a layout file matching the positioning of EEG electrodes, see Section 11.6.

4.5.7 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 4.16.

4.5.8 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, above.

4.5.9 Averaging preferences

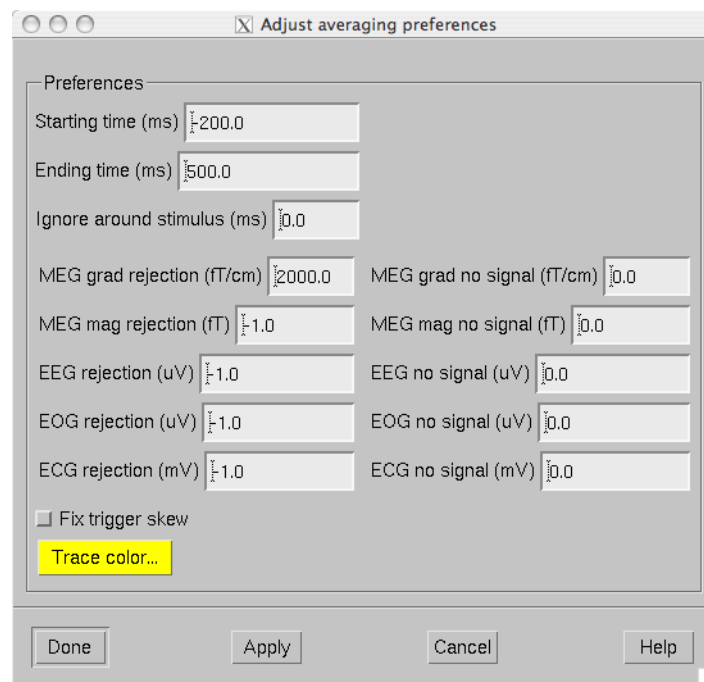


Figure 4.7 Averaging preferences.

Selecting *Averaging preferences...* from the *Adjust* menu pops up the dialog shown in Figure 4.7. These settings apply only to the simple averages calculated with help of tools residing just below the main raw data display, see Section 4.11. These settings are also applied when a covariance matrix is computed to create a SSP operator as described in Section 4.6.4 and in the computation of a covariance matrix from raw data, see Section 4.6.3.

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.

Ignore around stimulus (ms)

Ignore this many milliseconds on both sides of the trigger when considering the epoch. This parameter is useful for ignoring large stimulus artefacts, *e.g.*, from electrical somatosensory stimulation.

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 and axial gradiometers.

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.

MEG grad no signal (fT/cm)

Signal detection criterion for MEG planar gradiometers. The peak-to-peak value of all planar gradiometer signals must exceed this value, for the epoch to be included. This criterion allows rejection of data with saturated or otherwise dysfunctional channels.

MEG mag no signal (fT)

Signal detection criterion for MEG magnetometers and axial gradiometers.

EEG no signal (μ V)

Signal detection criterion for EEG channels.

EOG no signal (μ V)

Signal detection criterion for EOG channels.

ECG no signal (mV)

Signal detection criterion for ECG channels.

Fix trigger skew

This option has the same effect as the *FixSkew* parameter in averaging description files, see Section 4.13.2.

Trace color

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

4.6 The *Process* menu

4.6.1 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 4.13. All parameters for the averaging are taken from the description file, *i.e.*, the parameters set in the averaging preferences dialog (Section 4.5.9) do not effect the result.

4.6.2 Estimation 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 estimation of a covariance matrix. The structure of these files is described in Section 4.14.

4.6.3 Estimation 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 estimation 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 4.5.9. The time range can be also selected interactively from the main raw data display by doing a range selection with shift left button drag.

4.6.4 Creating a new SSP operator

The *Create a new SSP operator...* menu choice computes a new SSP operator as discussed in Section 4.16.2.

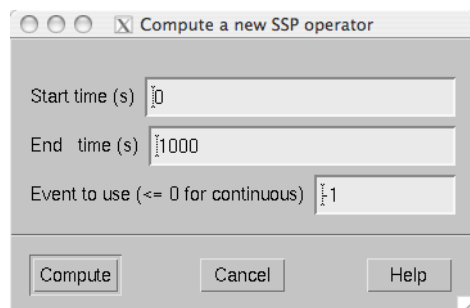


Figure 4.8 Time range specification for SSP operator calculation

When *Create a new SSP operator...* selected, a window shown in Figure 4.8 is popped up. It allows the specification of a time range to be employed in the calculation of a raw data covariance matrix. The time range can be also selected interactively from the main raw data display by doing a range selection with shift left button drag. 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. The artifact rejection parameters specified averaging preferences will be applied in the covariance matrix calculation, see Section 4.5.9.

Instead of using continuous raw data, it is also possible to employ short epochs around triggers (events) in the calculation of the new SSP operator by specifying a positive event number in the time specification dialog.

This option is very useful, *e.g.*, to remove MCG/ECG artifacts from the data to facilitate detection of epileptic spikes:

1. Select left or right temporal channels to the display.
2. Mark several peaks of the MCG signal in the data: click on the first one and control click on the subsequent ones to extend the selection.
3. Select an event number next to the *Picked to* button in the tool bar, see Section 4.11, and click *Picked to*. As a result the lines marking the events will change color (by default from green to blue) indicating transition to user-created events.
4. Specify an epoch time range to be employed and the event number selected in the previous step for the SSP operator calculation.

Once the parameters are set, click *Compute* to calculate a covariance matrix according to your specifications. Once the covariance matrix is ready, the parts corresponding to magnetometer or axial gradiometer, planar gradiometer, and EEG channels are separated and the corresponding eigenvectors and eigenvalues are computed. Once complete, a projection selector with eight magnetometer eigenvectors, five planar gradiometer eigenvectors, three EEG eigenvectors, as well as the existing projection items 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. More information on the SSP method can be found in Section 4.16.

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: The command-line options for *mne_process_raw* allow calculation of the SSP operator from continuous data in the batch mode, see Section 4.2.3.

4.7 The *Windows* menu

The *Windows* menu contains the following items:

Show full view...

Brings up the topographical display of epochs extracted from the raw data, see Section 4.12.

Show averages...

Brings up the topographical display showing averaged data. These data may include data averaged in the current *mne_browse_raw* session or those loaded from files, see Section 4.4.2.

Show event list...

Brings up a window containing a list of the currently defined events. Clicking on an event in the list, the event is selected, a green cursor appears at the event, and the event is brought to the middle of the raw data display. The event list displayed can be also restricted to user-defined events (annotations) and user-defined events can be deleted. For further information, see Section 4.10.

Show annotator...

Brings up a window which allows adding new events to the data with annotations or comments. For details, see Section 4.10.

Manage averages...

Brings up a dialog to control the averaged data sets, see Section 4.15.

Start mne_analyze...

Start interaction between *mne_browse_raw* and *mne_analyze*. For details, see Section 4.18.

Show head position

Starts *mne_analyze* in the head position visualization mode and shows the relative position of the MEG sensor array and the head using the data in the presently open raw data file. For more details on the head position visualization mode, see Section 7.21.

Quit mne_analyze...

Quits the *mne_analyze* program started with *Start mne_analyze...*

4.8 The *Help* menu

The contents of the *Help* menu is shown in Figure 4.9:

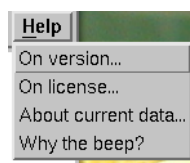


Figure 4.9 The *Help* menu.

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.

4.9 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 4.5.5. The length of the data section displayed is controlled from the scales dialog (Section 4.5.2) and the filtering from the filter dialog (Section 4.5.1). A signal-space projection can be applied to the data by loading a projection operator (Section 4.4.5). The selection of the projection operator items is controlled from the projection dialog described in Section 4.5.7.

The control and browsing functions of the main data display are:

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.

Browsing

Browsing through the data. The section of data displayed can be selected from the scroll bar at the bottom of the display. Additional browsing functionality will be discussed n In addition, if the strip-

chart display has the keyboard focus, you can scroll back and forth with the page up and page down keys.

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 4.5.2), a display of an epoch of data specified in the scales dialog will appear. For more information on full view, see Section 4.12. 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 tool bar offers functions to operate on the selected time points, see Section 4.11.

Range selection

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.

Saving a copy of the display

The right mouse button invokes a popup menu which allows saving of the display in various formats. Best quality is achieved with the Illustrator format. This format has the benefit that it is object oriented and can be edited in Adobe Illustrator.

Drag and drop

Graphics can be moved to one of the Elekta-Neuromag report composer (*cliplab*) view areas with the middle mouse button.

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

Note: The *cliplab* drag-and-drop functionality requires that you have the proprietary Elekta-Neuromag analysis software installed. *mne_browse_raw* is compatible with *cliplab* versions 1.2.13 and later.

4.9.1 Browsing data

If the strip-chart display has the input focus (click on it, if you are unsure) the keyboard and mouse can be used to browse the data as follows:

Up and down arrow keys

Activate the previous or next selection in the selection list.

Left and right arrow keys

If a single time point is selected (green line), move the time point forward and backward by ± 1 ms. If the shift key is down, the time point is moved by ± 10 ms. If the control key is down (with or without shift), the time point is moved by ± 100 ms. If *mne_browse_raw* is controlling *mne_analyze* (see Section 4.18), the *mne_analyze* displays will be updated accordingly. If the picked time point falls outside the currently displayed section of data, the display will be automatically scrolled backwards or forwards as needed.

Rotate the mouse wheel or rotate the trackball up/down

Activate the previous or next selection in the selection list.

Rotate the trackball left/right or rotate the wheel with shift down

Scroll backward or forward in the data by one screen. With *Alt* key (*Command* or *Apple* key in the Mac keyboard), the amount of scrolling will be 1 s instead of the length of one screen. If shift key is held down with the trackball, both left/right and up/down movements scroll the data in time.

Note: The trackball and mouse wheel functionality is dependent on your X server settings. On Mac OSX these settings are normally correct by default but on a LINUX system some adjustments to the X server settings maybe necessary. Consult your system administrator or Google for details.

4.10 Events and annotations

4.10.1 Overview

In *mne_browse_raw* and *mne_process_raw* *events* mark interesting time points in the data. When a raw data file is opened, a standard event file is consulted for the list of events. If this file is not present, the digital trigger channel, defined by the *--digtrig* option or the *MNE_TRIGGER_CH_NAME* environment variable is scanned for events. For more information, see Sections 4.2.1 and 4.4.1.

In addition to the events detected on the trigger channel, it is possible to associate user-defined events to the data, either by marking data points interactively as described in Section 4.10.4 or by loading event data from files, see Section 4.10.3. Especially if there is a comment associated with a user-defined event, we will sometimes call it an *annotation*.

If a data files has annotations (user-defined events) associated with it in *mne_browse_raw*, information about them is automatically saved to an *annotation file* when a data file is closed, *i.e.*, when you quit *mne_browse_raw* or load a new data file. This annotation file is called *<raw data file name without fif extension>-annot.fif* and will be

stored in the same directory as the raw data file. Therefore, write permission to this directory is required to save the annotation file.

Both the events defined by the trigger channel and the user-defined events have three properties:

1. The *time* when the event occurred.
2. The *value* on the trigger channel just before the change and now. For user-defined events the value before is always zero and the current value is user defined and does not necessarily reflect a change on the trigger channel. The trigger channel events may also indicate changes between two non-zero values and from a non-zero to zero. The event list described in Section 4.10.2 shows only transitions from zero to a non-zero value. Similarly, the *Jump to* item in the tool bar, described in Section 4.11, only detects transitions from zero to a nonzero value.
3. An optional *comment* text, which is especially helpful in associating user-defined events with real-world activity, *e.g.*, the subject closing or opening his/her eyes or an epileptic patient showing indications of a seizure.

4.10.2 The event list

The *Windows/Show event list...* menu choice shows a window containing a list of currently defined events. The list can be restricted to user-defined events by checking *User-defined events only*. When an event is selected from the list, the main display jumps to the corresponding time. If a user-defined event is selected, it can be deleted with the *Delete a user-defined event* button.

4.10.3 Loading and saving event files

Using the Load/Save events choices in the file menu, events can be saved in text and fif formats, see Section 4.10.5, below. The loading dialogs have the following options:

Match comment with

Only those events which will contain comments and in which the comment matches the entered text are loaded. This filtering option is useful, *e.g.*, in loading averaging or covariance matrix computation log files, see Sections 4.13.2 and page 81. If the word *omit* is entered as the filter, only events corresponding to discarded epochs are loaded and the reason for rejection can be investigated in detail.

Add as user events

Add the events as if they were user-defined events. As a result, the annotation file saved next time *mne_browse_raw* closes this raw file will contain these events.

Keep existing events

By default, the events loaded will replace the currently defined ones. With this option checked, the loaded event will be merged with the currently existing ones.

The event saving dialogs have the following options controlling the data saved:

Save events read from the data file

Save only those event which are not designated as user defined. These are typically the events corresponding to changes in the digital trigger channel. Another possible source for these events is an event file manually loaded *without* the *Add as user events* option.

Save events created here

Save the user-defined events.

Save all trigger line transitions

By default only those events which are associate with a transition from zero to non-zero value are saved. These include the user-defined events and leading edges of pulses on the trigger line. When this option is present, all events included with the two above options are saved, regardless the type of transition indicated (zero to non-zero, non-zero to another non-zero value, and non-zero value to zero).

Tip: If you have a text format event file whose content you want to include as user-defined events and create the automatic annotation file described in Section 4.10.1, proceed as follows:

1. Load the event file with the option *Add as user events* set.
2. Open another data file or quit *mne_browse_raw*.
3. Optionally remove unnecessary events using the event list dialog.

The directory in which the raw data file resides now contains an annotation file which will be automatically loaded each time the data file is opened. A text format event file suitable for this purpose can be created manually, extracted from an EDF+ file using the `--tal` option in *mne_edf2fiff* discussed in Section 9.2.8, or produced by custom software used during data acquisition.

4.10.4 Defining annotated events

The *Windows/Show annotator...* shows a window to add annotated user-defined events. In this window, the buttons in first column mark one or more selected time points with the event number shown in the second column with an associated comment specified in the third column. Marking also occurs when return is pressed on any of the second and third column text fields.

When the dialog is brought up for the first time, the file `$HOME/.mne/mne_browse_raw.annot` is consulted for the definitions of the second and third column values, *i.e.*, event numbers and comments. You can save the current definitions with the *Save defs* button and reload the annotation definition file with *Load defs*. The annotation definition file may contain comment lines starting with ‘%’ or ‘#’ and data lines which contain an event number and an optional comment, separated from the event number by a colon.

Tip: If you want to add a user-defined event without an a comment, you can use the *Picked to* item in the tool bar, described in Section 4.11.

