Standard data format MEG/EEG

ATR Neural Information Analysis Laboratories (May 10, 2017)

Contents

Revision History

2011-06-15　version 1.0.0

2012-06-29　version 1.0.1

- corrected the unit of eeg_data (3.1.1)

2017-05-10　version 2.0.0

- corrected MEGinfo.device of Minimum format(='BASIC').
- fileinfo format

1. Introduction

This document describes MEG/EEG data formats that are accepted by VBMEG. Since there is no standard file format for MEG/EEG data, unlike MRI image data for which there are several standard file formats such as DICOM, Analyze, and NIFTI, it is necessary to import device-dependent MEG/EEG data into specific file formats used in this software.

We prepared Matlab functions to import Yokogawa MEG data, Biosemi EEG data, and NeuroMag MEG data. Although we have used Yokogawa and Biosemi in our laboratory, we have not fully tested the implementation for NeuroMag. For other devices, user should prepare a `.mat` file with the format described in this document.

In this document, MEG-MAT refers to a `.meg.mat` file and EEG-MAT to a `.eeg.mat` file.

1.1. Conventions in this document

- *Italic*: Indicates variables and functions.
- `Constant width`: Indicates file names, file extensions, and program segments.

2. MEG

VBMEG accepts two MEG file formats: the standard format and the minimum data format.

2.1. Minimum MEG-MAT format

The minimum format contains MEG signals and essential information to calculate the leadfield and requires the variables listed in the tables below. In order to execute current source estimation, it is necessary to import a set of sensor positions/directions as well as channel data. MEG signals are stored in *bexp* and the positions and directions of sensors are in *pick* and *Qpick*. *MEGinfo.sensor_weight* is the coefficients that translate sensor signals into MEG channels (see Section 2.1.3). This is used for calculating the leadfield, which represent magnetic fields at the MEG channels generated by cortical currents and can be calculated by using the Sarvas equation or BEM (boundary element method); both methods are implemented in VBMEG. In general, there are multiple sensors for each MEG channel. For example, there are two sensors for one gradiometer channel.

2.1.1. Primary variables

variable	commentary
bexp	MEG measurement signals [*Nchannel*×*Nsample*×*Nrepeat*] [Tesla]
pick	(*x*, *y*, *z*) coordinates of each sensor [*Nsensor*×3] [m]
Qpick	(*x*, *y*, *z*) direction of each sensor [*Nsensor*×3] (vector with unit norm)

2.1.2. Measurement information variables

variable	commentary
Measurement	Measurement type code 'MEG' or 'EEG'
MEGinfo	<struct> Measurement information of data

2.1.2.1. MEGinfo

field	commentary
Measurement	Measurement type code 'MEG' or 'EEG'
device	Device name = ‘BASIC’
Nchannel	Number of channels in MEG data
Nsample	Number of time samples in MEG data
Nrepeat	Number of trials in MEG data
Pretrigger	Number of samples before trigger onset
SampleFreq	Sampling frequency [Hz]
sensor_weight	Sensor weight for each channel [*Nchannel*×*Nsensor*]

2.1.3. Calculation of the leadfield and *sensor_weight*

Let $G_{\text{sensor}}(m,j)$ be the magnetic field at the *m*-th sensor produced by a unit current at the *j*-th cortical vertex. In order to calculate it, we have to know the *m*-th sensor's position *pick*(*m*,:) and direction *Qpick*(*m*,:). It is also necessary to know the *j*-th cortical vertex's position $V(j,:)$ and its current direction $xx(j,:)$ stored in a `.brain.mat` file. Then, the leadfield at the *n*-th channel is given as

$$G_{channel}(n,j) = \sum_{m} sensor\_weight(n,m) \cdot G_{sensor}(m,j)$$

or

$$G_{channel} = sensor\_weight \cdot G_{sensor}$$

in matrix notation.

If all the channels are magnetometer, *sensor_weight* is the identity matrix:

$$sensor\_weight(n,m) = \begin{cases} 1 & for \quad n{=}m \\ 0 & otherwise \end{cases}$$

If there is an gradiometer at the *n*-th channel and the associated two sensors are 2n-1 and $2n$,

$$sensor\_weight(n,2n-1) = -1,$$

$$sensor\_weight(n,2n) = 1, and$$

$$sensor\_weight(n,m) = 0 \quad for \quad m \neq 2n-1, 2n$$

2.2. Standard MEG-MAT format

The standard format includes additional information useful for preprocessing and requires the variables described in the tables below. Data files made by an import function of VBMEG, such as `vb_job_meg.m`, have this file format. A data file in the standard format has standard information that is independent of the device type and device-dependent information that is useful to know detailed measurement conditions. The device-dependent information is not required in general.