4.10.5 Event files

A text format 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> <text>
```

where

- <sample>** is the sample number. This sample number takes into account the initial empty space in a raw data file as indicated by the `FIFF_FIRST_SAMPLE` and/or `FIFF_DATA_SKIP` tags in the beginning of raw data. Therefore, the event file contents are independent of the *Keep initial skip* setting in the open dialog.
- <time>** is the time from the beginning of the file to this sample in seconds.
- <from>** is the value of the digital trigger channel at `<sample>-1`.
- <to>** is the value of the digital trigger channel at `<sample>`.
- <text>** is an optional annotation associated with the event. This comment will be displayed in the event list and on the message line when you move to an event.

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>`.

Each event file starts with a “pseudo event” where both `<from>` and `<to>` fields are equal to zero.

Warning: In previous versions of the MNE software, the event files did not contain the initial empty pseudo event. In addition the sample numbers did not take into account the initial empty space in the raw data files. The present version of MNE software is still backwards compatible with the old version of the event files and interprets the sample numbers appropriately. However, the recognition of the old and new event file formats

depends on the initial pseudo event and, therefore, this first event should never be removed from the new event files. Likewise, if an initial pseudo event with *<from>* and *<to>* fields equal to zero is added to an old event file, the results will be unpredictable.

Note: If you have created Matlab, Excel or other scripts to process the event files, they may need revision to include the initial pseudo event in order for *mne_browse_raw* and *mne_process_raw* to recognize the edited event files correctly.

Note: Events can be also stored in fif format. This format can be read and written with the Matlab toolbox functions *mne_read_events* and *mne_write_events*.

4.11 The tool bar

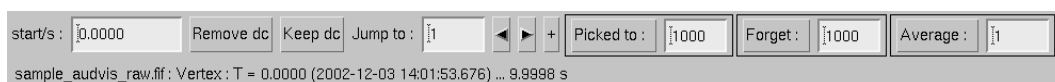


Figure 4.10 The tool bar controls.

The tool bar controls are shown in Figure 4.10. They perform the following functions:

start/s

Allows specification of the starting time of the display as a numeric value. Note that this value will be rounded to the time of the nearest sample when you press return. If you click on this text field, you can also change the time with the up and down cursor keys (1/10 of the window size), and the page up and down (or control up and down cursor) keys (one window size).

Remove dc

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

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 4.5.2. If the '+' button is active, previous selections are kept, otherwise they are cleared.

Picked to

Make user events with this event number at all picked time points. It is also possible to add annotated user events with help of the annotation dialog. For further information, see Section 4.10.

Forget

Forget desired user events.

Average

Compute an average to this event.

The tool bar status line shows the starting time and the length of the window in seconds as well as the cursor time point. The dates and times in parenthesis show the corresponding wall-clock times in the time zone where *mne_browse_raw* is run.

Note: The wall-clock times shown are based on the information in the *fif* file and may be offset from the true acquisition time by about 1 second. This offset is constant throughout the file. The times reflect the time zone setting of the computer used to analyze the data rather than the one used to acquire them.

4.12 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 `$MNE_ROOT/share/mne/mne_analyze/layout`. In addition any layout files residing in `$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 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. Best quality is

achieved with the Illustrator format. This format has the benefit that it is object oriented and can be edited in Adobe Illustrator.

Middle button

Drag and drop graphics to one of the *cliplab* view areas.

Note: The *cliplab* drag-and-drop functionality requires that you have the proprietary Elekta-Neuromag analysis software installed. *mne_browse_raw* is compatible with *cliplab* versions 1.2.13 and later.

Note: The graphics output files will contain a text line stating of the time and vertical scales if the zero level/time and/or viewport frames have been switched on in the scales dialog, see Section 4.5.2.

4.13 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/share/mne/mne_browse_raw/templates`.

4.13.1 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.

4.13.2 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 recognized from the file name: if it ends with *.fif*, the file is assumed to be in fif format, otherwise a text file is expected. The text event file format is described in Section 4.10.5.

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 and axial gradiometer channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the magnetometer or axial gradiometer 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.

gradFlat <value / T/m>

Signal detection criterion for MEG planar gradiometers. The peak-to-peak value of all planar gradiometer signals must exceed this value, for the epoch to be included. This criterion allows rejection of data with saturated or otherwise dysfunctional channels. The default value is zero, *i.e.*, no rejection.

magFlat <value / T>

Signal detection criterion for MEG magnetometers and axial gradiometers channels.

eegFlat <value / V>

Signal detection criterion for EEG channels.

eogFlat <value / V>

Signal detection criterion for EOG channels.

ecgFlat <value / V>

Signal detection criterion for ECG channels.

stimIgnore <time / s>

Ignore this many seconds on both sides of the trigger when considering the epoch. This parameter is useful for ignoring large stimulus artefacts, *e.g.*, from electrical somatosensory stimulation.

fixSkew

Since the sampling of data and the stimulation devices are usually not synchronized, all trigger input bits may not turn on at the same sample. If this option is included in the off-line averaging description file, the following procedure is used to counteract this: if there is a transition from zero to a nonzero value on the digital trigger channel at sample n , the following sample will be checked for a transition from this nonzero value to another nonzero value. If such an event pair is found, the two events will be jointly considered as a transition from zero to the second non-zero value. With the *fixSkew* option, *mne_browse_raw/mne_process_raw* behaves like the Elekta-Neuromag on-line averaging and Maxfilter™ software.

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.

4.13.3 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 digital trigger channel. The interpretation of the values on the trigger channel can be further modified by the *ignore* and *mask* keywords. If multiple event parameters are present for a category, all specified events will be included in the average.

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* ($2^2 + 2^7 = 132$).

mask <number>

Works similarly to *ignore* except that a mask specifies the trigger channel bits to be included. For example, to look at trigger input lines one to three only, ignoring others, specify *mask 7* ($2^0 + 2^1 + 2^2 = 7$).

prevevent <number>

Specifies the event that is required to occur immediately before the event(s) specified with *event* parameter(s) in order for averaging to occur. Only one previous event number can be specified.

prevignore <number>

Works like *ignore* but for the events specified with *prevevent*. If *prevignore* and *prevmask* are missing, the mask implied by *ignore* and *mask* is applied to *prevevent* as well.

prevmask <number>

Works like *mask* but for the events specified with *prevevent*. If *prevignore* and *prevmask* are missing, the mask implied by *ignore* and *mask* is applied to *prevevent* as well.

nextevent <number>

Specifies the event that is required to occur immediately after the event(s) specified with *event* parameter(s) in order for averaging to occur. Only one next event number can be specified.

nextignore <number>

Works like *ignore* but for the events specified with *nextevent*. If *nextignore* and *nextmask* are missing, the mask implied by *ignore* and *mask* is applied to *nextevent* as well.

nextmask <*number*>

Works like *mask* but for the events specified with *nextevent*. If *nextignore* and *nextmask* are missing, the mask implied by *ignore* and *mask* is applied to *nextevent* as well.

delay <*time / s*>

Adds a delay to the time of the occurrence of an event. Therefore, if this parameter is positive, the zero time point of the epoch will be later than the time of the event and, correspondingly, if the parameter is negative, the zero time point of the epoch will be earlier than the event. By default, there will be no delay.

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.

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.

stderr

The standard error of mean will be computed for this category and included in the output fif file.

Note: Specification of the baseline limits does not any more imply the estimation of the standard error of mean. Instead, the *stderr* parameter is required to invoke this option.

4.14 Description files for covariance matrix estimation

Covariance matrix estimation 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/share/mne/mne_browse_raw/templates`.

4.14.1 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.

4.14.2 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. This file can be either in fif or text format (see Section 4.10.5). The event file format is recognized from the file name: if it ends with `.fif`, the file is assumed to be in fif format, otherwise a text file is expected. If this parameter is present, the trigger events in the raw data file are ignored and this event file is consulted instead. The event file format is described in Section 4.10.5.

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 and axial gradiometer channels. If the peak-to-peak amplitude within the extracted epoch exceeds this value on any of the magnetometer or axial gradiometer 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.

gradFlat <value / T/m>

Signal detection criterion for MEG planar gradiometers. The peak-to-peak value of all planar gradiometer signals must exceed this value, for the epoch to be included. This criterion allows rejection of data with saturated or otherwise dysfunctional channels. The default value is zero, *i.e.*, no rejection.

magFlat <value / T>

Signal detection criterion for MEG magnetometers and axial gradiometers channels.

eegFlat <value / V>

Signal detection criterion for EEG channels.

eogFlat <value / V>

Signal detection criterion for EOG channels.

ecgFlat <value / V>

Signal detection criterion for ECG channels.

stimIgnore <time / s>

Ignore this many seconds on both sides of the trigger when considering the epoch. This parameter is useful for ignoring large stimulus artefacts, *e.g.*, from electrical somatosensory stimulation.

fixSkew

Since the sampling of data and the stimulation devices are usually not synchronized, all trigger input bits may not turn on at the same sample. If this option is included in the off-line averaging description file, the following procedure is used to counteract this: if there is a transition from zero to a nonzero value on the digital trigger channel at sample n , the following sample will be checked for a transition from this nonzero value to another nonzero value. If such an event pair is found, the two events will be jointly considered as a transition from zero to the second non-zero value.

keepsamplemean

The means at individual samples will *not* be subtracted in the estimation of the covariance matrix. For details, see Section 4.17.2. This parameter is effective only for estimating the covariance matrix from epochs. It is recommended to specify this option. However, for compatibility with previous MNE releases, *keepsamplemean* is not on by default.

4.14.3 Covariance definitions

The covariance definitions starting with `def` specify the epochs to be included in the estimation 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 digital trigger channel. The interpretation of the values on the trigger channel can be further modified by the `ignore` and `mask` keywords. If multiple event parameters are present in a definition, all specified events will be included. If the event parameter is missing or set to zero, the covari-

ance matrix is computed over a section of the raw data, defined by the `tmin` and `tmax` parameters.

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` ($2^2 + 2^7 = 132$).

mask *<number>*

Works similarly to `ignore` except that a mask specifies the trigger channel bits to be included. For example, to look at trigger input lines one to three only, ignoring others, specify `mask 7` ($2^0 + 2^1 + 2^2 = 7$).

delay *<time / s>*

Adds a delay to the time of the occurrence of an event. Therefore, if this parameter is positive, the zero time point of the epoch will be later than the time of the event and, correspondingly, if the parameter is negative, the zero time point of the epoch will be earlier than the time of the event. By default, there will be no delay.

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 matrix estimation. 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 covariance matrix estimation.

basemin *<time / s>*

Synonym for `bmin`.

bmax *<time / s>*

End time point of the baseline, see above.

basemax *<time / s>*

Synonym for `bmax`.

4.15 Managing averages

This selection pops up a dialog which allows the management of computed averages. The controls in the dialog, shown in Figure 4.11, 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 4.11 The dialog for managing available averages.

In the example of Figure 4.11, 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. Note that this is now a scrolled window and some of the loaded data may be below or above the current view area.

4.16 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.

4.16.1 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.

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 4.12 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.

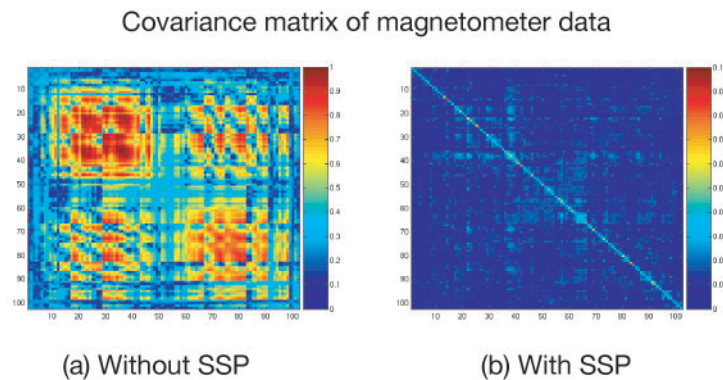


Figure 4.12 An example of the effect of SSP. The covariance matrix \mathbf{C}_n of noise data on the 102 Vectorview magnetometers was computed (a) before and (b) after the application of SSP with three-dimensional noise subspace. The plotted quantity is $\sqrt{|(\mathbf{C}_n)_{jk}|}$. Note that the vertical scale in (b) is ten times smaller than in (a).

4.16.2 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, also 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 basis for the noise subspace. It is also customary to use a separate set of vectors for magnetometers and gradiometers in the Vectorview system.

4.16.3 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 the projection:

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

where

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

4.17 Covariance matrix estimation

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

4.17.1 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 collect the accepted M samples from all channels to 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 band-pass filter should be applied.

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

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

4.17.2 Epochs

The calculation of the covariance matrix is slightly more complicated in the epoch mode. If the *bmin* and *bmax* parameters are specified in the covariance matrix description file (see Section 4.14.3), baseline correction is first applied to each epoch.

Let the vectors

$$\mathbf{s}_{rpj}, p = 1, \dots, P_r, j = 1, \dots, N_r, r = 1, \dots, R$$

be the samples from all channels in 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.

If the recommended *keepsamplemean* option is specified in the covariance matrix definition file, the baseline correction is applied to the epochs but the means at individual samples are not subtracted. Thus the covariance matrix will be computed as:

$$\hat{\mathbf{C}} = \frac{1}{N_C} \sum_{r,j,p} \mathbf{s}_{rpj} \mathbf{s}_{rpj}^T,$$

where

$$N_C = \sum_{r=1}^R N_r P_r.$$

If *keepsamplemean* is not specified, we estimate the covariance matrix as

$$\hat{\mathbf{C}} = \frac{1}{N_C} \sum_{r=1}^R \sum_{j=1}^{N_r} \sum_{p=1}^{P_r} (\mathbf{s}_{rpj} - \bar{\mathbf{s}}_{rj})(\mathbf{s}_{rpj} - \bar{\mathbf{s}}_{rj})^T,$$

where

$$\bar{\mathbf{s}}_{rj} = \frac{1}{P_r} \sum_{p=1}^{P_r} \mathbf{s}_{rpj}$$

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 . It is easy to see that the expression for the covariance matrix estimate can be cast into a more convenient form

$$\hat{\mathbf{C}} = \frac{1}{N_c} \sum_{r,j,p} \mathbf{s}_{rpj} \mathbf{s}_{rpj}^T - \frac{1}{N_c} \sum_r P_r \sum_j \bar{\mathbf{s}}_{rj} \bar{\mathbf{s}}_{rj}^T.$$

Subtraction of the means at individual samples is useful if it can be expected that the evoked response from previous stimulus extends to part of baseline period of the next one.

4.17.3 Combination of covariance matrix estimates

Let us assume that we have computed multiple covariance matrix estimates $\mathbf{C}_1 \dots \mathbf{C}_Q$ 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

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

4.17.4 SSP information included with covariance matrices

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.

4.18 Interacting with *mne_analyze*

To facilitate interactive analysis of raw data, *mne_browse_raw* can run *mne_analyze* as a child process. In this mode, *mne_analyze* is “remote controlled” by *mne_browse_raw* and will also send replies to *mne_browse_raw* to keep the two programs synchronized. A practical application of this communication is to view field or potential maps and cortically-constrained source estimates computed from raw data instantly.

The subordinate *mne_analyze* is started and stopped from *Start mne_analyze* and *Quit mne_analyze* in the *Windows* menu, respectively. The following settings are communicated between the two processes:

The raw data file

If a new raw data file is opened and a subordinate *mne_analyze* is active, the name of the raw data file is communicated to *mne_analyze* and a simplified version of the open dialog appears in *mne_analyze* allowing selection of an inverse operator or are MEG/MRI coordinate transformation. If a raw data file is already open in *mne_browse_raw* when *mne_analyze* is started, the open dialog appears immediately.

Time point

When a new time point is selected in *mne_browse_raw* the *mne_analyze* time point selection is updated accordingly. Time point selection in *mne_analyze* is not transferred to *mne_browse_raw*.

Scales

The vertical scales are kept synchronized between the two programs. In addition, the settings of the sample time limits are communicated from *mne_browse_raw* to *mne_analyze*.

Filter

The filter settings are kept synchronized.

CHAPTER 5 The forward solution

5.1 Overview

This Chapter covers the definitions of different coordinate systems employed in MNE software and FreeSurfer, the details of the computation of the forward solutions, and the associated low-level utilities.

5.2 MEG/EEG and MRI coordinate systems

The coordinate systems used in MNE software (and FreeSurfer) and their relationships are depicted in Figure 5.1. Except for the *Sensor coordinates*, all of the coordinate systems are Cartesian and have the “RAS” (Right-Anterior-Superior) orientation, *i.e.*, the x axis points to the right, the y axis to the front, and the z axis up.

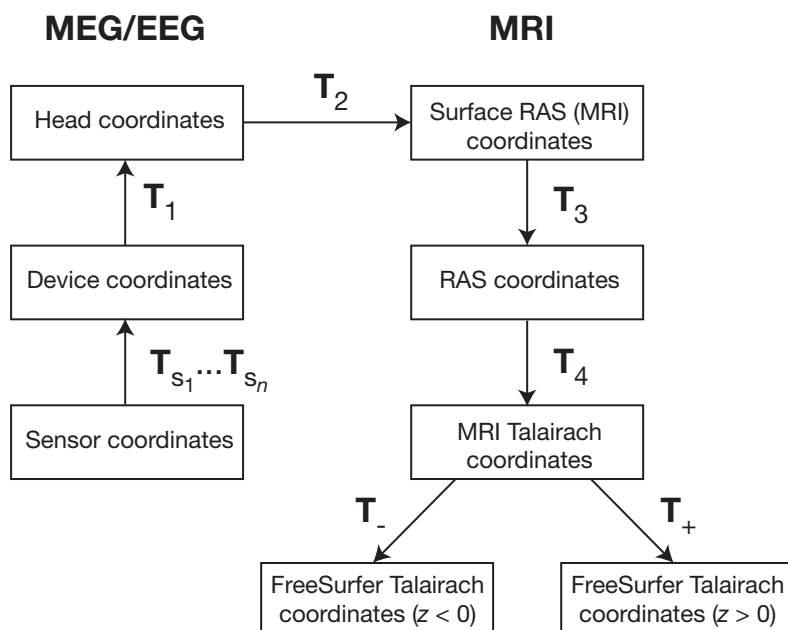


Figure 5.1 MEG/EEG and MRI coordinate systems. The coordinate transforms present in the *fif* files in MNE and the FreeSurfer files as well as those set to fixed values are indicated with T_x , where x identifies the transformation.

The coordinate systems related to MEG/EEG data are:

Head coordinates

This is a coordinate system defined with help of the fiducial landmarks (nasion and the two auricular points). In *fif* files, EEG electrode locations are given in this coordinate system. In addition, the head digitization data acquired in the beginning of an MEG, MEG/EEG, or EEG acquisition are expressed in head coordinates. For details, see Section 5.2.

Device coordinates

This is a coordinate system tied to the MEG device. The relationship of the Device and Head coordinates is determined during an MEG measurement by feeding current to 3 – 5 head-position indicator (HPI) coils and by determining their locations with respect to the MEG sensor array from the magnetic fields they generate.

Sensor coordinates

Each MEG sensor has a local coordinate system defining the orientation and location of the sensor. With help of this coordinate system, the numerical integration data needed for the computation of the magnetic field can be expressed conveniently as discussed in Section 5.8. The channel information data in the *fif* files contain the information to specify the coordinate transformation between the coordinates of each sensor and the MEG device coordinates.

The coordinate systems related to MRI data are:

Surface RAS coordinates

The FreeSurfer surface data are expressed in this coordinate system. The origin of this coordinate system is at the center of the conformed FreeSurfer MRI volumes (usually 256 x 256 x 256 isotropic 1-mm³ voxels) and the axes are oriented along the axes of this volume. The BEM surface and the locations of the sources in the source space are usually expressed in this coordinate system in the *fif* files. In this manual, the *Surface RAS coordinates* are usually referred to as *MRI coordinates* unless there is need to specifically discuss the different MRI-related coordinate systems.

RAS coordinates

This coordinate system has axes identical to the *Surface RAS coordinates* but the location of the origin is different and defined by the original MRI data, *i.e.*, the origin is in a scanner-dependent location. There is hardly any need to refer to this coordinate system explicitly in the analysis with the MNE software. However, since the Talairach coordinates, discussed below, are defined with respect to *RAS coordinates* rather than the *Surface RAS coordinates*, the RAS coordinate system is implicitly involved in the transformation between *Surface RAS coordinates* and the two *Talairach* coordinate systems.

MNI Talairach coordinates

The definition of this coordinate system is discussed, *e.g.*, in <http://imaging.mrc-cbu.cam.ac.uk/imaging/MniTalairach>. This transformation is determined during the FreeSurfer reconstruction process.

FreeSurfer Talairach coordinates

The problem with the MNI Talairach coordinates is that the linear MNI Talairach transform does not match the brains completely to the Talairach brain. This is probably because the Talairach atlas brain is a rather odd shape, and as a result, it is difficult to match a standard brain to the atlas brain using an affine transform. As a result, the MNI brains are slightly larger (in particular higher, deeper and longer) than the Talairach brain. The differences are larger as you get further from the middle of the brain, towards the outside. The FreeSurfer Talairach coordinates mitigate this problem by adding an additional transformation, defined separately for negative and positive MNI Talairach z coordinates. These two transformations, denoted by \mathbf{T}_- and \mathbf{T}_+ in Figure 5.1, are fixed as discussed in <http://imaging.mrc-cbu.cam.ac.uk/imaging/MniTalairach> (*Approach 2*).

The different coordinate systems are related by coordinate transformations depicted in Figure 5.1. The arrows and coordinate transformation symbols (\mathbf{T}_x) indicate the transformations actually present in the FreeSurfer files. Generally,

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \mathbf{T}_{12} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & x_0 \\ R_{21} & R_{22} & R_{23} & y_0 \\ R_{31} & R_{32} & R_{33} & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix},$$

where x_k , y_k , and z_k are the location coordinates in two coordinate systems, \mathbf{T}_{12} is the coordinate transformation from coordinate system “1” to “2”, x_0 , y_0 , and z_0 is the location of the origin of coordinate system “1” in coordinate system “2”, and R_{jk} are the elements of the rotation matrix relating the two coordinate systems. The coordinate transformations are present in different files produced by FreeSurfer and MNE as summarized in Table 5.1. The fixed transformations \mathbf{T}_- and \mathbf{T}_+ are:

$$\mathbf{T}_- = \begin{bmatrix} 0.99 & 0 & 0 & 0 \\ 0 & 0.9688 & 0.042 & 0 \\ 0 & -0.0485 & 0.839 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{T}_+ = \begin{bmatrix} 0.99 & 0 & 0 & 0 \\ 0 & 0.9688 & 0.046 & 0 \\ 0 & -0.0485 & 0.9189 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note: This section does not discuss the transformation between the MRI voxel indices and the different MRI coordinates. However, it is important to note that in FreeSurfer, MNE, as well as in Neuromag software an integer voxel coordinate corresponds to the location of the center of a voxel. Detailed information on the FreeSurfer MRI systems can be found at <https://surfer.nmr.mgh.harvard.edu/fswiki/CoordinateSystems>.

Transformation	FreeSurfer	MNE
\mathbf{T}_1	Not present	Measurement data files Forward solution files (*-fwd.fif) Inverse operator files (*-inv.fif)
$\mathbf{T}_{s_1} \dots \mathbf{T}_{s_n}$	Not present	Channel information in files containing \mathbf{T}_1 .
\mathbf{T}_2	Not present	MRI description files Separate coordinate transformation files saved from <i>mne_analyze</i> Forward solution files Inverse operator files
\mathbf{T}_3	mri/*.mgz files	MRI description files saved with <i>mne_make_cor_set</i> if the input is in mgz or mgh format.
\mathbf{T}_4	mri/transforms/talairach.xfm	MRI description files saved with <i>mne_make_cor_set</i> if the input is in mgz or mgh format.
\mathbf{T}_-	Hardcoded in software	MRI description files saved with <i>mne_make_cor_set</i> if the input is in mgz or mgh format.
\mathbf{T}_+	Hardcoded in software	MRI description files saved with <i>mne_make_cor_set</i> if the input is in mgz or mgh format.

Table 5.1 Coordinate transformations in FreeSurfer and MNE software packages. The symbols \mathbf{T}_x are defined in Figure 5.1 Note: *mne_make_cor_set/mne_setup_mri* prior to release 2.6 did not include transformations \mathbf{T}_3 , \mathbf{T}_4 , \mathbf{T}_- , and \mathbf{T}_+ in the fif files produced.).

5.3 The head and device coordinate systems

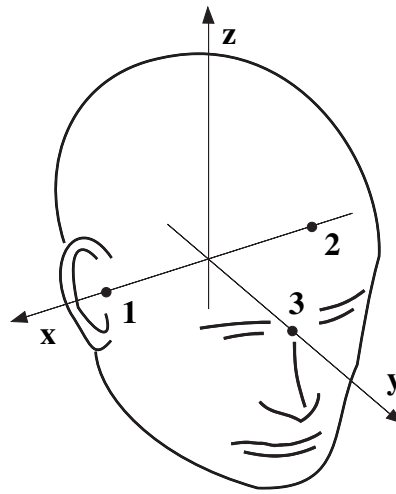


Figure 5.2 The head coordinate system

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 MEG/EEG (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 in this convention the auricular points are not necessarily located on y coordinate axis. The file conversion utilities (see Section 9.2) take care of these idiosyncrasies and convert all coordinate information to the MNE software head coordinate frame.

5.4 Creating a surface-based 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.

--subject <name>

Name of the subject in `SUBJECTS_DIR`. In the absense of this option, the `SUBJECT` environment variable will be consulted. If it is not defined, *mne_setup_source_space* exits with an error.

--morph <name>

Name of a subject in `SUBJECTS_DIR`. If this option is present, the source space will be first constructed for the subject defined by the `--subject` option or the `SUBJECT` environment variable and then morphed to this subject. This option is useful if you want to create a source spaces for several subjects and want to directly compare the data across subjects at the source space vertices without any morphing procedure afterwards. The drawback of this approach is that the spacing between source locations in the “morph” subject is not going to be as uniform as it would be without morphing.

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

FreeSurfer surface file names specifying the source surfaces, separated by colons.

--spacing <spacing/mm>

Specifies the approximate grid spacing of the source space in mm.

--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 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 `<dir>/<name>-src.fif`

Note: If both `--ico` and `--spacing` options are present the later one on the command line takes precedence.

Note: Due to the differences between the FreeSurfer and MNE libraries, the number of source space points generated with the `--spacing` option may be different between the current version of MNE and versions 2.5 or earlier (using `--spacing` option to `mne_setup_source_space`) if the FreeSurfer surfaces employ the (old) quadrangle format or if there are topological defects on the surfaces. All new FreeSurfer surfaces are specified as triangular tessellations and are free of defects.

5.5 Creating a volumetric or discrete source space

In addition to source spaces confined to a surface, the MNE software provides some support for three-dimensional source spaces bounded by a surface as well as source spaces comprised of discrete, arbitrarily located source points. The `mne_volume_source_space` utility assists in generating such source spaces.

The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--surf <name>

Specifies a FreeSurfer surface file containing the surface which will be used as the boundary for the source space.

--bem <name>

Specifies a BEM file (ending in `-bem.fif`). The inner skull surface will be used as the boundary for the source space.

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

If neither of the two surface options described above is present, the source space will be spherical with the origin at this location, given in MRI (RAS) coordinates.

--rad <radius/mm>

Specifies the radius of a spherical source space. Default value = 90 mm

--grid <spacing/mm>

Specifies the grid spacing in the source space.

--mindist <distance/mm>

Only points which are further than this distance from the bounding surface are included. Default value = 5 mm.

--exclude <distance/mm>

Exclude points that are closer than this distance to the center of mass of the bounding surface. By default, there will be no exclusion.

--mri <name>

Specifies a MRI volume (in *mgz* or *mgh* format). If this argument is present the output source space file will contain a (sparse) interpolation matrix which allows *mne_volume_data2mri* to create an MRI overlay file, see Section 9.4.

--pos <name>

Specifies a name of a text file containing the source locations and, optionally, orientations. Each line of the file should contain 3 or 6 values. If the number of values is 3, they indicate the source location, in millimeters. The orientation of the sources will be set to the z-direction. If the number of values is 6, the source orientation will be parallel to the vector defined by the remaining 3 numbers on each line. With `--pos`, all of the options defined above will be ignored. By default, the source position and orientation data are assumed to be given in MRI coordinates.

--head

If this option is present, the source locations and orientations in the file specified with the `--pos` option are assumed to be given in the MEG head coordinates.

--meters

Indicates that the source locations in the file defined with the `--pos` option are given in meters instead of millimeters.

--src <name>

Specifies the output file name. Use a name <dir>/<name>-src.fif

--all

Include all vertices in the output file, not just those in use. This option is implied when the **--mri** option is present. Even with the **--all** option, only those vertices actually selected will be marked to be “in use” in the output source space file.

5.6 Creating the BEM meshes

The *mne_surf2bem* utility converts surface triangle meshes from ASCII and FreeSurfer binary file formats to the fif format. The resulting fif 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.

5.6.1 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 5.6.2.

--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 5.6.2. The format of these files is described in Section 5.6.3.

--check

Check that the surfaces are complete and that they do not intersect. This is a recommended option. For more information, see Section 5.6.4.

--checkmore

In addition to the checks implied by the **--check** option, check skull and skull thicknesses. For more information, see Section 5.6.4.

--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.

5.6.2 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 some software packages have an opposite order. For these files, the **--swap** option is required. This option does not have any effect on the interpretation of the *FreeSurfer* surface files specified with the **--surf** option.

--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, 3 = 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 with *mri_watershed* are isomorphic with the 5th subdivision of a 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`.

5.6.3 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.

5.6.4 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.

5.7 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 5.6) 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`.

--method <approximation method>

Select the BEM approach. If *<approximation method>* is `constant`, the BEM basis functions are constant functions on each triangle and the collocation points are the midpoints of the triangles. With `linear`, the BEM basis functions are linear functions on each triangle and the collocation points are the vertices of the triangula-

tion. 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.

5.8 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.

5.8.1 The sensor coordinate system

The sensor 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 sensor 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 sensor 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

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

5.8.2 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 distributed 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.

5.8.3 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 5.2 lists the parameters of the *normal* coil geometry descriptions Table 5.3 lists the *accurate* descriptions. For *simple* accuracy, please consult the coil definition file, see Section 5.8.4. 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 sensor 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/share/mne/coil_def.dat`, which is automatically created by the utility `mne_list_coil_def`, see Section 5.8.5.

Id	Description	n	r/mm	w
2	Neuromag-122 planar gradiometer	2	$\pm(8.1, 0, 0)\text{mm}$	$\pm 1/16.2\text{mm}$
2000	A point magnetometer	1	$(0, 0, 0)\text{mm}$	1
3012	Vectorview type 1 planar gradiometer	2	$(\pm 8.4, 0, 0.3)\text{mm}$	$\pm 1/16.8\text{mm}$

Table 5.2 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
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
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 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
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	8	(± 3.875 , ± 3.875 , 0.0)mm (± 3.875 , ± 3.875 , 50.0)mm	1/4 -1/4

Table 5.2 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	(± 5.44 , ± 7.68 , 0)mm , (± 11.1 , ± 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	(± 5.89 , ± 6.71 , 0.3)mm , (± 10.8 , ± 6.71 , 0.3)mm	$\pm 1/(4 \cdot 16.69\text{mm})$
3013	Vectorview type 2 planar gradiometer	8	(± 5.89 , ± 6.71 , 0.3)mm , (± 10.8 , ± 6.71 , 0.3)mm	$\pm 1/(4 \cdot 16.69\text{mm})$
3022	Vectorview type 1 magnetometer	16	(± 9.68 , ± 9.68 , 0.3)mm , (± 9.68 , ± 3.23 , 0.3)mm (± 3.23 , ± 9.68 , 0.3)mm (± 3.23 , ± 3.23 , 0.3)mm	1/16
3023	Vectorview type 2 magnetometer	16	(± 9.68 , ± 9.68 , 0.3)mm , (± 9.68 , ± 3.23 , 0.3)mm (± 3.23 , ± 9.68 , 0.3)mm (± 3.23 , ± 3.23 , 0.3)mm	1/16
3024	Vectorview type 3 magnetometer	16	(± 7.88 , ± 7.88 , 0.3)mm , (± 7.88 , ± 2.63 , 0.3)mm (± 2.63 , ± 7.88 , 0.3)mm (± 2.63 , ± 2.63 , 0.3)mm	1/16
4001	Magnes WH magnetometer	7	(0.0, 0.0, 0.0)mm (± 9.390 , 0.0, 0.0)mm (± 4.695 , ± 8.132 , 0.0)mm	1/4 1/8 1/8

Table 5.3 Accurate coil descriptions

Id	Description	n	r/mm	w
4002	Magnes WH 3600 axial gradiometer	14	(0.0, 0.0, 0.0)mm (± 7.348 , 0.0, 0.0)mm (± 3.674 , ± 6.364 , 0.0)mm (0.0, 0.0, 50.0)mm (± 7.348 , 0.0, 50.0)mm (± 3.674 , ± 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	(± 20 , ± 20 , 0.0)mm (± 20 , ± 20 , 135)mm	1/4 -1/4
4005	Magnes reference gradiometer measuring off-diagonal gradients	8	(87.5, ± 20 , 0.0)mm (47.5, ± 20 , 0.0)mm (-87.5, ± 20 , 0.0)mm (-47.5, ± 20 , 0.0)mm	1/4 -1/4 1/4 -1/4
4004	Magnes reference gradiometer measuring diagonal gradients	8	(± 20 , ± 20 , 0.0)mm (± 20 , ± 20 , 135)mm	1/4 -1/4
5001	CTF 275 axial gradiometer	14	(0.0, 0.0, 0.0)mm (± 7.348 , 0.0, 0.0)mm (± 3.674 , ± 6.364 , 0.0)mm (0.0, 0.0, 50.0)mm (± 7.348 , 0.0, 50.0)mm (± 3.674 , ± 6.364 , 50.0)mm	1/4 1/8 1/8 -1/4 -1/8 -1/8
5002	CTF reference magnetometer	4	(± 4 , ± 4 , 0)mm	1/4
5003	CTF reference gradiometer measuring diagonal gradients	8	(± 8.6 , ± 8.6 , 0.0)mm (± 8.6 , ± 8.6 , 78.6)mm	1/4 -1/4
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

Table 5.3 Accurate coil descriptions

Id	Description	n	r/mm	w
6001	MIT KIT system axial gradiometer	14	(0.0, 0.0, 0.0)mm	1/4
			(±6.328, 0.0, 0.0)mm	1/8
			(±3.164, ±5.48, 0.0)mm	1/8
			(0.0, 0.0, 50.0)mm	−1/4
			(±6.328, 0.0, 50.0)mm	−1/8
			(±3.164, ±5.48, 50.0)mm	−1/8

Table 5.3 Accurate coil descriptions

5.8.4 The coil definition file

The coil geometry information is stored in the text file \$MNE_ROOT/share/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 5.4.

<id>

Coil id value. This value is listed in the first column of Tables 5.2 and 5.3.

<accuracy>

The coil representation accuracy. Possible values and their meanings are listed in Table 5.5.

<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 5.4 Coil class values

Value	Meaning
1	The simplest representation available
2	The standard or <i>normal</i> representation (see Table 5.2)
3	The most <i>accurate</i> representation available (see Table 5.3)

Table 5.5 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 5.2 and 5.3).

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

Indicates the location of the integration point (fourth column in Tables 5.2 and 5.3).

<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.

5.8.5 Creating the coil definition file

The standard coil definition file `$MNE_ROOT/share/mne/coil_def.dat` is included with the MNE software package. The coil definition file can be recreated with the utility *mne_list_coil_def* as follows:

```
mne_list_coil_def --out $MNE_ROOT/share/mne/coil_def.dat
```

5.9 Computing the forward solution

5.9.1 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*.

5.9.2 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 5.7. 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 approximate 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 5.9.4. If this option is absent the file `$HOME/.mne/EEG_models` will be consulted if it exists.

--eegmodel <model name>

Specifies the name of the sphere model to be used for EEG. If this option is missing, the model *Default* will be employed, see Section 5.9.4.

--eegrad <radius/mm>

Specifies the radius of the outermost surface (scalp) of the EEG sphere model, see Section 5.9.4. The default value is 90 mm.

--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. This option should be used only for surface-based and discrete source spaces.

--all

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

--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. With `--trans`, `--mri` option is not required.

--notrans

The MEG/MRI coordinate transformation is taken as the identity transformation, *i.e.*, the two coordinate systems are the same. This option is useful only in special circumstances. If more than one of the `--mri`, `--trans`, and `--notrans` options are specified, the last one remains in effect.

--mricoord

Do all computations in the MRI coordinate system. The forward solution matrix is not affected by this option if the source orientations are fixed to be normal to the cortical mantle. If all three source components are included, the forward three source orientations parallel to the coordinate axes is computed. If `--mricoord` is present, these axes correspond to MRI coordinate system rather than the default MEG head coordinate system. This option is useful only in special circumstances.

--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.

--grad

Include the derivatives of the fields with respect to the dipole position coordinates to the output, see Section 5.9.6.

5.9.3 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.

5.9.4 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 radiuses 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 radiuses 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 file is

not present, a model called *Default* is always provided. This model has the structure given in Table 5.6

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 5.6 Structure of the default EEG model

5.9.5 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 Moshier, Zhang, and Berg, see Section 13.3 (Moshier *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.

5.9.6 Field derivatives

If the `--grad` option is specified, *mne_forward_solution* includes the derivatives of the forward solution with respect to the dipole location coordinates to the output file. Let

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{g}_{xk} & \mathbf{g}_{yk} & \mathbf{g}_{zk} \end{bmatrix}$$

be the $N_{\text{chan}} \times 3$ matrix containing the signals produced by three orthogonal dipoles at location \mathbf{r}_k making up $N_{\text{chan}} \times 3N_{\text{source}}$ the gain matrix

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 & \dots & \mathbf{G}_{N_{\text{source}}} \end{bmatrix}.$$

With the `--grad` option, the output from *mne_forward_solution* also contains the $N_{\text{chan}} \times 9N_{\text{source}}$ derivative matrix

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & \dots & \mathbf{D}_{N_{\text{source}}} \end{bmatrix},$$

where

$$\mathbf{D}_k = \begin{bmatrix} \frac{\partial \mathbf{g}_{xk}}{\partial x_k} & \frac{\partial \mathbf{g}_{xk}}{\partial y_k} & \frac{\partial \mathbf{g}_{xk}}{\partial z_k} & \frac{\partial \mathbf{g}_{yk}}{\partial x_k} & \frac{\partial \mathbf{g}_{yk}}{\partial y_k} & \frac{\partial \mathbf{g}_{yk}}{\partial z_k} & \frac{\partial \mathbf{g}_{zk}}{\partial x_k} & \frac{\partial \mathbf{g}_{zk}}{\partial y_k} & \frac{\partial \mathbf{g}_{zk}}{\partial z_k} \end{bmatrix},$$

where x_k , y_k , and z_k are the location coordinates of the k^{th} dipole. If the dipole orientations are to the cortical normal with the `--fixed` option, the dimensions of \mathbf{G} and \mathbf{D} are $N_{\text{chan}} \times N_{\text{source}}$ and $N_{\text{chan}} \times 3N_{\text{source}}$, respectively. Both \mathbf{G} and \mathbf{D} can be read with the *mne_read_forward_solution* Matlab function, see Table 10.1.

5.10 Averaging forward solutions

5.10.1 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. The program averages both MEG and EEG forward solutions. Usually the EEG forward solution is identical across runs because the electrode locations do not change.

5.10.2 Command line options

mne_average_forward_solutions 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.

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 prior 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 strengths 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

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

where $\tilde{\mathbf{G}} = \mathbf{C}^{-1/2}\mathbf{G}$ is the spatially whitened gain matrix. The expected current values are $\hat{\mathbf{j}}(t) = \tilde{\mathbf{M}}\tilde{\mathbf{x}}(t)$, where $\tilde{\mathbf{x}}(t) = \mathbf{C}^{-1/2}\mathbf{x}(t)$ 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 un-whitened 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

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

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

$$\mathbf{A} = \tilde{\mathbf{G}}\mathbf{R}^{1/2} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$$

where the superscript $1/2$ indicates a square root of \mathbf{R} . For a diagonal matrix, one simply takes the square root of \mathbf{R} while in the more general case one can use the Cholesky factorization $\mathbf{R} = \mathbf{R}_C\mathbf{R}_C^T$ and thus $\mathbf{R}^{1/2} = \mathbf{R}_C$.

With the above SVD it is easy to show that

$$\tilde{\mathbf{M}} = \mathbf{R}^{1/2}\mathbf{V}\mathbf{\Gamma}\mathbf{U}^T$$

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 $\text{trace}(\mathbf{G}\mathbf{R}\mathbf{G}^T)/\text{trace}(\mathbf{I}) = 1$ 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 $\mathbf{j}(t)$ 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 $\hat{\mathbf{x}}(t)$, 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 orienta-

- tion. 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 coordinate 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-guided 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 weighting 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 correspond 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.

--noisecov <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. For backward compatibility, **--senscov** can be used as a synonym for **--noisecov**.

--noiserank <value>

Specifies the rank of the noise covariance matrix explicitly rather than trying to reduce it automatically. This option is seldom needed,

--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, an *a priori* weighting can be applied to the source covariance matrix. The source of the weighting is usually fMRI but may be also some other data, provided that the weighting 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 weight files, respectively. The application of the weighting 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.

--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 **--noisecov** 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 4.16. 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, jpg, png, 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**, **--jpg**, **--png**, and **--w** options control which file types are actually produced. When a **--pick** option is encountered, the effect of any preceding **--pickrange** option is ignored.

--pickrange

All previous **--pick** options will be ignored. Instead, snapshots are produced as indicated by the **--tmin**, **--tmax**, and **--tstep** options. This is useful, *e.g.*, for producing input for scripts merging the individual graphics snapshots into a composite “filmstrip” representation. However, such scripts are not yet part of the MNE software.

--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.

--morphgrade <number>

Adjusts the number of vertices in the stc files produced when morphing is in effect. By default the number of vertices is 10242 corresponding to `--morphgrade` value 5. Allowed values are 3, 4, 5, and 6 corresponding to 642, 2562, 10242, and 40962 vertices, respectively.

--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/share/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.

--nocomments

Do not include the comments in the image output files or movies.

--noscalebar

Do not include the scalebar in the image output files or movies.

--alpha <value>

Adjust the opacity of maps shown on the cortical surface (0 = transparent, 1 = totally opaque). The default value is 1.

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 overlayed color will 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 backwards 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 -<viname>. 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.

--rate <rate>

Specifies the frame rate of the QuickTime movies. The default value is $1/(10t_{\text{step}})$, where t_{step} is the time between subsequent movie frames produced in seconds.

--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 -<viname>. 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 -<viname>. 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.

--jpg <name>

Produce jpg 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, `.jpg` is added to indicate an rgb output file. Files are produced for all picked times as dictated by the `--pick` and `--integ` options.

--png <name>

Produce png 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, `.png` is added to indicate an rgb output file. Files are produced for all picked times as dictated by the `--pick` and `--integ` options.

--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.

--norm <name>

Indicates that a separate w file containing the noise-normalization values will be produced. The option `--spm` must also be present. Nevertheless, the movies and stc files output will contain MNE values. The noise normalization data files will be called <name>-<SNR>-lh.w and <name>-<SNR>-rh.w.

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. This option cannot be used with stc input files (`--stc in`) because the stc files do not contain the coordinates of the vertices.

--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 (Am). 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 **--stc** 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 and evoked 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 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.

--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.

--labeldir <directory>

All previous `--label` options will be ignored when this option is encountered. For each label in the directory, the output file defined with the `--out` option will contain a summarizing waveform which is the average of the waveforms in the vertices of the label. The `--labeldir` option implies `--align_z` and `--picknormal-comp` options.

--orignames

This option is used with the `--labeldir` option, above. With this option, the output file channel names will be the names of the label files, truncated to 15 characters, instead of names containing the vertex numbers.

--out <name>

Required with `--labeldir`. This is the output file for the data.

--extra <name>

By default, the output includes the current estimate signals and the digital trigger channel, see `--digtrig` option, below. With the `--extra` option, a custom set of additional channels can be included. The extra channel text file should contain the names of these channels, one channel name on each line. With this option present, the digital trigger channel is not included unless specified in the extra channel file.

--noextra

No additional channels will be included with this option present.

--digtrig <name>

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

--split <size/MB>

Specifies the maximum size of the raw data files saved. By default, the output is split into files which are just below 2 GB so that the fif file maximum size is not exceed.

Tip: The digital trigger channel can also be set with the `MNE_TRIGGER_CH_NAME` environment variable. Underscores in the variable value will *not* be replaced with spaces by `mne_compute_raw_inverse`. Using the `--digtrig` option supersedes the `MNE_TRIGGER_CH_NAME` environment variable.

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] <number>`, 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 inver-

sion 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 9.13. 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. Fitting of current dipoles to the data.
6. Connection to *tkmedit* (part of FreeSurfer) to display data on MRI slices.
7. Production of QuickTime™ movies and graphics snapshots in several image file formats.
8. Connection to *cliplab* (part of Elekta-Neuromag software) to produce graphics reports, see Section 7.20.
9. Inquiry and saving of source waveforms at selected surface points or within ROIs defined by label files.
10. On-line morphing of the current distributions.
11. Output of snapshots in *w* file format.
12. Display of overlay data delivered in *w* and *stc* file formats.
13. Creation of ROI (label) files.
14. Viewing of continuous head-position data delivered by Elekta-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.

--digtrig <name>

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

--digtrigmask <number>

Mask to be applied to the raw data trigger channel values before considering them. This option is useful if one wants to set some bits in a don't care state. For example, some finger response pads keep the trigger lines high if not in use, *i.e.*, a finger is not in place. Yet, it is convenient to keep these devices permanently connected to the acquisition system. The number can be given in decimal or hexadecimal format (beginning with 0x or 0X). For example, the value 255 (0xFF) means that only the lowest order byte (usually trigger lines 1 - 8 or bits 0 - 7) will be considered.

--visualizehpi

Start *mne_analyze* in the restricted *head position visualization* mode. For details, see Section 7.21.

--dig <filename>

Specify a file containing the head shape digitization data. This option is only usable if the *head position visualization* position visualization mode has been first invoked with the **--visualizehpi** option.

--hpi <filename>

Specify a file containing the transformation between the MEG device and head coordinate frames. This option is only usable if the *head position visualization* position visualization mode has been first invoked with the **--visualizehpi** option.

--scalehead

In *head position visualization* mode, scale the average scalp surface according to the head surface digitization data before aligning them to the scalp surface. This option is recommended.

--rthelmet

Use the room-temperature helmet surface instead of the MEG sensor surface when showing the relative position of the MEG sensors and the head in the *head position visualization* mode.

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

Note: Strictly speaking, trigger mask value zero would mean that all trigger inputs are ignored. However, for convenience, setting the mask to zero or not setting it at all has the same effect as `0xFFFFFFFF`, *i.e.*, all bits set.

Tip: The digital trigger channel can also be set with the `MNE_ENV_TRIGGER_CH` environment variable. Underscores in the variable value will *not* be replaced with spaces by *mne_analyze*. Using the `--digtrig` option supersedes the `MNE_ENV_TRIGGER_CH` environment variable.

Tip: The digital trigger channel mask can also be set with the `MNE_TRIGGER_CH_MASK` environment variable. Using the `--digtrigmask` option supersedes the `MNE_TRIGGER_CH_MASK` environment variable.

7.3 The main window

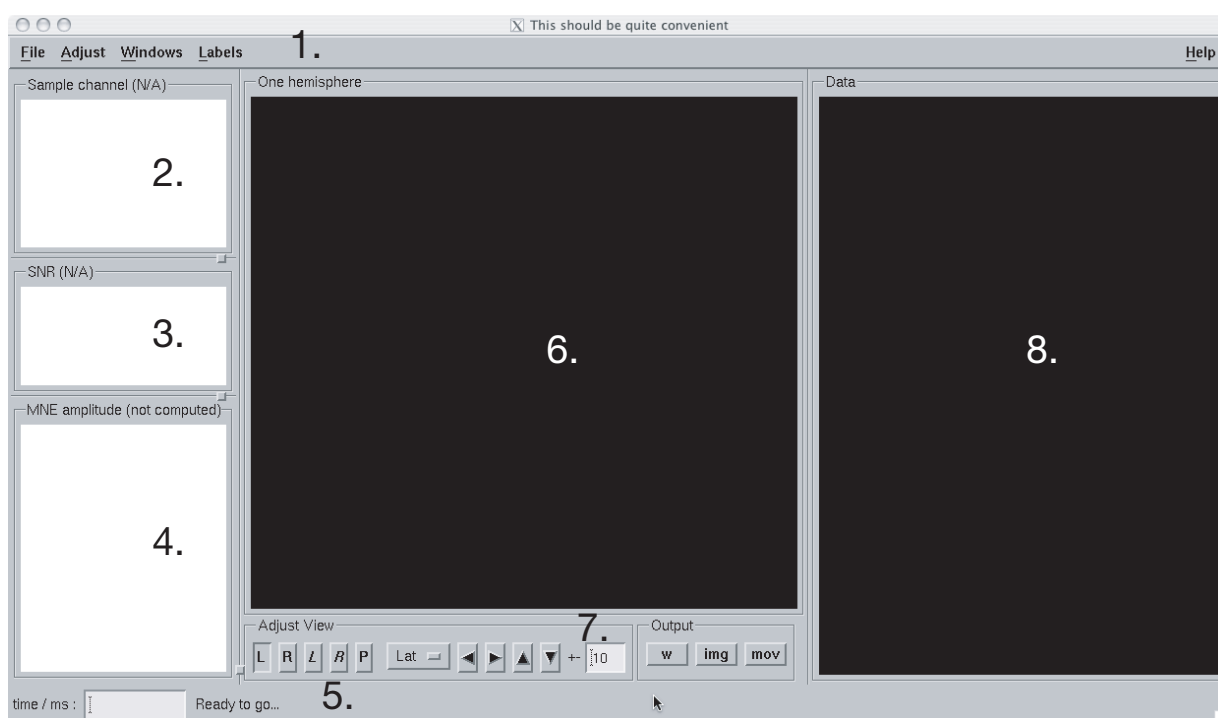


Figure 7.1 The main window of *mne_analyze*.

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, time-point selection text field, an ECD fit button, a text field for selecting a vertex on the surface, and a message text label;
6. Display area for the current estimates;
7. Controls for the current estimate display;
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:

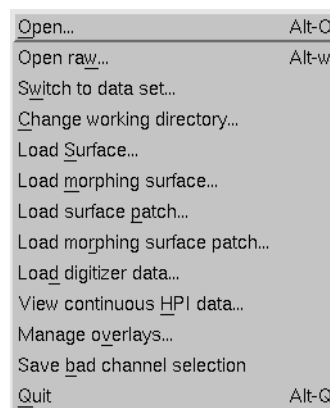


Figure 7.2 The *File* menu.

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:

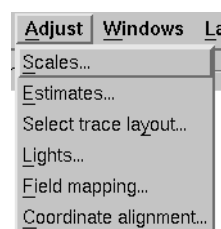


Figure 7.3 The *Adjust* menu.

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 *View* menu

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

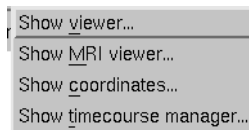


Figure 7.4 The *View* menu.

Show viewer...

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

Show MRI viewer...

Bring up the *tkmedit* program to view MRI slices, see Section 7.18.

Show coordinates...

Show the coordinates of a vertex, see Section 7.8.4.

Show timecourse manager...

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

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.

Load parcellation...

Load cortical parcellation data produced by *FreeSurfer* from directory \$SUBJECTS_DIR/\$SUBJECT/label and add the cortical regions defined to the label list.

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 *Dipoles* menu

The contents of the dipoles menu is shown in Figure 7.6:

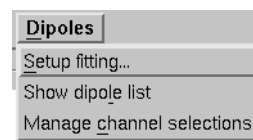


Figure 7.6 The dipole fitting menu.

Setup fitting...

Define the dipole fitting parameters, see Section 7.15.1.

Show dipole list...

Show the list of imported and fitted dipoles, see Section 7.15.3.

Manage channel selections...

Manage the selections of channels used in dipole fitting, see Section 7.15.4.

7.4.6 The *Help* menu

The contents of the *Help* menu is shown in Figure 7.7:

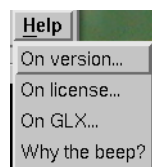


Figure 7.7 The *Help* menu.

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 7.8 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 using the *Select...* button, usually located in `mri/T1-neuromag/sets` under the subject's FreeSurfer directory. The *Default* button uses the default transformation file which must be called `$SUBJECTS_DIR/$SUBJECT/bem/$SUBJECT-trans.fif`. This can be one of the MRI description files in `mri/T1-neuromag/sets` or a transformation file stored from *mne_analyze*, see Section 7.16.
 - 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.

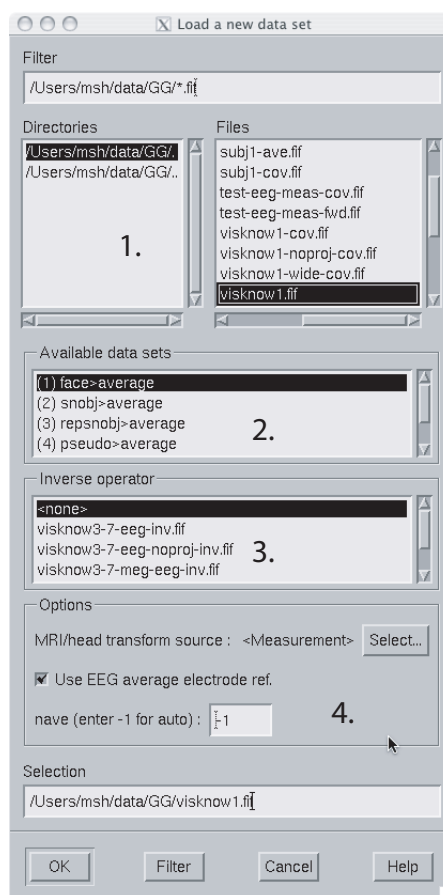


Figure 7.8 The open dialog.

After the data set(s) 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 7.8) except that data set selector is replaced by the epoch selector shown in Figure 7.9.

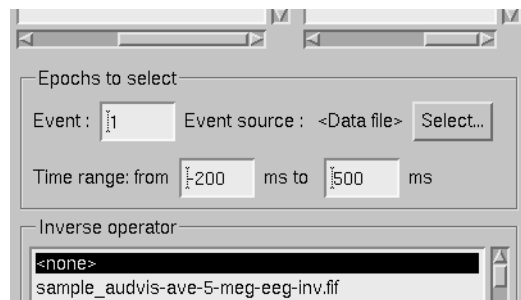


Figure 7.9 The raw data epoch selector.

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, *i.e.*, the digital trigger channel or an event data file produced with *mne_browse_raw* or *mne_process_raw*, see Section 4.4.10. Using an event data file is useful if, *e.g.*, the epochs to be processed 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 right most 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 system-wide layouts reside in `$MNE_ROOT/share/mne_analyze/lout`. In addition any layout files residing in `$HOME/.mne/lout` are listed. The format of the layout files and selection of the default layout is discussed in Section 4.5.6.

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 or drag

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 or drag with control key

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

Right button

Adjusts the channel selection used for dipole fitting in the same way as the middle button selects bad channels. For more information on channel selections, see Section 7.15.4.

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 7.10 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 digital trigger channel data in the sample view together with the sample channel.

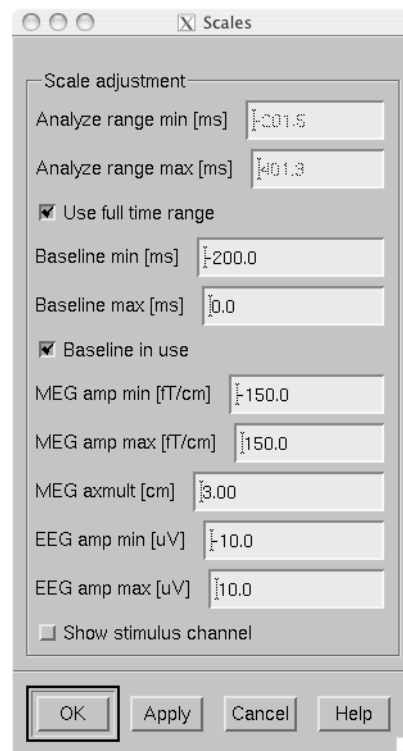


Figure 7.10 The *Scales* dialog.

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 7.11 appears.

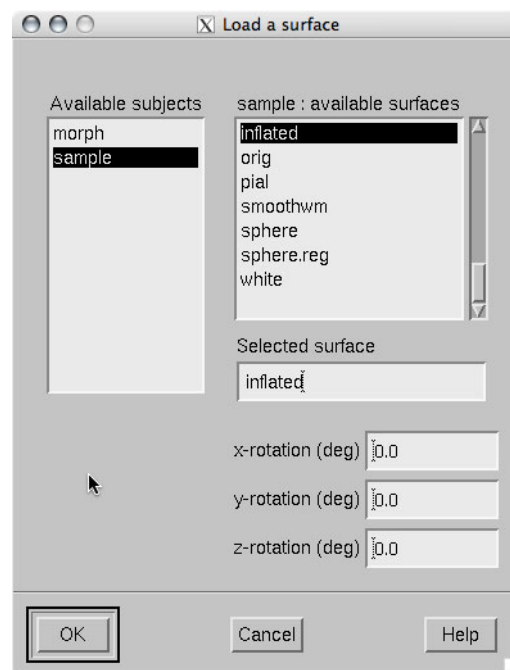


Figure 7.11 The surface selection dialog.

The dialog has the following components:

List of subjects

This list contains the subjects available in the directory set with the SUBJECTS_DIR environment variable.

List of available surfaces for the selected subject

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. Sometimes 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 7.12, 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).

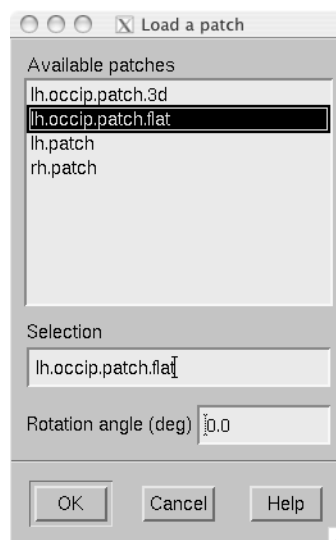


Figure 7.12 The patch selection dialog.

7.8.3 Controlling the surface display

The main surface display has a section called *Adjust view*, which has the controls shown in Figure 7.13:

L and R

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

B

Display the surfaces for both hemispheres.

M

Display the surfaces loaded *File/Load morphing surface...* according to the L, R, and B hemisphere selectors

P

Select the patch associated with the currently selected surface. For this to work, either L or R must be selected.

Option menu

Select one of the predefined view orientations, see Section 7.8.5, below.

Arrow buttons

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

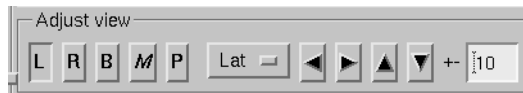


Figure 7.13 Surface controls.

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 to make it active:

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 7.8.7.

s

Produces a raster image file which contains a snapshot of the currently displayed image. For information on snapshot mode, see Section 7.8.7.

.

Stops the rotation invoked with the 'r' key, see above.

In addition, the mouse wheel or trackball can be used to rotate the image. If a trackball is available, *e.g.*, with the Apple MightyMouse, the image can be rotated up and down or left and right with the trackball. With a mouse wheel the image will rotated up and down when the wheel is rotated. Image rotation in the left-right direction is achieved by holding down the *shift* key when rotating the wheel. The *shift* key has the same effect on trackball operation.

Note: The trackball and mouse wheel functionality is dependent on your X server settings. On Mac OSX these settings are normally correct by default but on a LINUX system some adjustments to the X server settings maybe necessary. Consult your system administrator or Google for details.

7.8.4 Selecting vertices

When you click on the surface with the left mouse button, the corresponding vertex number and the associated value will be displayed on the message line at the bottom of the display. In addition, the time course at this vertex will be shown, see Section 7.13.1. You can also select a vertex by entering the vertex number to the text field at the bottom of the display. If the *MRI viewer* is displayed and *Track surface location in MRI* is selected in the *MRI viewer control* dialog, the cursor in the MRI slices will also follow the vertex selection, see Section 7.18.

The *View* menu choice *Show coordinates...* brings up a window which shows the coordinates of the selected vertex on the *white matter* surface, *i.e.*, lh.white and rh.white FreeSurfer surfaces. If morphing surfaces have been loaded, the coordinates of both the subject being analyzed and those of the morphing subject will be shown. The *Coordinates* window includes the following lines:

MEG head

Indicates the vertex location in the *MEG head* coordinates. This entry will be present only if MEG/EEG data have been loaded.

Surface RAS (MRI)

Indicates the vertex location in the *Surface RAS* coordinates. This is the native coordinate system of the surfaces and this entry will always be present.

MNI Talairach

Shows the location in MNI Talairach coordinates. To be present, the MRI data of the subject must be in the mgz format (usually true with any recent FreeSurfer version) and the Talairach transformation must be appropriately defined during the *FreeSurfer* reconstruction workflow.

Talairach

Shows the location in the *FreeSurfer* Talairach coordinates which give a better match to the Talairach atlas.

The above coordinate systems are discussed in detail in Section 5.2.

Note: By default, the *tkviewer* program, part of the *FreeSurfer* package, shows the vertex locations on the *orig* rather than *white* surfaces. Therefore, the coordinates shown in *mne_analyze* and *tkviewer* are by default slightly different (usually by < 1 mm). To make the two programs consistent, you can start *tkviewer* with the `-orig white` option.

7.8.5 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 `$MNE_ROOT/share/mne/mne_analyze/eyes`. If the file `$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 accommodate all the controls. The widths of the main window panes can be adjusted from the squares at the vertical sashes separating the panes.

7.8.6 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 7.14, 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.

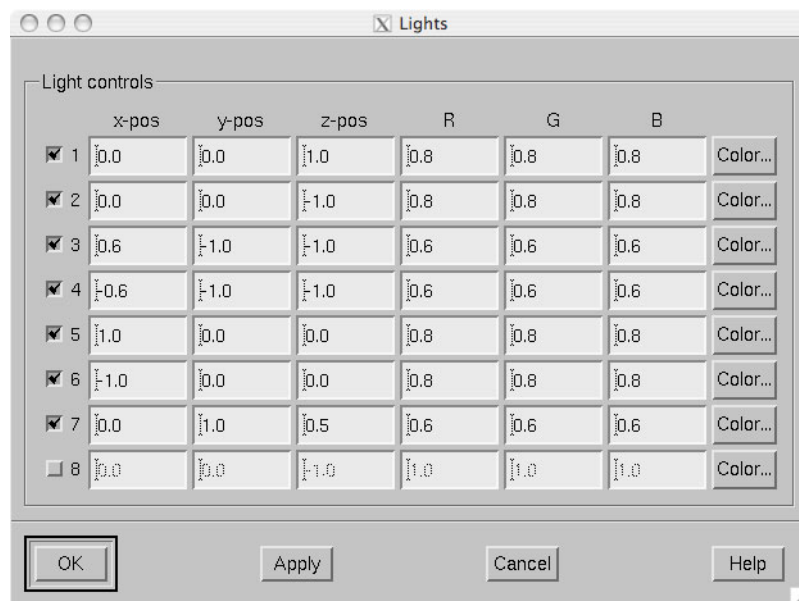


Figure 7.14 The lighting adjustment dialog.

7.8.7 Producing output files

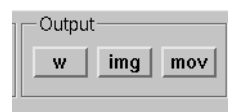


Figure 7.15 Graphics output controls.

Three types of output files can be produced from the main surface display using the graphics output buttons shown in Figure 7.15:

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 7.16. For more details, see Section 7.8.8. 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 7.17 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.

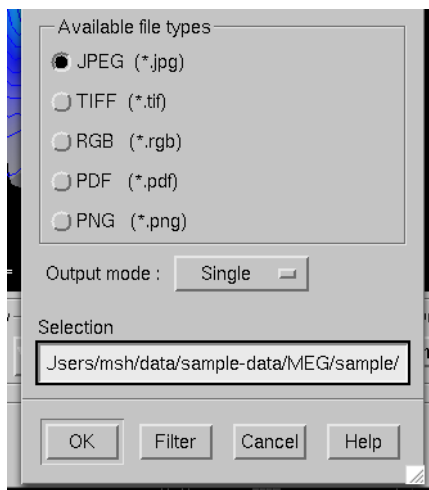


Figure 7.16 File type selection in the image saving dialog.

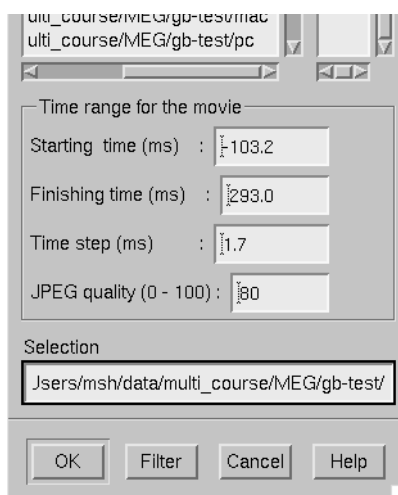


Figure 7.17 The controls in the movie saving dialog.

7.8.8 Image output modes

The image saving dialog shown in Figure 7.16 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 formats to work, the following programs must be present in your system: *tifftopdf*, *tifftopnm*, *pnmtjpeg*, and *pnmtopng*.

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 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

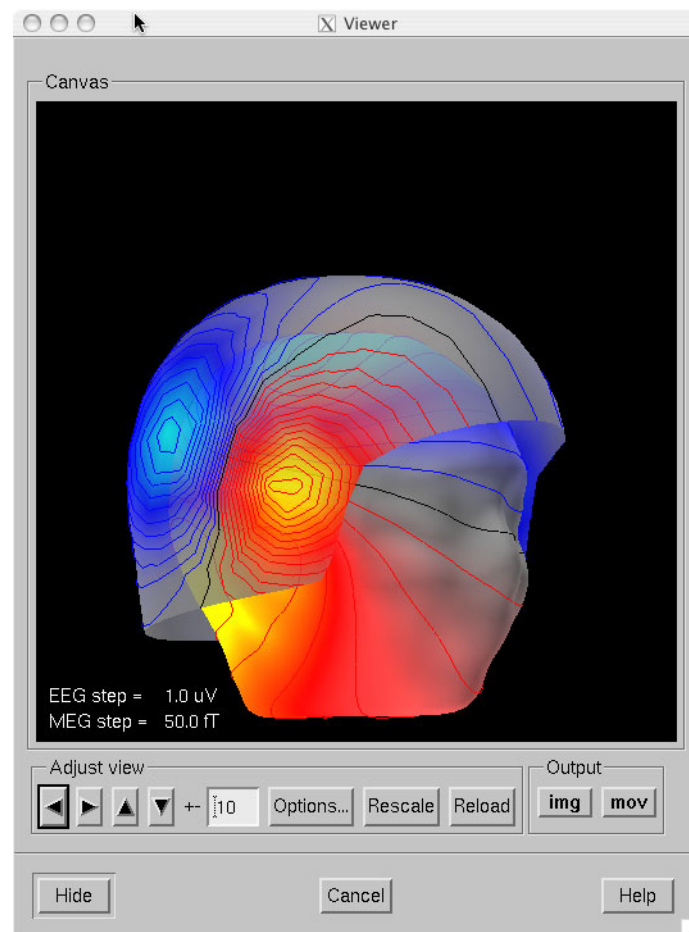


Figure 7.18 The viewer window with a visualization of MEG and EEG contour maps.

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.

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 5.6.

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 5.7. 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 7.18 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 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 7.8.7.

s

Produces a image file which contains a snapshot of the currently displayed image. For information on snapshot mode, see Section 7.8.7.

.

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.

In addition, the mouse wheel or trackball can be used to rotate the image. If a trackball is available, *e.g.*, with the Apple MightyMouse, the image can be rotated up and down or left and right with the trackball. With a mouse wheel the image will rotated up and down when the wheel is rotated. Image rotation in the left-right direction is achieved by holding down the *shift* key when rotating the wheel. The *shift* key has the same effect on trackball operation.

Note: The trackball and mouse wheel functionality is dependent on your X server settings. On Mac OSX these settings are normally correct by default but on a LINUX system some adjustments to the X server settings maybe necessary. Consult your system administrator or Google for details.

7.10.2 Viewer options

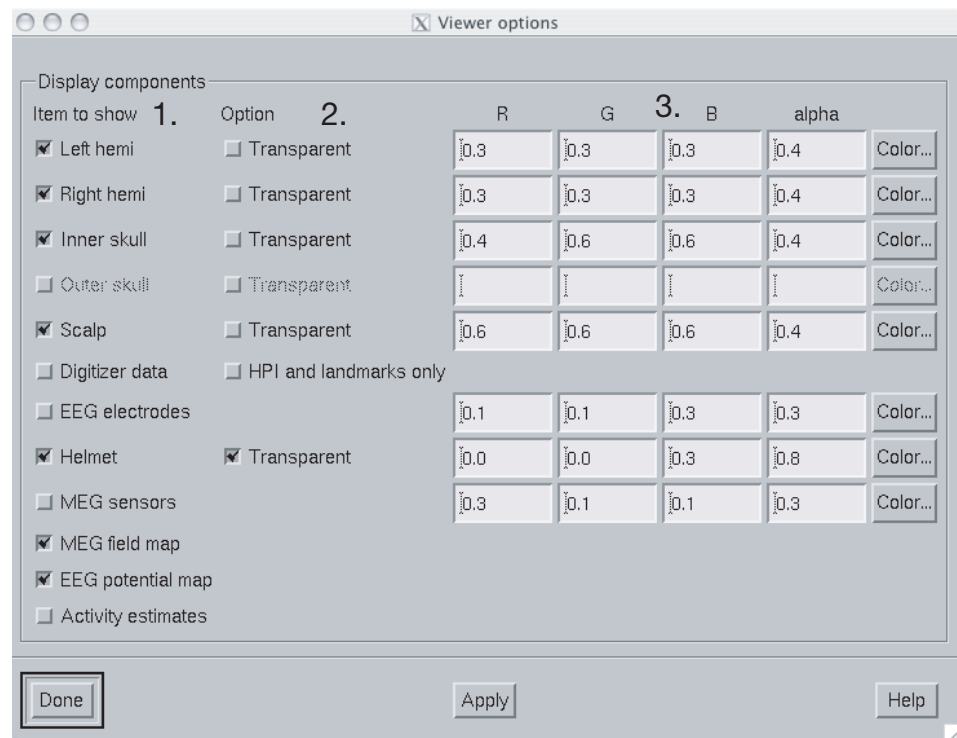


Figure 7.19 The viewer options window

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 can be displayed 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 can be displayed 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 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 Technical 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

$$1 - \frac{\sum_{j=1}^p \lambda_j}{\sum_{j=1}^N \lambda_j} < \varepsilon$$

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

$$\Gamma'_{jk} = \langle \mathbf{L}'_j | \mathbf{L}_k \rangle ,$$

where \mathbf{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 7.20 which is accessed through the *Adjust/Field mapping...* menu item.

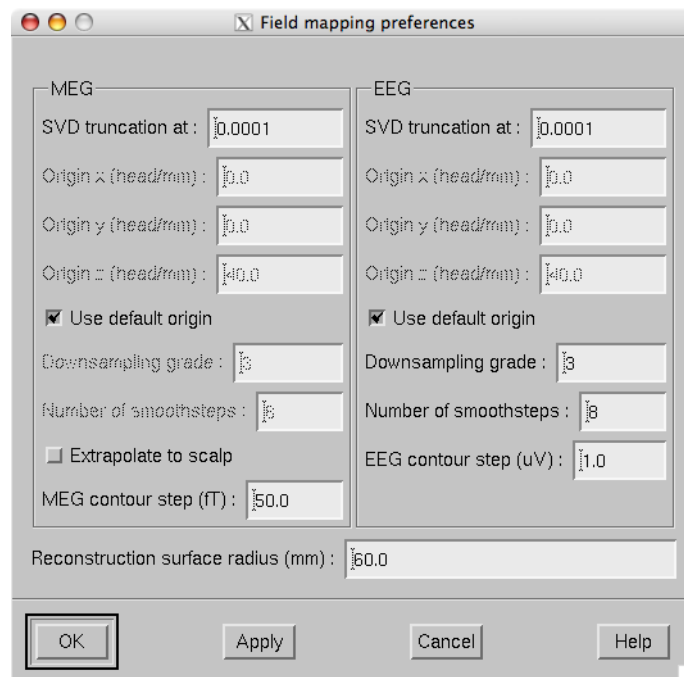


Figure 7.20 Field mapping preferences dialog.

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 7.21, 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 $tc_{mult} f_{mult} f_{max}$.

Table 7.1 The color scale parameters.

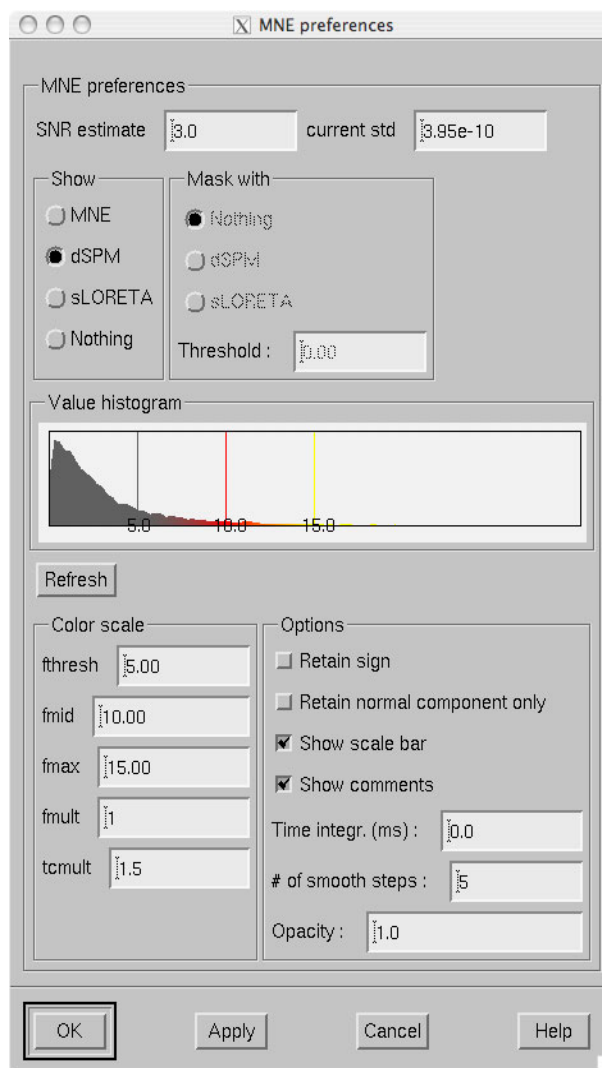


Figure 7.21 Estimate preferences dialog.

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 7.22 appears. Each time a new label is added to the list, the names will be reordered to alphabetical order. 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.

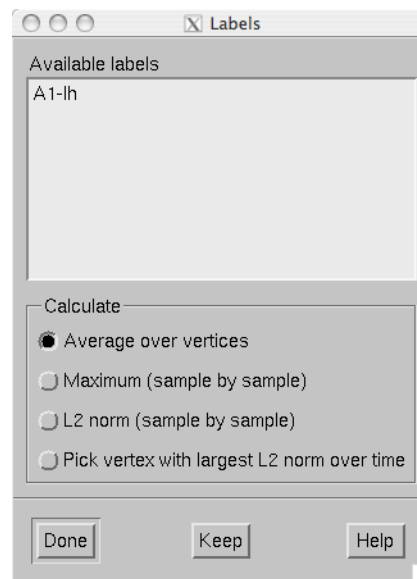


Figure 7.22 The label list.

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.

L2 norm (sample by sample)

Compute the l_2 norm over the values in the vertices at each time point.

Pick vertex with largest L2 norm over time

Compute the l_2 norm over time in each vertex and show the time course at the vertex with the largest norm.

7.13.3 The timecourse manager

The timecourse manager shown in Figure 7.23 has the following controls for each timecourse stored:

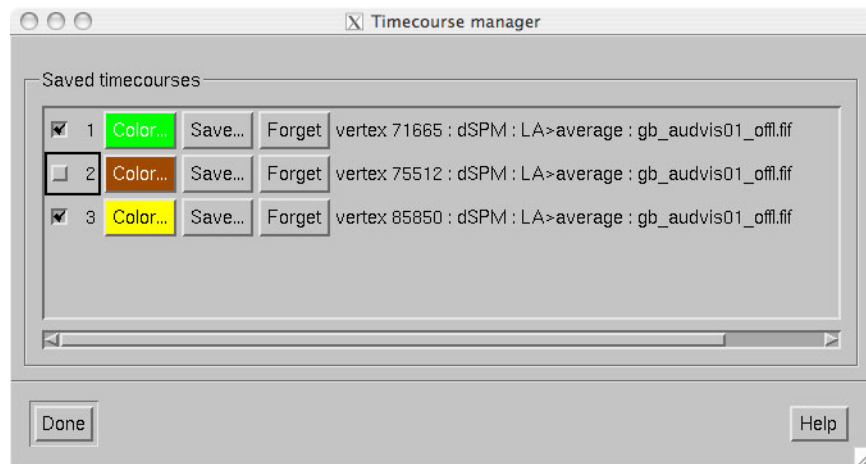


Figure 7.23 The timecourse manager.

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* 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 7.24. The vertex-by-vertex output formats is summarized in Table 7.2.

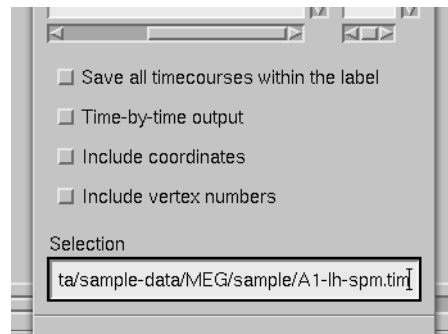


Figure 7.24 Label timecourse saving options.

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 time points, 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. The shortest path will be calculated along the white matter surface. Note that sometimes the shortest paths appear to be un-intuitive 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

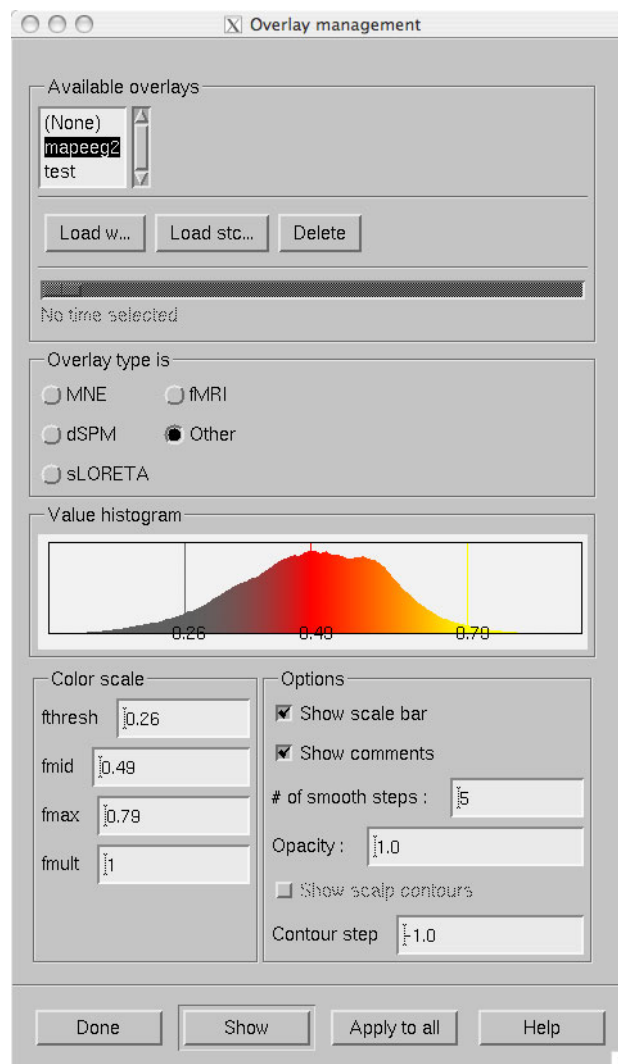


Figure 7.25 The overlay management dialog.

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 time slice (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 7.25 which is invoked from *File/Manage overlays....*

This dialog contains the following controls:

List of overlays loaded

Lists the names of the overlays loaded so far.

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 surface 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*, *fmid*, 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.

Show

Show the selected overlay and assign the settings to the current overlay.

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 Fitting current dipoles

Starting from MNE software version 2.6, *mne_analyze* includes routines for fitting current dipoles to the data. At present *mne_analyze* is limited to fitting single equivalent current dipole (ECD) at one time point. The parameters affecting the dipole fitting procedure are described in Section 7.15.1. The results are shown in the dipole list (Section 7.15.3). The selection of channels included can be adjusted interactively or by pre-defined selections as described in Section 7.15.4.

Warning: The current dipole fitting has been added recently and has not been tested comprehensively. Especially fitting dipoles to EEG data may be unreliable.

7.15.1 Dipole fitting parameters

Prior to fitting current dipoles, the fitting parameters must be set with the *Dipole fitting preferences* dialog shown in Figure 7.26. The dialog is brought up from the *Setup fitting...* choice in the *Dipoles* menu. This dialog contains three sections: *Forward model*, *Modalities*, and *Noise estimate*.

The *Forward model* section specifies the forward model to be used:

Sphere model origin x/y/z [mm]

Specifies the origin of the spherically symmetric conductor model in MEG/EEG head coordinates, see Section 5.3.

EEG scalp radius [mm]

Specifies the radius of the outermost shell in the EEG sphere model. For details, see Section 5.9.4.

EEG sphere model name

Specifies the name of the EEG sphere model to use. For details, see Section 5.9.4.

BEM model

Selects the boundary-element model to use. The button labeled with ... brings up a file-selection dialog to select the BEM file. An existing selection can be cleared with the *Unset* button. If EEG data are included in fitting, this must be a three-compartment model. Note that the sphere model is used even with a BEM model in effect, see Section 7.15.2.

Accurate field calculation

Switches on the more accurate geometry definition of MEG coils, see Section 5.8. In dipole fitting, there is very little difference between the *accurate* and *normal* coil geometry definitions.

The *Modalities* section defines which kind of data (MEG/EEG) are used in fitting. If an inverse operator is loaded with the data, this section is fixed and greyed out. You can further restrict the selection of channels used in dipole fitting with help of channel selections discussed in Section 7.15.4.

The *Noise estimate* section of the dialog contains the following items:

Noise covariance

Selects the file containing the noise-covariance matrix. If an inverse operator is loaded, the default is the inverse operator file. The button labeled with ... brings up a file-selection dialog to select the noise covariance matrix file. An existing selection can be cleared with the *Unset* button.

Omit off-diagonal terms

If a noise covariance matrix is selected, this choice omits the off-diagonal terms from it. This means that individual noise estimates for each channel are used but correlations among channels are not taken into account.

Regularization

Regularize the noise covariance before using it in whitening by adding a multiple of an identity matrix to the diagonal. This is discussed in more detail in Section 6.2.4. Especially if EEG is included in fitting it is advisable to enter a non-zero value (around 0.1) here.

Planar fixed [fT/cm]

In the absense of a noise covariance matrix selection, a diagonal noise covariance with fixed values on the diagonal is used. This entry specifies the fixed value of the planar gradiometers.

Axial fixed [fT]

If a noise covariance matrix file is not specified, this entry specifies a fixed diagonal noise covariance matrix value for axial gradiometers and magnetometers.

EEG fixed [μ V]

If a noise covariance matrix file is not specified, this entry specifies a fixed diagonal noise covariance matrix value for axial gradiometers and magnetometers..

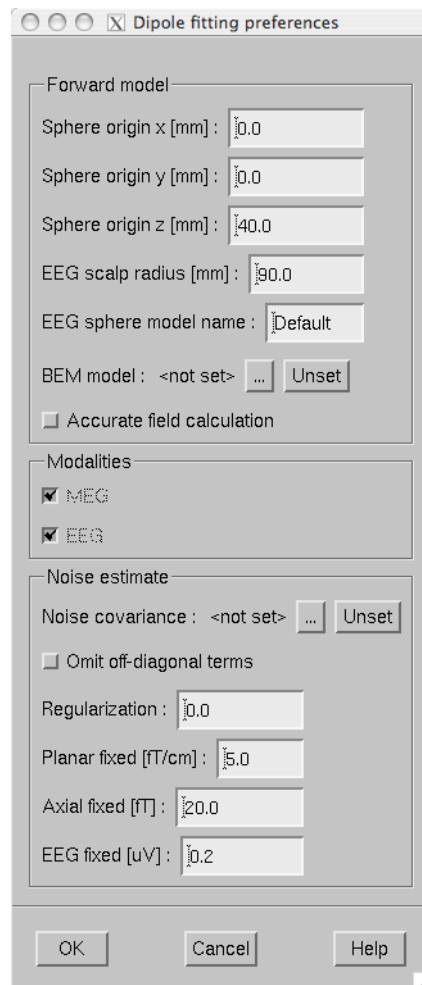


Figure 7.26 The dipole fitting preferences dialog.

7.15.2 The dipole fitting algorithm

When the dipole fitting preferences dialog is closed and the values have been modified the following preparatory calculations take place:

1. If EEG data are included in fitting present, the EEG sphere model specification corresponding to *EEG sphere model name* is loaded and scaled to the the *EEG scalp radius*.
2. If a boundary-element model is used, the additional data depending on the sensor locations are computed.
3. The noise covariance matrix is composed according to the specifications in the *Dipole fitting preferences* dialog.
4. The spatially whitened forward solution is computed in a grid of locations to establish the initial guess when a dipole is fitted. If a BEM is in use, the grid will be confined to the inner skull volume. For a sphere model, a spherical volume with an 80-mm radius, centered at the sphere model origin, will be employed. The dipole grid will be rectangular with a 10-mm spacing between the closest dipole locations. Any locations closer than 20 mm to the center of mass of the grid volume will be excluded as well as those closer than 10 mm to the surface. Note that this guess grid is only used for establishing the initial guess; the actual dipole fitting procedure does not constrain the solution to this grid.

When the *Fit ECD* button in the tool bar is clicked with a time point selected from the the response, the optimal Equivalent Current Dipole parameters (location, orientation, and amplitude) are determined using the following algorithm:

1. An initial guess for the location of the dipole is determined using the grid of locations determined in step 4., above. At each guess dipole location, the least squares error between the measured data and a dipole at that location is evaluated and the location corresponding to the smallest error is used as the initial guess location. In this process, the dipole amplitude parameters do not need to be explicitly calculated.
2. Using the Nelder-Mead simplex optimization algorithm, an optimal dipole location is determined with the sphere model used as the forward model. Again, the dipole amplitude parameters are not explicitly present in the fitting procedure.
3. A second optimization iteration using the boundary-element model (if available) or the sphere model as the forward model is conducted. The reason for repeating the optimization even with the sphere model is to reduce the likelihood of having been stuck in a local minimum of the least squares error criterion.
4. The optimal dipole amplitude parameters are determined for the optimal dipole location obtained in steps 2. and 3.
5. The dipole parameters are reported in the dipole list discussed in Section 7.15.3.

Additional notes:

1. The noise covariance matrix is always applied to the data and the forward solution as appropriate to correctly weight the different types of MEG channels and EEG. Depending on the dipole fitting settings, the

- noise covariance may be either a diagonal matrix or a full matrix including the correlations.
- Using the SVD of the whitened gain matrix of three dipole components at a given location, the component producing the weakest signal amplitude is omitted if the ratio of the smallest and largest singular values is less than 0.2.
 - The present MNE software package also contains a batch-mode dipole fitting program called *mne_dipole_fit*. This piece of software is not yet documented here. However, `mne_dipole_fit --help` lists the command-line options which have direct correspondence to the interactive dipole fitting options discussed here.

7.15.3 The dipole list

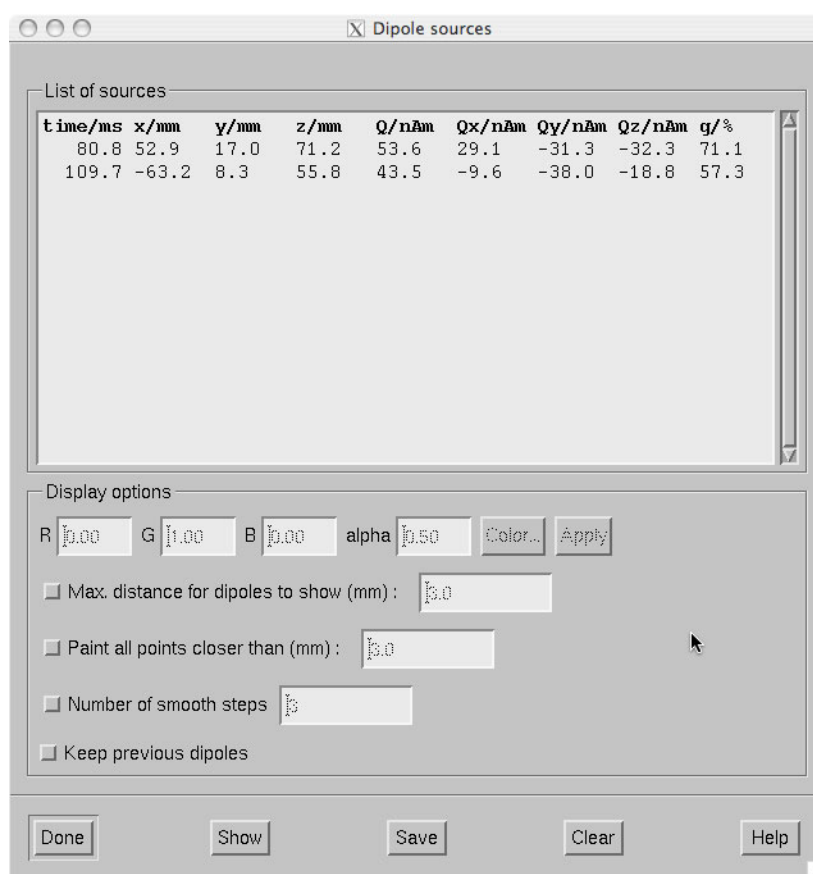


Figure 7.27 The dipole list.

The dipole list dialog shown in Figure 7.27 contains the parameters of the dipoles fitted. In addition, it is possible to 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.

The buttons at the bottom of the dialog perform the following functions:

Done	Hide the dialog.
Show	Show the currently selected dipoles as specified in <i>Display options</i> , see below.
Save	Save the selected (or all) dipoles. If the file name specified in the file selection dialog that pops up ends with .bdip , the dipole data will be saved in the binary <i>bdip</i> format compatible with the Neuromag <i>xfit</i> software, otherwise, a text format output will be used. In the text file, comments will be included on lines starting with the percent sign so that the text format can be easily loaded into Matlab.
Clear	Clear the selected dipoles from the list.

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 surface must be loaded to display dipole locations. To calculate the distance from the dipoles to the white matter surface, the white matter tessellation is loaded as needed. Depending on the precise location of the fitted dipole, the spot indicating the dipole site 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.

7.15.4 Channel selections

As mentioned in Section 7.7.1, the right mouse button in the topographical display of channels can be used to restrict the selection of channels taken into account in dipole fitting. In addition, the channel selections can be manipulated in the channel selection window, which pops up from *Dipoles/Manage channel selections....* Initially this dialog contains the selections defined in or `$HOME/.mne/mne_analyze.sel` or `$MNE_ROOT/share/mne/mne_analyze/mne_analyze.sel`, the personal file taking precedence over the system wide default. The *Save* button in this dialog save the current set of channel selections to the personal selection file. The format of this file is identical to the channel selection file in *mne_browse_raw*.

When a channel selection file is in effect, the variances of the unselected channels are increased by a factor of 900. This means that unselected channels receive virtually no weight in the least-squares error function or, equivalently, that they are considered to be 30 times more noisy than their true noise value. Since this implementation of channel selections requires recomputation of the initial guess candidate data discussed in Section 7.15.2, above, changing the selection may take a finite amount of time, especially if a BEM is used for the forward calculation.

Important: Please note that when making a channel selection in the topographical displays, the channels not present in a particular layout are also affected. For example, if you select channels in a layout showing the Vectorview planar gradiometers, the magnetometer channels and EEG channels will be unselected.

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....*

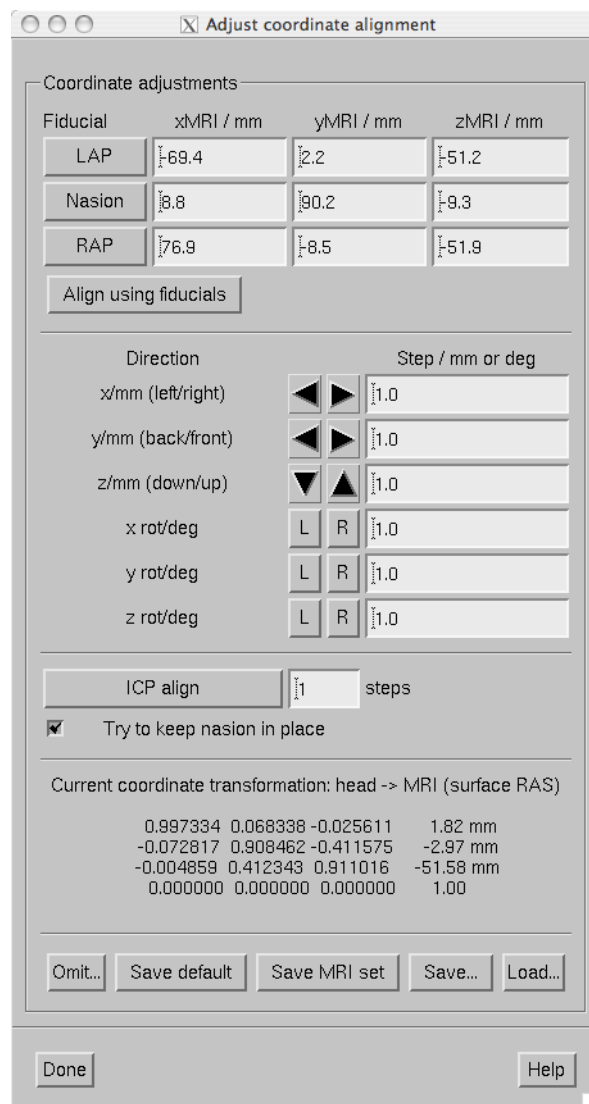


Figure 7.28 The coordinate frame alignment dialog.

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...*), and for saving and loading the coordinate transformation.

The saving and loading choices are:

Save default

Saves a file which contains the MEG/MRI coordinate transformation only. The file name is generated from the name of the file from which the digitization data were loaded by replacing the ending `.fif` with `-trans.fif`. If this file already exists, it will be overwritten without any questions asked.

Save MRI set

This option searches for a file called `COR.fif` in `$SUBJECTS_DIR/$SUBJECT/mri/T1-neuromag/sets`. The file is copied to `COR-<username>-<date>-<time>.fif` and the current MEG/MRI coordinate transformation as well as the fiducial locations in MRI coordinates are inserted.

Save...

Saves a file which contains the MEG/MRI coordinate transformation only. The ending `-trans.fif` is recommended. The file name selection dialog as a button to overwrite.

Load...

Loads the MEG/MRI coordinate transformation from the file specified.

The MEG/MRI coordinate transformation files are employed in the forward calculations. The convenience script *mne_do_forward_solution* described in Section 3.11 uses a search sequence which is compatible with the file naming conventions described above. It is recommended that *-trans.fif* file saved with the *Save default* and *Save...* options in the *mne_analyze* alignment dialog are used because then the `$SUBJECTS_DIR/$SUBJECT` directory will be composed of files which are dependent on the subjects' anatomy only, not on the MEG/EEG data to be analyzed.

Each iteration step of the Iterative Closest Point (ICP) algorithm consists of two matching 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.

Tip: One possible practical approach to coordinate frame alignment is discussed in Section 12.11.

7.16.1 Using a high-resolution head surface tessellations

The newest 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`.

If you have Matlab software available on your system, you can also benefit from the script *mne_make_scalp_surfaces*. This script invokes *mkheadsurf* and subsequently decimates it using the `mne_reduce_surface` function in the MNE Matlab toolbox, which in turn invokes the `reducepatch` Matlab function. As a result, the `$SUBJECTS_DIR/$SUBJECT/bem` directory will contain ‘dense’, ‘medium’, and ‘sparse’ scalp surface tessellations. The dense tessellation contains the output of *mkheadsurf*

while the medium and sparse tessellations comprise 30,000 and 2,500 triangles, respectively. You can then make a symbolic link of one of these to `<subject>-head.fif`. The medium grade tessellation is an excellent compromise between geometric accuracy and speed in the coordinate system alignment.

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.

7.16.2 Using fiducial points identified by other software

If you have identified the three fiducial points in software outside *mne_analyze*, it is possible to display this information on the head surface visualization. To do this, you need to copy the file containing the fiducial location information in MRI (surface RAS) coordinates to `$SUBJECTS_DIR/$SUBJECT/bem/$SUBJECT-fiducials.fif`. There are three supported ways to create this file:

1. Use the `mne_make_fiducial_file.m` Matlab function (not yet written) to create this file.
2. Copy a MRI description file with the MEG-MRI coordinate transformation created with MRILab (typically `$SUBJECTS_DIR/$SUBJECT/mri/T1-neuromag/sets/COR-<date>.fif`) to `$SUBJECTS_DIR/$SUBJECT/bem/$SUBJECT-fiducials.fif`.
3. For the average subject, *fsaverage*, copy the `fsaverage-fiducials.fif` file provided with *mne_analyze* in place, see Section 7.19.

7.17 Viewing continuous HPI data

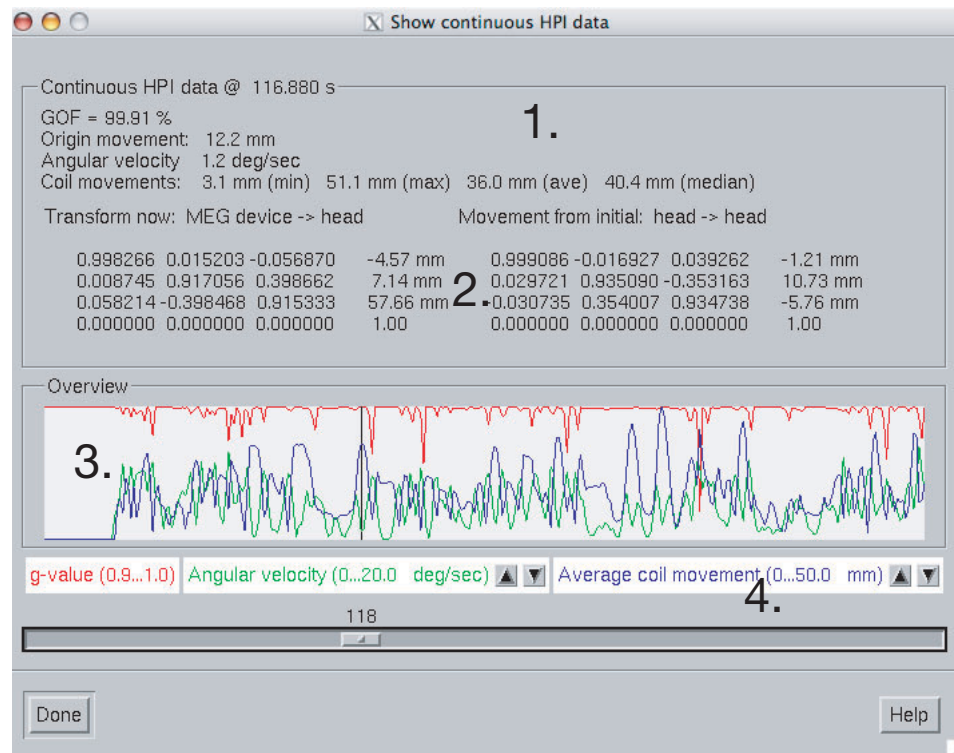


Figure 7.29 Continuous HPI data overview.

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 7.29 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.

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.

7.18 Working with the MRI viewer

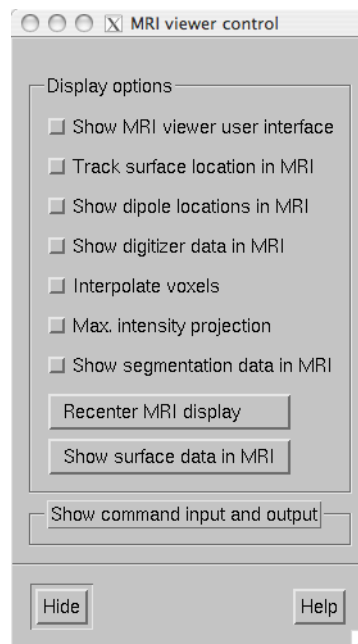


Figure 7.30 The MRI viewer control window.

Selecting *Show MRI viewer...* from the *View* menu starts the *FreeSurfer* MRI viewer program *tkmedit* to work in conjunction with *mne_analyze*. After a few moments, both *tkmedit* with the current subject's T1 MRI data shown and the MRI viewer control window shown in Figure 7.30 appear. Note that the *tkmedit* user interface is initially hidden. The surfaces of a subject must be loaded before starting the MRI viewer.

The MRI viewer control window contains the following items:

Show MRI viewer user interface

If this item is checked, the *tkmedit* user interface window will be shown.

Track surface location in MRI

With this item checked, the cursor in the MRI data window follows the current (clicked) location in surface display or viewer. Note that for the *viewer* window the surface location will be inquired from the surface closest to the viewer. The MEG helmet surface will not be considered. For example, if you click at an EEG electrode location with the scalp surface displayed, the location of that electrode on the scalp will be shown. The cortical surface locations are inquired from the white matter surface.

Show dipole locations in MRI

If this option is selected, whenever a dipole is displayed in the surface view using the dipole list dialog discussed in Section 7.15.3 the cursor will also move to the same location in the MRI data window.

Show digitizer data in MRI

If digitizer data are loaded, this option shows the locations with green diamonds in the MRI data.

Interpolate voxels

Toggles trilinear interpolation in the MRI data on and off.

Max. intensity projection

Shows a maximum-intensity projection of the MRI data. This is useful in conjunction with the *Show digitizer data in MRI* option to evaluate the MEG/MRI coordinate alignment

Recenter MRI display

Brings the cursor to the center of the MRI data.

Show surface data in MRI

This button creates an MRI data set containing the surface data displayed and overlays in with the MRI slices shown in the MRI viewer.

Show segmentation data in MRI

If available, the standard automatically generated segmentation volume (`mri/aparc+aseg`) is overlaid on the MRI using the standard FreeSurfer color lookup table (`$FREESURFER_HOME/FreeSurferColorLUT.txt`). As a result, the name of the brain structure or region of cortex at the current location of the cursor will be reported if the *tkmedit* user interface is visible. After the segmentation is loaded this button toggles the display of the segmentation on and off.

Show command input and output

Allows sending *tcl* commands to *tkmedit* and shows the responses received. The *tkmedit* *tcl* scripting commands are discussed at <https://surfer.nmr.mgh.harvard.edu/fswiki/TkMeditGuide/TkMeditReference/TkMeditScripting>.

7.19 Working with the average brain

The FreeSurfer software includes an average subject (`fsaverage`) with a cortical surface reconstruction. In some cases, the average subject can be used as a surrogate if the MRIs of a subject are not available.

The MNE software comes with additional files which facilitate the use of the average subject in conjunction with *mne_analyze*. These files are located in the directory `$MNE_ROOT/mne/setup/mne_analyze/fsaverage`:

fsaverage_head.fif

The approximate head surface triangulation for fsaverage.

fsaverage_inner_skull-bem.fif

The approximate inner skull surface for fsaverage.

fsaverage-fiducials.fif

The locations of the fiducial points (LPA, RPA, and nasion) in MRI coordinates, see Section 7.16.2.

fsaverage-trans.fif

Contains a default MEG-MRI coordinate transformation suitable for fsaverage. For details of using the default transformation, see Section 7.5.

7.20 Compatibility with *cliplab*

The following graphics displays are compatible with the Elekta-Neuromag report composer *cliplab*:

1. The main surface display area in the main window, see Section 7.8.
2. The viewer, see Section 7.10.
3. The sample channel display, see Section 7.7.2.
4. The topographical data display, see Section 7.7.1.
5. The SNR time course display, see Section 7.12.2.
6. The source time course display, see Section 7.13

The graphics can be dragged and dropped from these windows to one of the *cliplab* view areas using the middle mouse button. Because the topographical display area has another function (bed channel selection) tied to the middle mouse button, the graphics is transferred by doing a middle mouse button drag and drop from the label showing the current time underneath the display area itself.

Note: The *cliplab* drag-and-drop functionality requires that you have the proprietary Elekta-Neuromag analysis software installed. *mne_analyze* is compatible with *cliplab* versions 1.2.13 and later.

7.21 Visualizing the head position

When *mne_analyze* is invoked with the `--visualizehpi` option, a simplified user interface shown in Figure 7.31 is displayed. This interface consists only of the viewer window. This *head position visualization* mode can be used with existing data files but is most useful for showing immediate feedback of the head position during experiments with an Elekta-Neuro-mag MEG system.

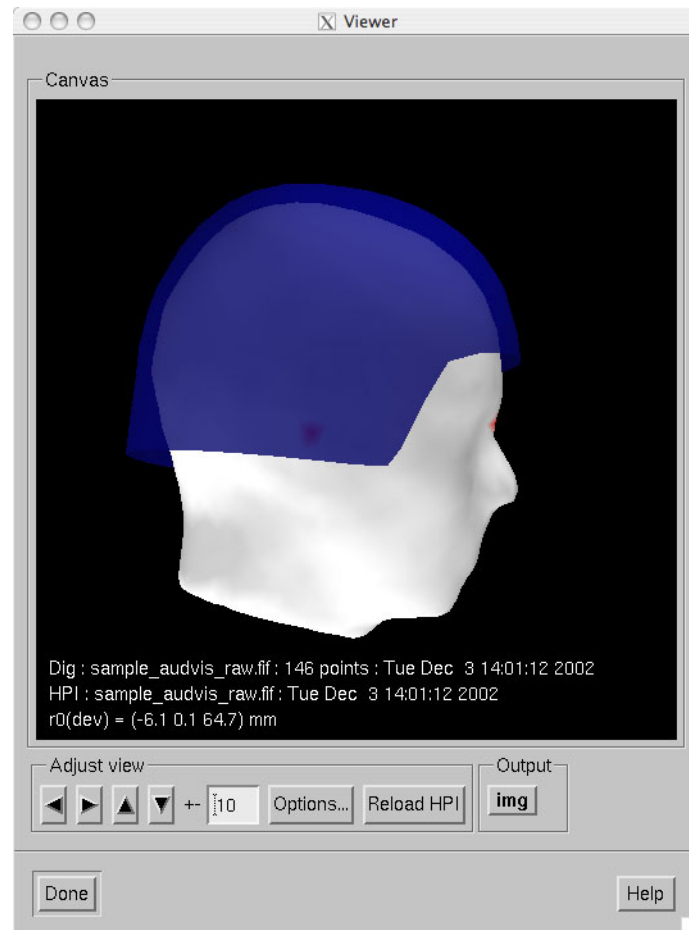


Figure 7.31 Snapshot of *mne_analyze* in the head position visualization mode.

As described in Section 7.2, the head position visualization mode can be customized with the `--dig`, `--hpi`, `--scalehead`, and `--rthelmet` options. For this mode to be useful, the `--dig` and `--hpi` options are mandatory. If existing saved data are viewed, both of these can point to a average or raw data file. For on-line operation with the Elekta-Neuromag systems, the following files in should be used:

```
--dig /neuro/dacq/meas_info/isotrak
--hpi /neuro/dacq/meas_info/hpi_result
```

Note: Since MNE software runs only on LINUX and Mac OS X platforms, one usually needs to NFS mount the volume containing /neuro directory to another system and access these files remotely. However, Neuromag has indicated that future versions of their acquisition software will run on the LINUX platform as well and the complication of remote operation can then be avoided.

When *mne_analyze* starts in the head position visualization mode and the *--dig* and *--hpi* options have been specified, the following sequence operations takes place:

1. The digitizer data, the coordinate transformation between the MEG device and head coordinate frames, as well as the average head surface provided with the MNE software are loaded.
2. If the *--scalehead* option is invoked, the average head surface is scaled to the approximate size of the subject's head by fitting a sphere to the digitizer and to the head surface points lying above the plane of the fiducial landmarks, respectively. The standard head surface is then scaled by the ration of the radiuses of these two best-fitting spheres. Without *--scalehead*, the standard head surface is used as is without scaling.
3. The known positions of (scaled) fiducial landmarks are matched with those available in the digitizer data. This initial alignment is then refined using the ICP algorithm, see Section 7.16. This automatic procedure is found to be accurate enough for the visualization purposes.
4. Using the coordinate transformation thus established between the coordinate system of the scalp surface (MRI coordinates) and the MEG head coordinates together with the coordinate transformation between the MEG head and device coordinate frames established with HPI, the position of the MEG helmet surface is shown in the viewer window.

If the *--rthelmet* option was present, the room-temperature helmet surface is shown instead of the MEG sensor surface. The digitizer and HPI data files are reloaded and the above steps 1. - 4. are repeated when the *Reload HPI* button is pressed. The comment lines in the viewer window show information about the digitizer and HPI data files as well as the location of the MEG device coordinate origin in the MEG head coordinate system.

Tip: The appearance of the viewer visualization can be customized using the *Options...* button, see Section 7.10.2. Since only the scalp and MEG device surfaces are loaded, only a limited number of options is active. The display can also be saved as an image from the *img* button, see Section 7.8.7.

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 $\mathbf{x}_j^{(B)}$ 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.

--prefix <prefix>

Adds *<prefix>* in the beginning of the output label names. A dash will be inserted between *<prefix>* and the rest of the name.

--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.

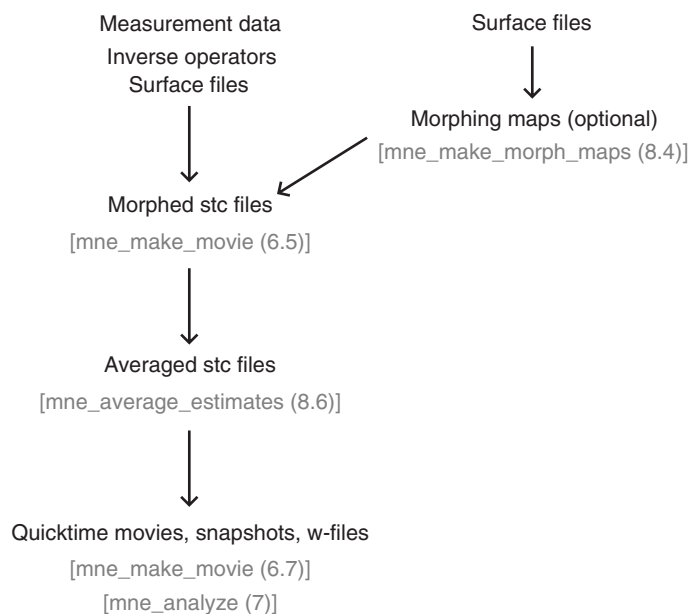


Figure 8.1 Workflow of the cross-subject averaging process in MNE software. References in parenthesis indicate sections and chapters of this manual

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 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 an 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

$$B_{jk} = \sum_{p=1}^P \bar{w}_p [\text{sgn}(S_{jk}^{(p)})]^\alpha |S_{jk}^{(p)}|^{\beta_p}$$

and

$$\bar{w}_p = w_p / \sum_{p=1}^P |w_p|.$$

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: It is straightforward to read and write stc files using the MNE Matlab toolbox described in Chapter 10 and thus write custom Matlab functions to realize more complicated custom group analysis tools.

CHAPTER 9 Data conversion

9.1 Overview

This Chapter describes the data conversion utilities included with the MNE software.

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 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.

Note: In newer CTF data the EEG position information maybe present in the **res4** file. If the **eeg** file is present, the positions given there take precedence over the information in the **res4** 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 end with three lines beginning with *left*, *right*, or *nasion* to indicate the locations of the fiducial landmarks, respectively.

Note: The sequential numbers should be unique within a file. In particular, the numbers 1, 2, and 3 must not appear more than once if the `--numfids` option is used.

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. The format of this text file is described in Section 9.3.1.

--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 compensate 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 re-compensated 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 com-

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/share/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. Therefore, 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 corre-

sponding 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 $\langle cname_k \rangle$ 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 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 5.3. 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.

9.2.8 Importing EEG data saved in the EDF, EDF+, or BDF format

9.2.8.1 Overview

The *mne_edf2fiff* allows conversion of EEG data from EDF, EDF+, and BDF formats to the fiff format. Documentation for these three input formats can be found at:

EDF: <http://www.edfplus.info/specs/edf.html>
EDF+: <http://www.edfplus.info/specs/edfplus.html>
BDF: http://www.biosemi.com/faq/file_format.htm

EDF (European Data Format) and EDF+ are 16-bit formats while BDF is a 24-bit variant of this format used by the EEG systems manufactured by a company called BioSemi.

None of these formats support electrode location information and head shape digitization information. Therefore, this information has to be provided separately. Presently hpts and elp file formats are supported to include digitization data. For information on these formats, see Section 9.3.1 and http://www.sourcesignal.com/formats_probe.html. Note that it is mandatory to have the three fiducial locations (nasion and the two auricular points) included in the digitization data. Using the locations of the fiducial points the digitization data are converted to the MEG head coordinate system employed in the MNE software, see Section 5.3. In the comparison of the channel names only the initial segment up to the first '-' (dash) in the EDF/EDF+/BDF channel name is significant.

The EDF+ files may contain an annotation channel which can be used to store trigger information. The Time-stamped Annotation Lists (TALs) on

the annotation data can be converted to a trigger channel (STI 014) using an annotation map file which associates an annotation label with a number on the trigger channel. The TALs can be listed with the `--listtal` option, see below.

Warning: The data samples in a *BDF* file are represented in a 3-byte (24-bit) format. Since 3-byte raw data buffers are not presently supported in the *fif* format these data will be changed to 4-byte integers in the conversion. Since the maximum size of a *fif* file is 2 GBytes, the maximum size of a *BDF* file to be converted is approximately 1.5 GBytes

Warning: The EDF/EDF+/BDF formats support channel dependent sampling rates. This feature is not supported by *mne_edf2fiff*. However, the annotation channel in the EDF+ format can have a different sampling rate. The annotation channel data is not included in the *fif* files output.

9.2.8.2 Using *mne_edf2fiff*

The command-line options of *mne_edf2fiff* are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--edf <filename>

Specifies the name of the raw data file to process.

--tal <filename>

List the time-stamped annotation list (TAL) data from an EDF+ file here. This output is useful to assist in creating the annotation map file, see the `--annotmap` option, below. This output file is an event file compatible with *mne_browse_raw* and *mne_process_raw*, see Chapter 4. In addition, in the mapping between TAL labels and trigger numbers provided by the `--annotmap` option is employed to assign trigger numbers in the event file produced. In the absense of the `--annotmap` option default trigger number 1024 is used.

--annotmap <filename>

Specify a file which maps the labels of the TALs to numbers on a trigger channel (STI 014) which will be added to the output file if this option is present. This *annotation map* file may contain comment lines starting with the ‘%’ or ‘#’ characters. The data lines contain a label-number pair, separated by a colon. For example, a

line ‘Trigger-1:9’ means that each annotation labeled with the text ‘Trigger-1’ will be translated to the number 9 on the trigger channel.

--elp <filename>

Specifies the name of the an electrode location file. This file is in the “probe” file format used by the *Source Signal Imaging, Inc.* software. For description of the format, see http://www.sourcesignal.com/formats_probe.html. Note that some other software packages may produce electrode-position files with the *elp* ending not conforming to the above specification. As discussed above, the fiducial marker locations, optional in the “probe” file format specification are mandatory for *mne_edf2fiff*. When this option is encountered on the command line any previously specified *hpts* file will be ignored.

--hpts <filename>

Specifies the name of an electrode position file in the *hpts* format discussed in Section 9.3.1. The mandatory entries are the fiducial marker locations and the EEG electrode locations. It is recommended that electrode (channel) names instead of numbers are used to label the EEG electrode locations. When this option is encountered on the command line any previously specified *elp* file will be ignored.

--meters

Assumes that the digitization data in an *hpts* file is given in meters instead of millimeters.

--fif <filename>

Specifies the name of the fif file to be output.

9.2.8.3 Post-conversion tasks

This section outlines additional steps to be taken to use the EDF/EDF+/BDF file is converted to the fif format in MNE:

1. Some of the channels may not have a digitized electrode location associated with them. If these channels are used for EOG or EMG measurements, their channel types should be changed to the correct ones using the *mne_rename_channels* utility, see Section 11.4.5. EEG channels which do not have a location associated with them should be assigned to be MISC channels.
2. After the channel types are correctly defined, a topographical layout file can be created for *mne_browse_raw* and *mne_analyze* using the *mne_make_eeg_layout* utility, see Section 11.6.
3. The trigger channel name in BDF files is “Status”. This must be specified with the `--digtrig` option or with help of the

- MNE_TRIGGER_CH_NAME environment variable when *mne_browse_raw* or *mne_process_raw* is invoked, see Section 4.2.1.
- Only the two least significant bytes on the “Status” channel of BDF files are significant as trigger information the `--digtrigmask 0xff` option MNE_TRIGGER_CH_MASK environment variable should be used to specify this to *mne_browse_raw* and *mne_process_raw*, see Section 4.2.1.

9.2.9 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:

- The raw data file containing the acquired EEG data. The name of this file ends with the suffix `.raw`.
- 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`.
- 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. This file is in the “probe” file format used by the *Source Signal Imaging, Inc.* software. For description of the format, see http://www.sourcesignal.com/formats_probe.html. The fiducial marker locations, optional in the “probe” file format specification are mandatory for *mne_tufts2fiff*. Note that some other software packages may produce electrode-position files with the *elp* ending not conforming to the above specification.

Note: The conversion process includes a transformation from the Tufts head coordinate system convention to that used in the Neuromag systems.

Important: The fiducial landmark locations, optional in the probe file format, must be present for *mne_tufts2fiff*.

9.2.10 Importing BrainVision EEG data

The utility *mne_brain_vision2fiff* was created to import BrainVision EEG data. This utility also helps to import the eXimia (Nexstim) TMS-compatible EEG system data to the MNE software. The utility uses an optional fif file containing the head digitization data to allow source modeling. The MNE Matlab toolbox contains the function *fiff_write_dig_file* to write a digitization file based on digitization data available in another format, see Chapter 10.

The command-line options of *mne_brain_vision2fiff* are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--header <name>

The name of the BrainVision header file. The extension of this file is *vhdr*. The header file typically refers to a marker file (*vmrk*) which is automatically processed and a digital trigger channel (STI 014) is formed from the marker information. The *vmrk* file is ignored if the **--eximia** option is present.

--dig <name>

The name of the fif file containing the digitization data.

--orignames

Use the original EEG channel labels. If this option is absent the EEG channels will be automatically renamed to EEG 001, EEG 002, *etc.*

--eximia

Interpret this as an eXimia data file. The first three channels will be thresholded and interpreted as trigger channels. The composite digital trigger channel will be composed in the same way as in the *mne_kit2fiff* utility, see Section 9.2.7, above. In addition, the fourth channel will be assigned as an EOG channel. This option is normally used by the *mne_eximia2fiff* script, see Section 9.2.11.

--split <size/MB>

Split the output data into several files which are no more than <size> MB. By default, the output is split into files which are just below 2 GB so that the fif file maximum size is not exceeded.

--out <filename>

Specifies the name of the output fif format data file. If <filename> ends with .fif or _raw.fif, these endings are deleted. After these modifications, _raw.fif is inserted after the remaining part of the file name. If the file is split into multiple parts, the additional parts will be called <name>--<number>_raw.fif.

9.2.11 Converting eXimia EEG data

EEG data from the Nexstim eXimia system can be converted to the fif format with help of the *mne_eximia2fiff* script. It creates a BrainVision vhdr file and calls *mne_brain_vision2fiff*. Usage:

```
mne_eximia2fiff [--dig dfile] [--orignames] file1 file2 ...
```

where *file1 file2...* are eXimia *nxe* files and the *--orignames* option is passed on to *mne_brain_vision2fiff*. If you want to convert all data files in a directory, say

```
mne_eximia2fiff *.nxe
```

The optional file specified with the *--dig* option is assumed to contain digitizer data from the recording in the Nexstim format. The resulting fif data file will contain these data converted to the fif format as well as the coordinate transformation between the eXimia digitizer and MNE head coordinate systems.

Note: This script converts raw data files only.

9.3 Converting digitization data

The *mne_convert_dig_data* utility converts Polhemus digitization data between different file formats. The input formats are:

fif	The standard format used in MNE. The digitization data are typically present in the measurement files.
hpts	A text format which is a translation of the fif format data, see Section 9.3.1 below.
elp	A text format produced by the <i>Source Signal Imaging, Inc.</i> software. For description of this “probe” format, see http://www.sourcesignal.com/formats_probe.html .

The data can be output in *fif* and *hpts* formats. Only the last command-line option specifying an input file will be honored. Zero or more output file options can be present on the command line.

Important: The *elp* and *hpts* input files may contain textual EEG electrode labels. They will not be copied to the *fif* format output.

The command-line options of *mne_convert_dig_data* are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--fif <name>

Specifies the name of an input *fif* file.

--hpts <name>

Specifies the name of an input *hpts* file.

--elp <name>

Specifies the name of an input *elp* file.

--fifout <name>

Specifies the name of an output *fif* file.

--hptsout <name>

Specifies the name of an output *hpts* file.

--headcoord

The *fif* and *hpts* input files are assumed to contain data in the MNE head coordinate system, see Section 5.3. With this option present, the data are transformed to the MNE head coordinate system with help of the fiducial locations in the data. Use this option if this is not the case or if you are unsure about the definition of the coordinate system of the *fif* and *hpts* input data. This option is implied with *elp* input files. If this option is present, the *fif* format output file will contain the transformation between the original digitizer data coordinates the MNE head coordinate system.

9.3.1 The hpts format

The *hpts* format digitizer data file may contain comment lines starting with the pound sign (#) and data lines of the form:

<category> *<identifier>* *<x/mm>* *<y/mm>* *<z/mm>*

where

<category>

defines the type of points. Allowed categories are: *hpi*, *cardinal* (*fiducial*), *eeg*, and *extra* corresponding to head-position indicator coil locations, cardinal landmarks, EEG electrode locations, and additional head surface points, respectively. Note that *tkmedit* does not recognize the *fiducial* as an alias for *cardinal*.

<identifier>

identifies the point. The identifiers are usually sequential numbers. For cardinal landmarks, 1 = left auricular point, 2 = nasion, and 3 = right auricular point. For EEG electrodes, identifier = 0 signifies the reference electrode. Some programs (not *tkmedit*) accept electrode labels as identifiers in the *eeg* category.

<x/mm>, <y/mm>, <z/mm>

Location of the point, usually in the MEG head coordinate system, see Section 5.3. Some programs have options to accept coordinates in meters instead of millimeters. With *--meters* option, *mne_transform_points* lists the coordinates in meters.

9.4 Converting volumetric data into an MRI overlay

With help of the *mne_volume_source_space* utility (Section 5.5) it is possible to create a source space which is defined within a volume rather than a surface. If the *--mri* option was used in *mne_volume_source_space*, the source space file contains an interpolator matrix which performs a tri-linear interpolation into the voxel space of the MRI volume specified.

At present, the MNE software does not include facilities to compute volumetric source estimates. However, it is possible to calculate forward solutions in the volumetric grid and use the MNE Matlab toolbox to read the forward solution. It is then possible to compute, *e.g.*, volumetric beamformer solutions in Matlab and output the results into *w* or *stc* files. The purpose of the *mne_volume_data2mri* is to produce MRI overlay data compatible with FreeSurfer MRI viewers (in the *mgh* or *mgz* formats) from this type of *w* or *stc* files.

mne_volume_data2mri accepts the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--src <filename>

The name of the volumetric source space file created with *mne_volume_source_space*. The source space must have been cre-

ated with the `--mri` option, which adds the appropriate sparse tri-linear interpolator matrix to the source space.

--w <filename>

The name of a *w* file to convert into an MRI overlay.

--stc <filename>

The name of the *stc* file to convert into an MRI overlay. If this file has many time frames, the output file may be huge. Note: If both `-w` and `--stc` are specified, `-w` takes precedence.

--scale <number>

Multiply the *stc* or *w* by this scaling constant before producing the overlay.

--out <filename>

Specifies the name of the output MRI overlay file. The name must end with either `.mgh` or `.mgz` identifying the uncompressed and compressed FreeSurfer MRI formats, respectively.

9.5 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.

9.6 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.

--id <number>

Identify the surface to be listed. The surfaces are numbered starting with the innermost surface. Thus, for a three-layer model the surface numbers are: 4 = scalp, 3 = outer skull, 1 = inner skull
Default value is 4.

--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.

9.7 Converting surface data between different formats

The utility *mne_convert_surface* converts surface data files between different formats.

Important: The MNE Matlab toolbox functions enable reading of FreeSurfer surface files directly. Therefore, the `--mat` option has been removed. The *dfs* file format conversion functionality has been moved here from *mne_convert_dfs*. Consequently, *mne_convert_dfs* has been removed from MNE software.

9.7.1 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.

--fif <name>

Specifies a fif format input file. The first surface (source space) from this file will be read.

--tri <name>

Specifies a text format input file. The format of this file is described in Section 5.6.3.

--meters

The unit of measure for the vertex locations in a text input files is meters instead of the default millimeters. This option does not have any effect on the interpretation of the *FreeSurfer* surface files specified with the **--surf** option.

--swap

Swap the ordering of the triangle vertices. The standard convention in the MNE software is to have the vertices in text format files 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. This is also called the counterclockwise ordering. If your text input file does not comply with this right-hand rule, use the **--swap** option. This option does not have any effect on the interpretation of the *FreeSurfer* surface files specified with the **--surf** option.

--surf <name>

Specifies a *FreeSurfer* format input file.

--dfs <name>

Specifies the name of a dfs file to be converted. The surfaces produced by BrainSuite are in the dfs format.

--mghmri <name>

Specifies a mgh/mgz format MRI data file which will be used to define the coordinate transformation to be applied to the data read from a dfs file to bring it to the *FreeSurfer* MRI coordinates, *i.e.*, the coordinate system of the MRI stack in the file. In addition, this option can be used to insert “volume geometry” information to the *FreeSurfer* surface file output (**--surfout** option). If the input file already contains the volume geometry information, **--replacegeom** is needed to override the input volume geometry and to proceed to writing the data.

--replacegeom

Replaces existing volume geometry information. Used in conjunction with the **--mghmri** option described above.

--fifmri <name>

Specifies a fif format MRI description file which will be used to define the coordinate transformation to be applied to the data read

from a dfs file to bring it to the same coordinate system as the MRI stack in the file.

--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 MRI coordinates, see below. This option is rarely needed.

--flip

By default, the dfs surface nodes are assumed to be in a right-anterior-superior (RAS) coordinate system with its origin at the left-posterior-inferior (LPI) corner of the MRI stack. Sometimes the dfs file has left and right flipped. This option reverses this flip, *i.e.*, assumes the surface coordinate system is left-anterior-superior (LAS) with its origin in the right-posterior-inferior (RPI) corner of the MRI stack.

--shift <value/mm>

Shift the surface vertices to the direction of the surface normals by this amount before saving the surface.

--surfout <name>

Specifies a FreeSurfer format output file.

--fifout <name>

Specifies a fif format output file.

--triout <name>

Specifies an ASCII output file that will contain the surface data in the triangle file format described in Section 5.6.3.

--pntout <name>

Specifies a ASCII output file which will contain the vertex numbers only.

--metersout

With this option the ASCII output will list the vertex coordinates in meters instead of millimeters.

--swapout

Defines the vertex ordering of ASCII triangle files to be output. For details, see **--swap** option, above.

--smfout <name>

Specifies a smf (Simple Model Format) output file. For details of this format, see <http://people.scs.fsu.edu/~burkardt/data/smf.txt>.

Note: Multiple output options can be specified to produce outputs in several different formats with a single invocation of *mne_convert_surface*.

The coordinate transformation file specified with the `--trans` should contain a 4 x 4 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 $\mathbf{r}_{\text{dfs}} = \begin{bmatrix} x_{\text{dfs}} & y_{\text{dfs}} & z_{\text{dfs}} & 1 \end{bmatrix}^T$ and $\mathbf{r}_{\text{MRI}} = \begin{bmatrix} x_{\text{MRI}} & y_{\text{MRI}} & z_{\text{MRI}} & 1 \end{bmatrix}^T$, respectively,

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

9.8 Converting MRI data into the fif format

The utility *mne_make_cor_set* creates a *fif* format MRI description file optionally including the MRI data using FreeSurfer MRI volume data as input. The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--dir <directory>

Specifies a directory containing the MRI volume in COR format. Any previous `--mgh` options are cancelled when this option is encountered.

--withdata

Include the pixel data to the output file. This option is implied with the `--mgh` option.

--mgh <name>

An MRI volume file in *mgh* or *mgz* format. The `--with-data` option is implied with this type of input. Furthermore, the \mathbf{T}_3 transformation, the Talairach transformation \mathbf{T}_4 from the talairach.xfm file referred to in the MRI volume, and the the fixed transforms \mathbf{T}_- and \mathbf{T}_+ will added to the output file. For definition of the coordinate transformations, see Section 5.2.

--talairach <name>

Take the Talairach transform from this file instead of the one specified in *mgh/mgz* files.

--out <name>

Specifies the output file, which is a fif-format MRI description file.

9.9 Collecting coordinate transformations into one file

The utility *mne_collect_transforms* collects coordinate transform information from various sources and saves them into a single fif file. The coordinate transformations used by MNE software are summarized in Figure 5.1. The output of *mne_collect_transforms* may include all transforms referred to therein except for the sensor coordinate system transformations $\mathbf{T}_{s_1} \dots \mathbf{T}_{s_n}$. The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--meas <name>

Specifies a measurement data file which provides \mathbf{T}_1 . A forward solution or an inverse operator file can also be specified as implied by Table 5.1.

--mri <name>

Specifies an MRI description or a standalone coordinate transformation file produced by *mne_analyze* which provides \mathbf{T}_2 . If the **--mgh** option is not present *mne_collect_transforms* also tries to find \mathbf{T}_3 , \mathbf{T}_4 , \mathbf{T}_- , and \mathbf{T}_+ from this file.

--mgh <name>

An MRI volume file in *mgh* or *mgz* format. This file provides \mathbf{T}_3 . The transformation \mathbf{T}_4 will be read from the *talairach.xfm* file referred to in the MRI volume. The fixed transforms \mathbf{T}_- and \mathbf{T}_+ will be also created.

--out <name>

Specifies the output file. If this option is not present, the collected transformations will be output on screen but not saved.

9.10 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*.

9.11 Converting a lspcov 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

9.12 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.

Important: 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 additional files occupying disk space.

9.12.1 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.

9.12.2 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>
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.

9.12.3 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.
<i>ntime</i>	Number of time points in the measurement data.

Structure	Contents
<i><tag></i> _meas	Measured data
<i><tag></i> _inv	The inverse operator decomposition
<i><tag></i> _fwd	The forward solution
<i><tag></i> _noise	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 9.12.1. 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 5.2 and 5.3. For EEG electrodes, this value equals one.

Table 9.3 The fields of Matlab structures.

Variable	Size	Description
ch_pos	nchan x 3	The location information for each channel. The first three values specify the origin of the sensor coordinate system or the location of the electrode. For MEG channels, the following nine number specify the x , y , and z -direction unit vectors of the sensor 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
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 the <i>coord_frame</i> member.
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 <i>mne_make_source_space</i> was invoked. <i>mne_setup_source_space</i> lists the left hemisphere first.

Table 9.3 The fields of Matlab structures.

Variable	Size	Description
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 to 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.
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_eigenvec	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.

9.13 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 informa-

tion 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. Therefore, it is recommended that you use the Matlab toolbox rather than *mne_raw2mat* which creates large files occupying disk space unnecessarily.

9.13.1 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>_.

9.13.2 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 stucture 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 9.4. Further explanation of the *bufs* field is provided in Table 9.5.

Variable	Size	Description
orig_file	string	The name of the original fif file specified with the <code>--raw</code> option.
nchan	1	Number of channels.
nsamp	1	Total number of samples
bufs	nbuf x 4	This field is present if <code>--info</code> option was specified on the command line. For details, see Table 9.5.
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 5.2 and 5.3. 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 sensor 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 sensor 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.

Table 9.4 The fields of the raw data info structure.

Variable	Size	Description
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 <i>fiff</i> 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 9.4 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 <i>fif</i> 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 <i>fif</i> 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 9.5 The *bufs* member of the raw data info structure.

9.14 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.

Important: The MNE Matlab toolbox described in Chapter 10 provides direct access to raw *fif* files without conversion with *mne_epochs2mat* first. Therefore, it is recommended that you use the Matlab toolbox rather than *mne_epochs2mat* which creates large files occupying disk space unnecessarily. An exception to this is the case where you apply a filter to the data and save the band-pass filtered epochs.

9.14.1 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 4.10.5. 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 composite digital trigger channel. The default value is 'STI 014'. Underscores in the channel name will be replaced by spaces.

--digtrigmask <number>

Mask to be applied to the trigger channel values before considering them. This option is useful if one wants to set some bits in a don't care state. For example, some finger response pads keep the trigger lines high if not in use, *i.e.*, a finger is not in place. Yet, it is convenient to keep these devices permanently connected to the acquisition system. The number can be given in decimal or hexadecimal format (beginning with 0x or 0X). For example, the value 255 (0xFF) means that only the lowest order byte (usually trigger lines 1 - 8 or bits 0 - 7) will be considered.

--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.

Note: Strictly speaking, trigger mask value zero would mean that all trigger inputs are ignored. However, for convenience, setting the mask to zero or not setting it at all has the same effect as 0xFFFFFFFF, *i.e.*, all bits set.

Tip: The digital trigger channel can also be set with the MNE_TRIGGER_CH_NAME environment variable. Underscores in the variable value will *not* be replaced with spaces by *mne_browse_raw* or *mne_process_raw*. Using the **--digtrig** option supersedes the MNE_TRIGGER_CH_NAME environment variable.

Tip: The digital trigger channel mask can also be set with the MNE_TRIGGER_CH_MASK environment variable. Using the **--digtrigmask** option supersedes the MNE_TRIGGER_CH_MASK environment variable.

9.14.2 The binary epoch data file

mne_epochs2mat saves the epoch data extracted from the raw data file in 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 not calibrated, *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.

9.14.3 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 9.4. Further explanation of the *epochs* member is provided in Table 9.7.

Variable	Size	Description
orig_file	string	The name of the original fif file specified with the <code>--raw</code> option.
epoch_file	string	The name of the epoch data file produced by <i>mne_epochs2mat</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 9.7.
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 5.2 and 5.3. 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.

Table 9.6 The fields of the raw data info structure.

Variable	Size	Description
ch_pos	nchan x 12	The location information for each channel. The first three values specify the origin of the sensor 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 sensor 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> are not calibrated. The first column is the <i>range</i> member of the fiff 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 9.6 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 9.7 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 9.6.

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/share/matlab`. The names of the MNE Matlab toolbox functions begin either with *mne_* or with *fiff_*. When you source the *mne_setup* script as described in Section 2.4 on of the following actions takes place:

1. If you do not have the Matlab startup.m file, it will be created and lines allowing access to the MNE Matlab toolbox are added.
2. If you have startup.m and it does not have the standard MNE Matlab toolbox setup lines, you will be instructed to add them manually.
3. If you have startup.m and the standard MNE Matlab toolbox setup lines are there, nothing happens.

A summary of the available routines is provided in Tables 10.1 – 10.14. The toolbox also contains a set of examples which may be useful starting points for your own development. The names of these functions startwith *mne_ex* and they are listed in Table 10.14.

Important: The MNE Matlab Toolbox is compatible with Matlab versions 7.0 or later.

Important: The matlab function *fiff_setup_read_raw* has a significant change. The sample numbers now take into account possible initial skip in the file, *i.e.*, the time between the start of the data acquisition and the start of saving the data to disk. The *first_samp* member of the returned structure indicates the initial skip in samples. If you want your own routines, which assume that initial skip has been removed, perform identically with the previous version, subtract *first_samp* from the sample numbers you specify to *fiff_read_raw_segment*. Furthermore, *fiff_setup_read_raw* has

an optional argument to allow reading of unprocessed MaxShield data acquired with the Elekta MEG systems.

Function	Purpose
fiff_find_evoked	Find all evoked data sets from a file.
fiff_read_bad_channels	Read the bad channel list.
fiff_read_ctf_comp	Read CTF software gradient compensation data.
fiff_read_evoked	Read evoked-response data.
fiff_read_evoked_all	Read all evoked-response data from a file.
fiff_read_meas_info	Read measurement information.
fiff_read_mri	Read an MRI description file.
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	Apply a coordinate transformation to the sensor location data 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.

Table 10.5 Writing routines.

Function	Purpose
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.
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_int_matrix	Write a matrix of 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
fiff_write_evoked	Write an evoked-response data file.
fiff_finish_writing_raw	Write the closing tags to a raw data file.
fiff_start_writing_raw	Start writing raw data file, <i>i.e.</i> , write the measurement information.
fiff_write_dig_file	Write a fif file containing digitization data.
fiff_write_raw_buffer	Write one raw data buffer. This is used after a call to fiff_start_writing_raw.

Table 10.6 High-level data writing routines.

Function	Purpose
<code>mne_add_coil_defs</code>	Add coil definitions to an array of channel information structures.
<code>mne_load_coil_def</code>	Load a coil definition file.

Table 10.7 Coil definition utilities.

Function	Purpose
<code>mne_compensate_to</code>	Apply or remove CTF software gradient compensation from evoked-response data.
<code>mne_get_current_comp</code>	Get the state of software gradient compensation from measurement info.
<code>mne_make_compensator</code>	Make a compensation matrix which switches the status of CTF software gradient compensation from one state to another.
<code>mne_make_projector</code>	Create a signal-space projection operator with the projection item definitions and cell arrays of channel names and bad channel names as input.
<code>mne_make_projector_info</code>	Like <code>mne_make_projector</code> but uses the measurement info structure as input.
<code>mne_set_current_comp</code>	Change the information about the compensation status in measurement info.

Table 10.8 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_pick_channels_forward</code>	Pick desired channels (rows) from a forward solution.
<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_events</code>	Read an event list from a fif file produced by <i>mne_browse_raw</i> or <i>mne_process_raw</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.

Table 10.9 High-level routines for reading MNE data files.

Function	Purpose
<code>mne_read_morph_map</code>	Read an morphing map produced with <i>mne_make_morph_maps</i> , see Section 8.4.
<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 10.9 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_events</code>	Write a fif format event file compatible with <i>mne_browse_raw</i> and <i>mne_process_raw</i> .
<code>mne_write_inverse_sol_stc</code>	Write stc files containing an inverse solution or other dynamic data on the cortical surface.
<code>mne_write_inverse_sol_w</code>	Write w files containing an inverse solution or other static data on the cortical surface

Table 10.10 High-level routines for writing MNE data files.

Function	Purpose
<code>mne_read_stc_file</code>	Read data from one stc file. The vertex numbering in the returned structure will start from 0.
<code>mne_read_stc_file1</code>	Read data from one stc file. The vertex numbering in the returned structure will start from 1.
<code>mne_read_w_file</code>	Read data from one w file. The vertex numbering in the returned structure will start from 0.
<code>mne_read_w_file1</code>	Read data from one w file. The vertex numbering in the returned structure will start from 1.
<code>mne_write_stc_file</code>	Write a new stc file. It is assumed the the vertex numbering in the input data structure containing the stc information starts from 0.

Table 10.11 Routines related to stc, w, and label files.

Function	Purpose
mne_write_stc_file1	Write a new stc file. It is assumed the the vertex numbering in the input data structure containing the stc information starts from 1.
mne_write_w_file	Write a new w file. It is assumed the the vertex numbering in the input data structure containing the w file information starts from 0.
mne_write_w_file1	Write a new w file. It is assumed the the vertex numbering in the input data structure containing the w file information starts from 1.
mne_read_label_file	Read a label file (ROI)
mne_write_label_file	Write a label file (ROI)
mne_label_time_courses	Extract time courses corresponding to a label from an stc file.

Table 10.11 Routines related to stc, w, and label files.

Function	Purpose
mne_read_curvature	Read a curvature file.
mne_read_surface	Read one surface, return the vertex locations and triangulation info.
mne_read_surfaces	Read surfaces corresponding to one or both hemispheres. Optionally read curvature information and add derived surface data.
mne_reduce_surface	Reduce the number of triangles on a surface using the reducepatch Matlab function.
mne_write_surface	Write a FreeSurfer surface file.

Table 10.12 Routines for reading FreeSurfer surfaces.

Function	Purpose
mne_block_diag	Create a sparse block-diagonal matrix out of a vector.
mne_combine_xyz	Calculate the square sum of the three Cartesian components of several vectors listed in one row or column vector.

Table 10.13 Utility functions.

Function	Purpose
<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_make_combined_event_file</code>	Combine data from several trigger channels into one event file.
<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_transform_coordinates</code>	Transform locations between different coordinate systems. This function uses the output file from <i>mne_collect_transforms</i> described in Section 9.9 as input.
<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 10.13 Utility functions.

Function	Purpose
<code>mne_ex_average_epochs</code>	Example of averaging epoch data produced by <i>mne_epochs2mat</i> , see Section 9.14.
<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.
<code>mne_ex_evoked_grad_amp</code>	Compute tangential gradient amplitudes from planar gradiometer data.
<code>mne_ex_read_epochs</code>	Read epoch data from a raw data file.
<code>mne_ex_read_evoked</code>	Example of reading evoked-response data.

Table 10.14 Examples demonstrating the use of the toolbox.

Function	Purpose
<code>mne_ex_read_raw</code>	Example of reading raw data.
<code>mne_ex_read_write_raw</code>	Example of processing raw data (read and write).

Table 10.14 Examples demonstrating the use of the toolbox.

Important: In order for the inverse operator calculation to work correctly with data processed with the Elekta-Neuromag Maxfilter™ software, the so-called *processing history* block must be included in data files. Previous versions of the MNE Matlab functions did not copy processing history to files saved. As of March 30, 2009, the Matlab toolbox routines *fiff_start_writing_raw* and *fiff_write_evoked* have been enhanced to include these data to the output file as appropriate. If you have older raw data files created in Matlab from input which has been processed Maxfilter, it is necessary to copy the *processing history* block from the original to modified raw data file using the *mne_copy_processing_history* utility described in Section 11.4.8. The raw data processing programs *mne_browse_raw* and *mne_process_raw* have handled copying of the processing history since revision 2.5 of the MNE software.

10.2 Some data structures

The MNE Matlab toolbox relies heavily on structures to organize the data. This section gives detailed information about fields in the essential data structures employed in the MNE Matlab toolbox. In the structure definitions, data types referring to other MNE Matlab toolbox structures are shown in italics. In addition, Table 10.15 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 10.16.

taginfo

Contains the information about one tag, see Table 10.17.

directory

Contains the tag directory as a tree structure, see Table 10.18.

id

A fif ID, see Table 10.19.

named matrix

Contains a matrix with names for rows and/or columns, see Table 10.20. 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 10.21.

dig

A Polhemus digitizer data point, see Table 10.22.

coildef

The coil definition structure useful for forward calculations and array visualization, see Table 10.23. For more detailed information on coil definitions, see Section 5.8.

ch

Channel information structure, see Table 10.24.

proj

Signal-space projection data, see Table 10.25.

comp

Software gradiometer compensation data, see Table 10.26.

measurement info

Translation of the FIFFB_MEAS_INFO entity, see Table 10.27. This data structure is returned by *fiff_read_meas_info*.

surf

Used to represent triangulated surfaces and cortical source spaces, see Table 10.28.

cov

Used for storing covariance matrices, see Table 10.29.

fwd

Forward solution data returned by *mne_read_forward_solution*, see Table 10.30.

inv

Inverse operator decomposition data returned by *mne_read_inverse_operator*, see Table 10.31. 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*.

Important: The MNE Matlab toolbox tries it best to employ vertex numbering starting from 1 as opposed to 0 as recorded in the data files. There are, however, two exceptions where explicit attention to the vertex numbering convention is needed. First, the standard stc and w file reading and writing routines return and assume zero-based vertex numbering. There are now versions with names ending with '1', which return and assume

one-based vertex numbering, see Table 10.11. Second, the `logno` field of the channel information in the data files produced by *mne_compute_raw_inverse* is the zero-based number of the vertex whose source space signal is contained on this channel.

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.
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_DEVICE	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).

Table 10.15 FIFF constants.

Name	Value	Description
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.
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.

Table 10.15 FIFF constants.

Name	Value	Description
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
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_MNE_DEPTH_PRIOR_COV	5	Indicates the depth prior (depth weighting) covariance matrix
FIFFV_MNE_ORIENT_PRIOR_COV	6	Indicates the orientation (loose orientation constrain) prior 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_HOMOGRAD_FIELD	5	This projection item contains the three homogeneous field components as seen by the sensor array.

Table 10.15 FIFF constants.

Name	Value	Description
FIFFV_MNE_PROJ_ITEM_EEG_AVREF	10	This projection item corresponds to the average EEG reference.

Table 10.15 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 10.16 The *tag* structure.

Field	Data type	Description
kind	double	The kind of the data item.
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 10.17 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 10.18 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 past <i>secs</i>).

Table 10.19 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.
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 10.20 The *named matrix* structure.

Field	Data type	Description
from	int32	The source coordinate frame, see Table 10.15. 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 10.21 The *trans* structure.

Field	Data type	Description
kind	int32	The type of digitizing point. Possible values are listed in Table 10.15. Look for entries starting with FIFF_POINT.
ident	int32	Identifier for this point.
r	single(3)	The location of this point.

Table 10.22 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
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 10.23 The *coildef* structure. For more detailed information, see Section 5.8.

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 channel type (FIFFV_MEG_CH, FIFF_EEG_CH, etc., see Table 10.15).
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.

Table 10.24 The *ch* structure.

Field	Data type	Description
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 <i>fif</i> 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 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.
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 10.24 The *ch* structure.

Field	Data type	Description
kind	int32	The type of the projection item. Possible values are listed in Table 10.15. 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 10.25 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.
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 10.26 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.

Table 10.27 The *meas info* structure.

Field	Data type	Description
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	<i>cell(*)</i>	Bad channel list
projs	<i>proj(*)</i>	SSP operator data.
comps	<i>comp(*)</i>	Software gradient compensation data.

Table 10.27 The *meas info* structure.

Field	Data type	Description
id	int32	The surface ID
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.

Table 10.28 The *surf* structure.

Field	Data type	Description
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.
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.
dist	double(np,np)	Distances between vertices on this surface given as a sparse matrix. A zero off-diagonal entry in this matrix indicates that the corresponding distance has not been calculated.
dist_limit	double	The value given to <code>mne_add_patch_info</code> with the <code>--dist</code> option, see Section 11.7.2. This value is presently always negative, indicating that only distances between active source space vertices, as indicated by the <code>vertno</code> field of this structure, have been calculated.

Table 10.28 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.

Table 10.29 The *cov* structure.

Field	Data type	Description
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 10.29 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.
sol_grad	<i>named matrix</i>	The derivatives of the forward solution with respect to the dipole location coordinates, see Section 5.9.6. This field is present only if the forward solution was computed with the <code>--grad</code> option, see Section 5.9.2.
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 10.30 The *fwd* structure.

Field	Data type	Description
methods	int32	Has the solution been computed using MEG data (1), EEG data (2), or both (3).

Table 10.31 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
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.
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 10.31 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 – 10.14 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.

--blocks

Only list the blocks (the tree structure) of the file. The tags within each block are not listed.

--indent <number>

Number of spaces for indentation for each deeper level in the tree structure of the fif files. The default indentation is 3 spaces in terse and no spaces in verbose listing mode.

--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/share/mne/fiff_explanations.txt`.

11.4 Data file modification 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.

In newer Neuromag Vectorview systems with 16-bit digital inputs the upper two bytes of the samples may be incorrectly set when stimulus input 16 is used and the data are acquired in the 32-bit mode. This problem can be fixed by running *mne_fix_stim14* on a raw data file with the `--32` option:

```
mne_fix_stim14 --32 <raw data file>
```

In this case, the correction will be applied to the stimulus channels 'STI101' and 'STI201'.

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 5.2 and 5.3, 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 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 or the `MNE_TRIGGER_CH_NAME` environment variable every time a 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 text file which contains the modifications to be applied, see below.

--revert

Reverse the roles of old and new channel names in the alias file.

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.

Tip: You might consider renaming the EEG channels with descriptive labels related to the standard 10-20 system. This allows you to use standard EEG channel names when defining derivations, see Sections 11.5

and 4.4.12, as well as in the channel selection files used in *mne_browse_raw*, see Section 4.5.5.

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.

11.4.7 Removing identifying information

Depending on the settings during acquisition in the Elekta-Neuromag EEG/MEG systems the data files may contain subject identifying information in unencrypted form. The utility *mne_anonymize* was written to clear tags containing such information from a *fif* file. Specifically, this utility removes the following tags from the *fif* file:

Tag	Description
FIFF_SUBJ_FIRST_NAME	First name of the subject
FIFF_SUBJ_MIDDLE_NAME	Middle name of the subject
FIFF_SUBJ_LAST_NAME	Last name of the subject
FIFF_SUBJ_BIRTH_DAY	Birthday of the subject (Julian day number)
FIFF_SUBJ_SEX	The sex of the subject
FIFF_SUBJ_HAND	Handedness of the subject
FIFF_SUBJ_WEIGHT	Weight of the subject in kg
FIFF_SUBJ_HEIGHT	Height of the subject in m
FIFF_SUBJ_COMMENT	Comment about the subject

Table 11.2 Tags cleared by *mne_anonymize*.

Important: *mne_anonymize* normally keeps the FIFF_SUBJ_HIS_ID tag which can be used to identify the subjects uniquely after the information listed in Table 11.2 have been removed. If the `--his` option is specified on the command line, the FIFF_SUBJ_HIS_ID tag will be removed as well. The data of the tags listed in Table 11.2 and the optional FIFF_SUBJ_HIS_ID tag are overwritten with zeros and the space claimed by omitting these tags is added to the free space list of the file. Therefore, after *mne_anonymize* has processed a data file there is no way to recover the removed information. Use this utility with caution.

mne_anonymize recognizes the following command-line options:

--version

Show the program version and compilation date.

--help

List the command-line options.

--his

Remove the FIFF_SUBJ_HIS_ID tag as well, see above.

--file <name>

Specifies the name of the file to be modified.

Note: You need write permission to the file to be processed.

11.4.8 Copying the processing history

In order for the inverse operator calculation to work correctly with data processed with the Elekta-Neuromag Maxfilter™ software, the so-called *processing history* block must be included in data files. Previous versions of the MNE Matlab functions did not copy processing history to files saved. As of March 30, 2009, the Matlab toolbox routines *fiff_start_writing_raw* and *fiff_write_evoked* have been enhanced to include these data to the output file as appropriate. If you have older raw data files created in Matlab from input which has been processed Maxfilter, it is necessary to copy the *processing history* block from the original to modified raw data file using the *mne_copy_processing_history* utility described below. The raw data processing programs *mne_browse_raw* and *mne_process_raw* have handled copying of the processing history since revision 2.5 of the MNE software.

mne_copy_processing_history is simple to use:

`mne_copy_processing_history --from <from> --to <to>`,

where *<from>* is an original raw data file containing the processing history and *<to>* is a file output with older MNE Matlab routines. Be careful: this operation cannot be undone. If the *<from>* file does not have the processing history block or the *<to>* file already has it, the destination file remains unchanged.

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.

--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 $\langle name \rangle$ 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 4.4.12.

Tip: You might consider renaming the EEG channels with descriptive labels related to the standard 10-20 system using the *mne_rename_channels* utility, see Section 11.4.5. This allows you to use standard EEG channel names in the derivations you define as well as in the channel selection files used in *mne_browse_raw*, see Section 4.5.5.

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. 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, $R = 20/(\pi/2)$, *i.e.* at the equator ($\theta = \pi/2$) the multiplier is 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. The width and height of the viewport can be adjusted with the `--width` and `--height` options

The command-line options are:

--version

Show the program version and compilation date.

--help

List the command-line options.

--lout <name>

Specifies the name of the layout file to be output.

--nofit

Do not fit a sphere to the electrode locations but use a standard sphere center ($x = y = 0$, and $z = 40$ mm instead).

--prad <value>

Specifies a non-standard projection radius R , see above.

--width <value>

Specifies the width of the viewports. Default value = 5.

--height <value>

Specifies the height of the viewports. Default value = 4.

11.7 Adding neighborhood information to source spaces

11.7.1 Purpose

The utility *mne_add_patch_info* uses the detailed cortical surface geometry information to add data about cortical patches corresponding to each source space point. A new copy of the source space(s) included in the input file is created with the patch information included. In addition to the patch information, *mne_add_patch_info* can optionally calculate distances, along the cortical surface, between the vertices selected to the source space.

Note: Depending on the speed of your computer and the options selected, *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.

--verbose

Provide verbose output during the calculations.

--dist <dist/mm>

Invokes the calculation of distances between vertices included in the source space along the cortical surface. Only pairs whose distance in the three-dimensional volume is less than the specified distance are considered. For details, see Section 11.7.3, below.

--src <name>

The input source space file. The source space files usually end with `-src.fif`.

--srp <name>

The output source space file which will contain the patch information. If the file exists it will be overwritten without asking for permis-

sion. 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.

--labeldir <directory>

Create a label file corresponding to each of the patches in the given directory. The directory must be created before running *mne_add_patch_info*.

11.7.3 Computational details

By default, *mne_add_patch_info* creates a copy of the source space(s) with the following additional information for each vertex in the original dense triangulation of the cortex:

1. The number of the closest active source space vertex and
2. The distance to this vertex.

This information can be used to determine, *e.g.*, the sizes of the patches, their average normals, and the standard deviation of the normal directions. This information is also returned by the `mne_read_source_space` Matlab function as described in Table 10.28.

The `--dist` option to *mne_add_patch_info* invokes the calculation of inter-vertex distances. These distances are computed along the the cortical surface (usually the white matter) on which the source space vertices are located.

Since the calculation of all possible distances would take a very long time, the distance given with the `--dist` option allows restriction to the neighborhood of each source space vertex. This neighborhood is defined as the sphere around each source space vertex, with radius given by the `--dist` option. Because the distance calculation is done along the folded cortical surface whose details are given by the dense triangulation of the cortical surface produced by FreeSurfer, some of the distances computed will be larger than the value give with `--dist`.

11.8 Converting covariance data into an SSP operator

11.8.1 Purpose

The utility *mne_cov2proj* picks eigenvectors from a covariance matrix and outputs them as a signal-space projection (SSP) file.

11.8.2 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.

11.9 Fitting a sphere to a surface

11.9.1 Purpose

The utility *mne_fit_sphere_to_surf* finds the sphere which best fits a given surface.

11.9.2 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.

--mrtrans <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.

11.10 Computing sensitivity maps

11.10.1 Purpose

mne_sensitivity_map computes the size of the columns of the forward operator and outputs the result in w files.

11.10.2 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.

--proj <name>

Specifies a file containing an SSP operator to be applied. If necessary, multiple --proj options can be specified. For map types 1 - 4 (see below), SSP is applied to the forward model data. For map types 5 and 6, the effects of SSP are evaluated against the unmodified forward model.

--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 smooth steps 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 - 6. For details, see Section 11.10.3, below.

11.10.3 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 .

Furthermore, assume that the columns orthogonal matrix \mathbf{U}_P ($\mathbf{U}_P^T \mathbf{U}_P = \mathbf{I}$) contain the orthogonal basis of the noise subspace corresponding to the signal space projection (SSP) operator \mathbf{P} specified with one or more `--proj` options so that $\mathbf{P} = \mathbf{I} - \mathbf{U}_P \mathbf{U}_P^T$. For more information on SSP, see Section 4.16.

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*.

--map 5

Compute the subspace correlation between \mathbf{g}_z and \mathbf{U}_p : $\text{subcorr}^2(\mathbf{g}_z, \mathbf{U}_p) = (\mathbf{g}_z^T \mathbf{U}_p \mathbf{U}_p^T \mathbf{g}_z) / (\mathbf{g}_z^T \mathbf{g}_z)$. This index equals zero, if \mathbf{g}_z is orthogonal to \mathbf{U}_p and one if \mathbf{g}_z lies in the subspace defined by \mathbf{U}_p . This map shows how close the field pattern of a dipole oriented perpendicular to the cortex at each cortical location is to the subspace removed by the SSP.

--map 6

Compute $\sqrt{(\mathbf{g}_z^T \mathbf{P} \mathbf{g}_z) / (\mathbf{g}_z^T \mathbf{g}_z)}$, which is the fraction of the field pattern of a dipole oriented perpendicular to the cortex at each cortical location remaining after applying the SSP a dipole remaining

11.11 Transforming locations

11.11.1 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.

11.11.2 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 or fiducial, eeg, extra) and a sequence number, see Section 9.3.1.

--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.

11.12 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 limits specified by previous **--bmin** and **--bmax** options will not have an effect.

--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, previous **--baselines** option will be ignored.

--bmax <value/ms>

Upper limit of the baseline.

11.13 Data simulator

11.13.1 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.

11.13.2 Command-line options

mne_simu has 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. By default, this will be an evoked data file in the fif format.

--raw

Output the data as a raw data fif file instead of an evoked one.

--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. By default, the sources in the labels will have \cos^2 -shaped non-overlapping time-courses, see below.

--timecourse <name>

Specifies a text file which contains an expression for a source time course, see 11.13.5. If no --timecourse options are present, the standard source time courses described in Section 11.13.4 are used. Otherwise, the time course expressions are read from the files specified. The time course expressions are associated with the labels in the order they are specified. If the number of expressions is smaller than the number of labels, the last expression specified will be reused for the remaining labels.

--sfreq <freq/Hz>

Specifies the sampling frequency of the output data (default = 1000 Hz). This option is used only with the time course files.

--tmin <time/ms>

Specifies the starting time of the data, used only with time course files (default -200 ms).

--tmax <time/ms>

Specifies the ending time of the data, used only with time course files (default 500 ms).

--seed <number>

Specifies the seed for random numbers. This seed is used both for adding noise, see Section 11.13.3 and for random numbers in source waveform expressions, see Section 11.13.5. If no seed is specified, the current time in seconds since Epoch (January 1, 1970) is used.

--all

Activate all sources on the cortical surface uniformly. This overrides the --label options.

11.13.3 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 $L = 1$.

11.13.4 Simulated data

The default 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 $50 \text{ nAm}/N_k$.

Let us denote the sums of the magnetic fields and electric potentials produced by the dipoles normal to the cortical mantle at label k by \mathbf{x}_k . 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.

11.13.5 Source waveform expressions

The `--timecourse` option provides flexible possibilities to define the source waveforms in a functional form. The source waveform expression files consist of lines of the form:

<variable> = <arithmetic expression>

Each file may contain multiple lines. At the end of the evaluation, only the values in the variable y (q) are significant, see Table 11.3. They assume the role of $q_k(t_j)$ to compute the simulated signals as described in Section 11.13.4, above.

All expressions are case insensitive. The variables are vectors with the length equal to the number of samples in the responses, determined by the `--tmin`, `--tmax`, and `--sfreq` options. The available variables are listed in Table 11.3.

Variable	Meaning
x	time [s]
t	current value of x in [ms]
y	the source amplitude [Am]
q	synonym for y
a, b, c, d	help variables, initialized to zeros

Table 11.3 Available variable names in source waveform expressions.

The arithmetic expressions can use usual arithmetic operations as well as mathematical functions listed in Table 11.4. The arguments can be vectors or scalar numbers. In addition, standard relational operators (`<`, `>`, `==`, `<=`, `>=`) and their textual equivalents (`lt`, `gt`, `eq`, `le`, `ge`) are available. Table 11.5 gives some useful examples of source waveform expressions.

Function	Description
<code>abs(x)</code>	absolute value
<code>acos(x)</code>	$\cos^{-1} x$
<code>asin(x)</code>	$\sin^{-1} x$
<code>atan(x)</code>	$\tan^{-1} x$

Table 11.4 Mathematical functions available for source waveform expressions

Function	Description
atan2(x,y)	$\tan^{-1}(y/x)$
ceil(x)	nearest integer larger than x
cos(x)	$\cos x$
cosw(x,a,b,c)	\cos^2 -shaped window centered at b with a rising slope of length a and a trailing slope of length b .
deg(x)	The value of x converted to from radians to degrees
erf(x)	$\frac{1}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt$
erfc(x)	$1 - \text{erf}(x)$
exp(x)	e^x
floor(x)	Largest integer value not larger than x
hypot(x,y)	$\sqrt{x^2 + y^2}$
ln(x)	$\ln x$
log(x)	$\log_{10} x$
maxp(x,y)	Takes the maximum between x and y
minp(x,y)	Takes the minimum between x and y
mod(x,y)	Gives the remainder of x divided by y
pi	Ratio of the circumference of a circle and its diameter.
rand	Gives a vector of uniformly distributed random numbers from 0 to 1.
rnorm(x,y)	Gives a vector of Gaussian random numbers distributed as $N(x, y)$. Note that if x and y are vectors, each number generated will a different mean and variance according to the arguments.

Table 11.4 Mathematical functions available for source waveform expressions

Function	Description
shift(x,s)	Shifts the values in the input vector x by the number of positions given by s . Note that s must be a scalar.
sin(x)	$\sin x$
sqr(x)	x^2
sqrt(x)	\sqrt{x}
tan(x)	$\tan x$

Table 11.4 Mathematical functions available for source waveform expressions

Expression	Meaning
$q = 20e-9 * \sin(2 * \pi * 10 * x)$	A 10-Hz sine wave with 20 nAm amplitude
$q = 20e-9 * \sin(2 * \pi * 2 * x) * \sin(2 * \pi * 10 * x)$	A 10-Hz 20-nAm sine wave, amplitude modulated sinusoidally at 2 Hz.
$q = 20e-9 * \cosw(t, 100, 100, 100)$	\cos^2 -shaped pulse, centered at $t = 100$ ms with 100 ms leading and trailing slopes, 20 nAm amplitude.
$q = 30e-9 * (t > 0) * (t < 300) * \sin(2 * \pi * 20 * x)$	20-Hz sine wave, 30 nAm amplitude, cropped in time to 0...300 ms.

Table 11.5 Examples of source waveform expressions.

11.14 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 appear-

ance 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 2.4 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 2.4.
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
sample/audvis_raw.fif	The raw MEG/EEG data
audvis.ave	A template script for off-line averaging
auvis.cov	A template script for the computation of a noise-covariance matrix

Table 12.2 Contents of the MEG/sample directory.

File / directory	Contents
bem	Directory for the forward modelling data
bem/watershed	BEM surface segmentation data computed with the watershed algorithm
bem/inner_skull.surf	Inner skull surface for BEM
bem/outer_skull.surf	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. 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 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. Go to *Windows/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 4.13, 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 the original 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 topographical display 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 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 4.14

12.11 MEG-MRI coordinate system alignment

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 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. Three 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.

12.11.2 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: Use the ICP alignment option in *mne_analyze* with caution. The iteration will not converge to a reasonable solution unless an initial alignment is performed first according to Section 12.11.1. 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.

12.11.3 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.

12.12 The forward solution

To compute the forward solution, say:

```
cd $SAMPLE/MEG/sample
mne_do_forward_solution --mindist 5 \
```

```
--spacing oct-6 --bem sample-5120-5120-5120 \
--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 modeling

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.

F. Segonne, A. M. Dale, E. Busa, M. Glessner, D. Salat, H. K. Hahn, and B. Fischl, "A hybrid approach to the skull stripping problem in MRI," *Neuroimage*, vol. 22, pp. 1060-75, Jul 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: com-

binning 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*, vol. 27, pp. 1-13, 2006.

T. F. Oostendorp, J. Delbeke, and D. F. Stegeman, "The conductivity of the human skull: results of in vivo and in vitro measurements," *IEEE Trans Biomed Eng*, vol. 47, pp. 1487-92, Nov 2000.

S. I. Gonçalves, J. C. de Munck, J. P. Verbunt, F. Bijma, R. M. Heethaar, and F. Lopes da Silva, "In vivo measurement of the brain and skull resistivities using an EIT-based method and realistic models for the head," *IEEE Trans Biomed Eng*, vol. 50, pp. 754-67, 2003.

S. Lew, C. H. Wolters, A. Anwander, S. Makeig, and R. S. MacLeod, "Improved EEG source analysis using low-resolution conductivity estimation in a four-compartment finite element head model," *Hum Brain Mapp*, vol. 30, pp. 2862-78, 2009.

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 [Segonne *et al.*, 2004] 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. The following options can be specified:

--help

Prints the usage information.

--usage

Prints the usage information.

--noconvert

Skip conversion of the original MRI data. The original data are not needed and the preparatory steps 1. – 3. listed below are thus not needed.

--noflash30

The 30-degree flip angle data are not used.

--unwarp <type>

Run *grad_unwarp* with *-unwarp <type>* option on each of the converted data sets.

Before running *mne_flash_bem* do the following:

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 flip angle 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

Note: If `mne_flash_bem` is run with the `--noflash30` option, the `flash30` directory is not needed, *i.e.*, only the 5-degree flip angle flash data are employed.

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 creates an `mgz` file corresponding to each of the eight echoes in each of the FLASH directories in `mri/flash`. The files will be called `mef<flip-angle>_<echo-number>.mgz`.
2. If the `--unwarp` option is specified, run `grad_unwarp` and produce files `mef<flip-angle>_<echo-number>u.mgz`. These files will be then used in the following steps.
3. It creates parameter maps in `mri/flash/parameter_maps` using `mri_ms_fitparms`.
4. It creates a synthetic 5-degree flip angle volume in `mri/flash/parameter_maps/flash5.mgz` using `mri_synthesize`.
5. 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`.
6. 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.
7. It runs `mri_make_bem_surfaces` to create the BEM surface tessellations.
8. It creates the directory `bem/flash`, moves the tri-format tringulations there and creates the corresponding FreeSurfer surface files in the same directory.
9. The COR format volumes created by `mne_flash_bem` are removed.

If the `--noflash30` option is specified to `mne_flash_bem`, steps 3 and 4 in the above are replaced by averaging over the different echo times in 5-degree flip angle data.

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 manual. 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 Using *BrainSuite*

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 accomplished with the *mne_convert_surface* utility, covered in Section 9.7.

The workflow needed to employ the *BrainSuite* tessellations is:

Step 1

Using the *mri_convert* utility available in *FreeSurfer*, convert an MRI volume to the img (Analyze) format. This volume should be the T1.mgz volume or a volume registered with T1.mgz in *FreeSurfer*:

```
mri_convert <volume>.mgz <volume>.img
```

Step 2

Transfer *<volume>.mgz* to a location accessible to *BrainSuite*, running on Windows.

Step 3

Using *<volume>.img* as input, create the tessellations of scalp, outer skull, and inner skull surfaces in *BrainSuite*.

Step 4

Transfer the dfs files containing the tessellations in the *bem* directory of your subject's *FreeSurfer* reconstruction.

Step 5

Go to the *bem* directory where you placed the two dfs files. Using *mne_convert_surface*, convert them to the *FreeSurfer* surface for-

mat, e.g.:

```
mne_convert_surface \  
--dfs inner_skull.dfs \  
--mghmri ../mri/T1.mgz \  
--surf inner_skull_dfs.surf
```

Step 6

Using tkmedit, check that the surfaces are correct, e.g.:

```
tkmedit -f ../mri/T1.mgz \  
-surface inner_skull_dfs.surf
```

Step 7

Using the *mne_reduce_surface* function in Matlab, reduce the number of triangles on the surfaces to 10000 - 20000. Call the output files *outer_skin.surf*, *outer_skull.surf*, and *inner_skull.surf*.

Step 8

Proceed to *mne_setup_forward_model*. Use the *--surf* and *--noswap* options.

Note: If left and right are flipped in BrainSuite, use the *--flip* option in *mne_convert_surface* to set the coordinate transformation correctly.

Tip: The *BrainSuite* scalp surface can be also used for visualization in *mne_analyze*, see Section 7.16.1.

APPENDIX B Setup at the Martinos Center

This Appendix contains information specific to the Martinos Center setup.

B.1 User environment

In the Martinos Center computer network, the 2.7 version of MNE is located at `/usr/pubsw/packages/mne/stable`. To use this version, follow Section 2.4 substituting `/usr/pubsw/packages/mne/stable` for `<MNE>` and `/usr/pubsw/packages/matlab/current` for `<Matlab>`. For most users, the default shell is `tcsh`.

Tip: A new version of MNE is build every night from the latest sources. This version is located at `/usr/pubsw/packages/mne/nightly`.

B.2 Using Neuromag software

B.2.1 Software overview

The complete set of Neuromag software is available on the LINUX workstations. The programs can be accessed from the command line, see Table B.1. The corresponding manuals, located at `$NEUROMAG_ROOT/manuals` are listed in Table B.2.

Module	Description
xfit	Source modelling
xplotter	Data plotting
graph	General purpose data processor
mrilab	MEG-MRI integration
seglab	MRI segmentation
cliplab	Graphics clipboard

Table B.1 Principal Neuromag software modules.

Module	pdf
xfit	XFit.pdf
xplotter	Xplotter.pdf
graph	GraphUsersGuide.pdf GraphReference.pdf
mrilab	Mrilab.pdf
seglab	Seglab.pdf
cliplab	Cliplab.pdf

Table B.2 List of Neuromag software manuals.

To access the Neuromag software on the LINUX workstations in the Martinos Center, say (in tcsh or csh)

```
source /space/orsay/8/megdev/Neuromag-LINUX/  
neuromag_setup_csh
```

or in POSIX shell

```
. /space/orsay/8/megdev/Neuromag-LINUX/  
neuromag_setup_sh
```

B.2.2 Using MRilab for coordinate system alignment

The MEG-MRI coordinate system alignment can be also accomplished with the Neuromag tool MRilab, part of the standard software on Neuromag MEG systems.

In 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*.

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.

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 (Am). 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 up to 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.

APPENDIX C Installation and configuration

C.1 System requirements

The MNE software runs on Mac OSX and LINUX operating systems. The hardware and software requirements are:

1. Mac OSX version 10.5 (Leopard) or later.
2. LINUX kernel 2.6.9 or later
3. On both LINUX and Mac OSX 32-bit and 64-bit Intel platforms are supported. PowerPC version on Mac OSX can be provided upon request.
4. At least 2 GB of memory, 4 GB or more recommended.
5. Disk space required for the MNE software: 80 MB
6. Additional open source software on Mac OSX, see Section C.2.4.

C.2 Installation

The MNE software is distributed as a compressed tar archive (Mac OSX and LINUX) or a Mac OSX disk image (dmg).

C.2.1 Download the software

Download the software package of interest. The file names follow the convention:

MNE-*<version>*-*<rev>*-*<Operating system>*-*<Processor>*.*<ext>*

The present version number is 2.7.0. The *<rev>* field is the SVN revision number at the time this package was created. The *<Operating system>* field is either Linux or MacOSX. The *<processor>* field is either i386 or x86_64. The *<ext>* field is 'gz' for compressed tar archive files and 'dmg' for Mac OSX disk images.

C.2.2 Installing from a compressed tar archive

Go to the directory where you want the software to be installed:

```
cd <dir>
```

Unpack the tar archive:

```
tar zxvf <software package>
```

The name of the software directory under <dir> will be the same as the package file without the .gz extension.

C.2.3 Installing from a Mac OSX disk image

- Double click on the disk image file. A window opens with the installer package (<name>.pkg) inside.
- Double click the package file. The installer starts.
- Follow the instructions in the installer.

Tip: The software will be installed to /Applications/<name> by default. If you want another location, select *Choose Folder...* on the *Select a Destination* screen in the installer.

Tip: To provide centralized support in an environment with

C.2.4 Additional software

MNE uses the ‘Netpbm’ package (<http://netpbm.sourceforge.net/>) to create image files in formats other than tif and rgb from *mne_analyze* and *mne_browse_raw*. This package is usually present on LINUX systems. On Mac OSX, you need to install the netpbm package. The recommended way to do this is to use the MacPorts Project tools, see <http://www.macports.org/>:

1. If you have not installed the MacPorts software, goto <http://www.macports.org/install.php> and follow the instructions to install MacPorts.
2. Install the netpbm package by saying:

```
sudo port install netpbm
```

MacPorts requires that you have the XCode developer tools and X11 windowing environment installed. X11 is also needed by MNE. For Mac OSX Leopard, we recommend using XQuartz (<http://xquartz.macosforge.org/>). As of this writing, XQuartz does not yet exist for SnowLeopard; the X11 included with the operating system is sufficient.

C.2.5 Testing the performance of your OpenGL graphics

The graphics performance of *mne_analyze* depends on your graphics software and hardware configuration. You get the best performance if you are

using *mne_analyze* locally on a computer and the hardware acceleration capabilities are in use. You can check the *On GLX...* item in the help menu of *mne_analyze* to see whether the hardware acceleration is in effect. If the dialog popping up says *Direct rendering context*, you are using hardware acceleration. If this dialog indicates *Nondirect rendering context*, you are either using software emulation locally, rendering to a remote display, or employing VNC connection. If you are rendering to a local display and get an indication of *Nondirect rendering context*, software emulation is in effect and you should contact your local computer support to enable hardware acceleration for GLX. In some cases, this may require acquiring a new graphics display card. Fortunately, relatively high-performance OpenGL-capable graphics cards very inexpensive.

There is also an utility *mne_opengl_test* to assess the graphics performance more quantitatively. This utility renders an inflated brain surface repeatedly, rotating it by 5 degrees around the *z* axis between redraws. At each revolution, the time spent for the full revolution is reported on the terminal window where *mne_opengl_test* was started from. The program renders the surface until the interrupt key (usually control-c) is pressed on the terminal window.

mne_opengl_test is located in the `bin` directory and is thus started as:

```
$MNE_ROOT/bin/mne_opengl_test
```

On the fastest graphics cards, the time per revolution is well below 1 second. If this time longer than 10 seconds either the graphics hardware acceleration is not in effect or you need a faster graphics adapter.

C.3 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.4 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 2.4, 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 5.8.4.
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 9.2.9.

The output of the Matlab conversion utilities was changed to use structures. For details, see Sections 9.12, 9.7, and 9.13.

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 Section 7.16.
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 4.6.4.
- 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 4.4.12 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 9.14.

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 7.8.8.
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 7.15.3.
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 5.6.

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 9.2.7.

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.

D.3 Release notes for MNE software 2.6

D.3.1 Manual

The changes described below briefly are documented in the relevant sections of the manual. Change bars are employed to indicate changes with respect to manual version 2.5. Chapter 5 now contains a comprehensive discussion of the various coordinate systems used in MEG/EEG data.

D.3.2 Command-line options

All compiled C programs now check that the command line does not contain any unknown options. Consequently, scripts that have inadvertently specified some options which are invalid will now fail.

D.3.3 Changes to existing software

D.3.3.1 *mne_add_patch_info*

- Changed option `--in` to `--src` and `--out` to `--srcp`.
- Added `--labeldir` option.

D.3.3.2 *mne_analyze*

New features include:

- The name of the digital trigger channel can be specified with the `MNE_TRIGGER_CH_NAME` environment variable.
- Using information from the *fif* data files, the wall clock time corresponding to the current file position is shown on the status line
- *mne_analyze* can now be controlled by *mne_browse_raw* to facilitate interactive analysis of clinical data.
- Added compatibility with Elekta-Neuromag Report Composer (*cliplab*) and improved the quality of hardcopies.
- Both in *mne_browse_raw* and in *mne_analyze*, a non-standard default layout can be set on a user-by-user basis, see Section 4.5.6.
- Added the `--digtrigmask` option.
- Added new image rotation functionality using the mouse wheel or trackball.

- Added remote control of the *FreeSurfer* MRI viewer (*tkmedit*), see Section 7.18.
- Added fitting of single equivalent current dipoles and channel selections, see Section 7.15.
- Added loading of *FreeSurfer* cortical parcellation data as labels.
- Added support for using the *FreeSurfer* average brain (*fsaverage*) as a surrogate.
- The surface selection dialog was redesigned for faster access to the files and to remove problems with a large number of subjects.
- A shortcut button to direct a file selector to the appropriate default directory was added to several file loading dialogs.
- The vertex coordinates can now be displayed, see Section 7.8.4.

D.3.3.3 mne_average_forward_solutions

EEG forward solutions are now averaged as well, see Section 5.10.

D.3.3.4 mne_browse_raw and mne_process_raw

Improvements in the raw data processor *mne_browse_raw*/*mne_process_raw* include:

- The name of the digital trigger channel can be specified with the `MNE_TRIGGER_CH_NAME` environment variable.
- The format of the text event files was slightly changed. The sample numbers are now “absolute” sample numbers taking into account the initial skip in the event files. The new format is indicated by an additional “pseudoevent” in the beginning of the file. *mne_browse_raw* and *mne_process_raw* are still compatible with the old event file format. For details, see Section 4.10.5.
- Using information from the *fif* data files, the wall clock time corresponding to the current file position is shown on the status line
- *mne_browse_raw* can now control *mne_analyze* to facilitate interactive analysis of clinical data.
- If the length of an output raw data file exceeds the 2-Gbyte *fif* file size limit, the output is split into multiple files.
- `--split` and `--events` options was added to *mne_process_raw*.
- The `--allowmaxshield` option was added to *mne_browse_raw* to allow loading of unprocessed data with MaxShield in the Elekta-Neuromag systems. These kind of data should never be used as an input for source localization.
- The `--savehere` option was added, see Section 4.2.3.
- The *stderr* parameter was added to the averaging definition files, see Section 4.13.3.
- Added compatibility with Elekta-Neuromag Report Composer (*cliplab*) and improved the quality of hardcopies.
- Both in *mne_browse_raw* and in *mne_analyze*, a non-standard default layout can be set on a user-by-user basis, see Section 4.5.6.

- *mne_browse_raw* now includes an interactive editor to create derived channels, see Section 4.5.4.
- The menus in *mne_browse_raw* were reorganized and an time point specification text field was added
- Possibility to keep the old projection items added to the new projection definition dialog.
- Added `--cd` option.
- Added filter buttons for raw files and Maxfilter™ output to the open dialog.
- Added possibility to create a graph-compatible projection to the *Save projection* dialog
- Added possibility to compute a projection operator from epochs specified by events.
- Added the *keepsamplemean* option to the covariance matrix computation files.
- Added the `--digtrigmask` option.
- Added *Load channel selections...* item to the *File* menu.
- Added new browsing functionality using the mouse wheel or trackball, see Section 4.9.1.
- Added optional items to the topographical data displays, see Section 4.5.2.
- Added an event list window, see Section 4.10.2.
- Added an annotator window, see Section 4.10.4.
- Keep events sorted by time.
- User-defined events are automatically kept in a fif-format annotation file, see Section 4.10.
- Added the *delay* parameter to the averaging and covariance matrix estimation description files, see Sections 4.13.3 and 4.14.3.

Detailed information on these changes can be found in Chapter 4.

D.3.3.5 *mne_compute_raw_inverse*

The `--digtrig`, `--extra`, `--noextra`, `--split`, `--labeldir`, and `--out` options were added, see Section 6.6.

D.3.3.6 *mne_convert_surface*

The functionality of *mne_convert_dfs* was integrated into *mne_convert_surface*. Text output as a triangle file and and file file containing the list of vertex points was added. The Matlab output option was removed. Consequently, *mne_convert_dfs*, *mne_surface2mat*, and *mne_list_surface_nodes* were deleted from the distribution.

D.3.3.7 *mne_dump_triggers*

This obsolete utility was deleted from the distribution.

D.3.3.8 mne_epochs2mat

The name of the digital trigger channel can be specified with the MNE_TRIGGER_CH_NAME environment variable, see Section 9.14. Added the `--digtrigmask` option.

D.3.3.9 mne_forward_solution

Added code to compute the derivatives of with respect to the dipole position coordinates, see Section 5.9.

D.3.3.10 mne_list_bem

The `--surfno` option is replaced with the `--id` option, see Section 9.6.

D.3.3.11 mne_make_cor_set

Include data from mgh/mgz files to the output automatically. Include the Talairach transformations from the FreeSurfer data to the output file if possible. For details, see Section 9.8.

D.3.3.12 mne_make_movie

Added the `--noscalebar`, `--nocomments`, `--morphgrade`, `--rate`, and `--pick-range` options, see Section 6.5.

D.3.3.13 mne_make_source_space

The `--spacing` option is now implemented in this program, which means *mne_mris_trix* is now obsolete. The *mne_setup_source_space* script was modified accordingly. Support for tri, dec, and dip files was dropped, see Section 5.4.

D.3.3.14 mne_mdip2stc

This utility is obsolete and was removed from the distribution.

D.3.3.15 mne_project_raw

This utility is obsolete and was removed from the distribution. The functionality is included in *mne_process_raw*.

D.3.3.16 mne_rename_channels

Added the `--revert` option, see Section 11.4.5.

D.3.3.17 mne_setup_forward_model

Added the `--outershift` and `--scalpshift` options, see Section 3.7.

D.3.3.18 mne_simu

Added source waveform expressions and the `--raw` option, see Section 11.13.

D.3.3.19 mne_transform_points

Removed the `--tomrivol` option.

D.3.3.20 Matlab toolbox

Several new functions were added, see Chapter 10.

Important: The matlab function *fiff_setup_read_raw* has a significant change. The sample numbers now take into account possible initial skip in the file, *i.e.*, the time between the start of the data acquisition and the start of saving the data to disk. The *first_samp* member of the returned structure indicates the initial skip in samples. If you want your own routines, which assume that initial skip has been removed, perform indentially with the previous version, subtract *first_samp* from the sample numbers you specify to *fiff_read_raw_segment*. Furthermore, *fiff_setup_read_raw* has an optional argument to allow reading of unprocessed MaxShield data acquired with the Elekta MEG systems.

D.3.4 New utilities

D.3.4.1 mne_collect_transforms

This utility collects coordinate transformation information from several sources into a single file, see Section 9.9.

D.3.4.2 mne_convert_dig_data

This new utility convertes digitization (Polhemus) data between different file formats, see Section 9.3.

D.3.4.3 `mne_edf2fif`

This is a new utility to convert EEG data from EDF, EDF+, and BDF formats to the fif format, see Section 9.2.8.

D.3.4.4 `mne_brain_vision2fif`

This is a new utility to convert BrainVision EEG data to the fif format, see Section 9.2.10. This utility is also used by the `mne_eximia_2fif` script to convert EEG data from the Nexstim eXimia EEG system to the fif format, see Section 9.2.11.

D.3.4.5 `mne_anonymize`

New utility to remove subject identifying information from measurement files, see Section 11.4.7.

D.3.4.6 `mne_opengl_test`

New utility for testing the OpenGL graphics performance, see Section C.2.5.

D.3.4.7 `mne_volume_data2mri`

Convert data defined in a volume created with `mne_volume_source_space` to an MRI overlay, see Section 9.4.

D.3.4.8 `mne_volume_source_space`

Create a a grid of source points within a volume, see Section 5.5. `mne_volume_source_space` also optionally creates a trilinear interpolator matrix to facilitate converting values a distribution in the volume grid into an MRI overlay using `mne_volume_data2mri`, see Section 9.4.

D.3.4.9 `mne_copy_processing_history`

This new utility copies the processing history block from one data file to another, see Section 11.4.8.

D.4 Release notes for MNE software 2.7

D.4.1 Software engineering

There have been two significant changes in the software engineering since MNE Version 2.6:

- CMake is now used in building the software package and
- Subversion (SVN) is now used for revision control instead of Concurrent Versions System (CVS).

These changes have the effects on the distribution of the MNE software and setup for individual users:

- There is now a separate software package for each of the platforms supported.
- The software is now organized completely under standard directories (bin, lib, and share). In particular, the directory setup/mne has been moved to share/mne and the directories app-defaults and doc are now under share. All files under share are platform independent.
- The use of shared libraries has been minimized. This alleviates compatibility problems across operating system versions.
- The setup scripts have changed.

The installation and user-level effects of the new software organization are discussed in Chapter 2 and Appendix C.

In addition, several minor bugs have been fixed in the source code. Most relevant changes visible to the user are listed below.

D.4.2 Matlab tools

- The performance of the fiff I/O routines has been significantly improved thanks to the contributions of François Tadel at USC.
- Label file I/O routines `mne_read_label_file` and `mne_write_label_file` as well as a routine to extract time courses corresponding to a label from an stc file (`mne_label_time_courses`) have been added.
- The patch information is now read from the source space file and included in the source space data structure.

D.4.3 `mne_browse_raw`

- Rejection criteria to detect flat channels have been added, see Sections 4.13.2 and 4.14.2.
- Possibility to detect temporal skew between trigger input lines has been added, see Sections 4.13.2 and 4.14.2.
- `--allowmaxshield` option now works in the batch mode as well.

- Added the `--projevent` option to batch mode.
- It is now possible to compute an SSP operator for EEG, see Section 4.6.4.

D.4.4 `mne_analyze`

- Both hemispheres can now be displayed simultaneously, see Section 7.8.3.
- If the source space was created with `mne_make_source_space` version 2.3 or later, the subject's surface data are automatically loaded after loading the data and the inverse operator.

D.4.5 Miscellaneous

- `mne_smooth_w` was renamed to `mne_smooth` and can now handle both `w` and `stc` files. Say `mne_smooth --help` to find the options.
- All binaries now reside in `$MNE_ROOT/bin`. There are no separate `bin/mne` and `bin/admin` directories.
- `mne_anonymize` now has the `--his` option to remove the HIS ID of the subject, see Section 11.4.7,
- `mne_check_surface` now has the `--bem` and `--id` options to check surfaces from a BEM `fif` file. For details, try `mne_check_surface --help`.
- `mne_compute_raw_inverse` now has the `--orignames` option, see Section 6.6.1.
- Added `--headcoord` option to `mne_convert_dig_data`, see Section 9.3.
- Added `--talairach` option to `mne_make_cor_set`, see Section 9.8.
- Added the `--morph` option to `mne_setup_source_space` and `mne_make_source_space`, see Sections 3.5 and 5.4, respectively.
- Added the `--prefix` option to `mne_morph_labels`, see Section 8.5.
- Added the `--blocks` and `--indent` options to `mne_show_fiff`, see Section 11.3.
- Added the `--proj` option as well as map types 5 and 6 to `mne_sensitivity_map`, see Section 11.10.
- Fixed a bug in `mne_inverse_operator` which caused erroneous calculation of EEG-only source estimates if the data were processed with Maxfilter software and sometimes caused similar behavior on MEG/EEG source estimates.

D.5 Release notes for MNE software 2.7.1

D.5.1 `mne_analyze`

- Added a new restricted mode for visualizing head position within the helmet, see Sections 7.2 and 7.21.
- Added information about `mne_make_scalp_surfaces` to Section 7.16.1.

D.5.2 `mne_browse_raw`

- Added possibility for multiple event parameters and the mask parameter in averaging and noise covariance calculation, see Section 4.13.3.
- Added simple conditional averaging, see Section 4.13.3.

D.6 Release notes for MNE software 2.7.2

D.6.1 `mne_add_patch_info`

Added the capability to compute distances between source space vertices, see Section 11.7.

D.6.2 Matlab toolbox

- Added new functions to for stc and w file I/O to employ 1-based vertex numbering inside Matlab, see Table 10.11.
- `mne_read_source_spaces.m` now reads the inter-vertex distance information now optionally produced by `mne_add_patch_info`, see Section 11.7.

D.6.3 Miscellaneous

- Added `--shift` option to `mne_convert_surface`, see Section 9.7.1.
- Added `--alpha` option to `mne_make_movie`, see Section 6.5.5.
- Added `--noiserank` option to `mne_inverse_operator` and `mne_do_inverse_operator`, see Sections 6.4 and 3.13, respectively.
- The fif output from `mne_convert_dig_data` now includes the transformation between the digitizer and MNE head coordinate systems if such a transformation has been requested, see Section 9.3. This also affects the output from `mne_eximia2fiff`, see Section 9.2.11.
- Added `--noflash30`, `--noconvert`, and `--unwarp` options to `mne_flash_bem`, see Section A.2.

D.7 Release notes for MNE software 2.7.3

D.7.1 Miscellaneous

- Added preservation of the volume geometry information in the FreeSurfer surface files.
- The `--mghmri` option in combination with `--surfout` inserts the volume geometry information to the output of `mne_convert_surface`, see Section 9.7.
- Added `--replacegeom` option to `mne_convert_surface`, see Section 9.7.

- Modified *mne_watershed_bem* and *mne_flash_bem* to include the volume geometry information to the output. This allows viewing of the output surfaces in the FreeSurfer freeview utility.

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