2.2.1. Primary variables

variable	**commentary**
bexp	MEG measurement signals [*Nchannel*×*Nsample*×*Nrepeat*] [Tesla]
bexp_ext	External channel measurement signals [*Nchannel_ext*×*Nsample*×*Nrepeat*]
pick	(*x*, *y*, *z*) coordinates of each sensor [*Nsensor*×3] [m]
Qpick	(*x*, *y*, *z*) direction of each sensor [*Nsensor*×3] (vector with unit norm)

If *bexp* or *bexp_ext* is empty, signal data are saved as binary files (see the “Binary data file” section below).

2.2.2. Measurement information variables

variable	**commentary**
Measurement	Measurement type code ’MEG’ or ’EEG’
CoordType	Coordinate system for sensor position ’SPM_Right_m’
MEGinfo	<struct> Measurement information of data

2.2.2.1. MEGinfo

field	**commentary**
Measurement	Measurement type code 'MEG' or 'EEG'
device	Device name such as 'YOKOGAWA', 'BIOSEMI', and 'NEUROMAG'
Nchannel	Number of channels in MEG data
Nsample	Number of time samples in MEG data
Nrepeat	Number of trials in MEG data
Pretrigger	Number of time samples before trigger onset
SampleFreq	Sampling frequency [Hz]
sensor_weight	Sensor weight for each channel [*Nchannel*×*Nsensor*]
MEGch_id	List of original index of channel [*Nchannel* ×1]
MEGch_name	List of original label of channel {*Nchannel* ×1}
ActiveChannel	Active channel list (boolean array) [*Nchannel* ×1]
ActiveTrial	Active trial list (boolean array) [*Nrepeat* ×1]

Vcenter	Center coordinate of a spherical model [1×3] [m]
Vradius	Radius of a spherical model [m]
MEG_ID	MEG data ID
MRI_ID	ID for MRI image on which the sensor position is registered
Trial	<struct array> Information of each trial [*Nrepeat* ×1]
ChannelInfo	<struct> Information of MEG channels
ExtraChannelInfo	<struct> Information of extra channels
saveman	<struct> Information of binary data files

Binary data file

In some cases, the size of MEG data become so large size that it can not be held in one variable *bexp*. Therefore, imported data are saved as binary files in the default setting of `vb_job_meg.m`. In this case, each channel's data are saved in a separate binary file and *bexp* is set to be empty. On the other hand, if MEG data is set to *bexp*, the structure containing the path to a binary files folder *MEGinfo.saveman* should be empty.

The binary file path and data precision are described in *saveman* field of *MEGinfo*. Data are saved in the float64 format in sequence of time. If there are multiple trials, the first trial data are saved first, then the second trial data are saved after that, and so on. The file name is the label name of each channel and the extension of binary file is `.ch.meg.dat`. Label names are defined in *MEGinfo.MEGch_name* or in

MEGinfo.ExtraChannelInfo.Channel_name.

In Yokogawa case, binary files will have names such as `0.ch.meg.dat`, `477.ch.meg.dat`.

MEGinfo.saveman

field	commentary
data_dir	Relative path from a `.meg.mat` file to a directory where binary data file is stored
precision	Precision of a binary data: 'float64'

2.2.2.2. Substructure of *MEGinfo*

MEGinfo.ChannelInfo

field	commentary
ID	Original index list for each channel [*Nchannel* ×1]
Name	Label list for each channel [*Nchannel* ×1] cell array
Type	Type list for each channel [*Nchannel* ×1] cell array
Active	Boolean (0 or 1) list of active channel [*Nchannel* ×1]

MEGinfo.ExtraChannelInfo

field	commentary
gain	<struct> Struct array that has two fields, name and value
Channel_id	Original index list for each channel [*Nchannel_ext* ×1]

Channel_name	Label list for each channel [*Nchannel_ext* ×1] cell array
Channel_type	Type list for each channel [*Nchannel_ext* ×1] cell array
Channel_active	Boolean (0 or 1) list of active channel [*Nchannel_ext* ×1]

MEGinfo.Trial

This field is a struct array. Its size is [*Nrepeat* ×1]. It has following fields.

field	commentary
number	Index number for this trial
sample	List of time sample index for this trial
Active	Active flag for this trial (1: active, 0: non-active)

2.3. Device-dependent variables and fields

In general, the following variables and fields are not necessary.

For specific devices such as 'YOKOGAWA', the import program adds the following device-dependent variables and fields.

variable / field	commentary
ref_pick	coordinates of each reference sensor [*n_ref* ×3]
ref_Qpick	normal vectors of each reference sensor [*n_ref* ×3]
PositionFile	File path for the sensor position registration on MRI by our positioning program (`.pos.mat`) This file is used for importing sensor position by `vb_job_meg.m`. If this variable is empty, there is no such file.
MEGinfo.device_info	Device-dependent variables in *MEGinfo* are included into *device_info* struct.

2.3.1. MEGinfo.device_info

field	commentary
sensor_weight_ref	List of weight of each reference sensor [*n_ref* ×*n_ref*]
acq_type	Type of data acquisition 'Continuous_Raw'
acq_info	<struct> Data acquisition information
TransInfo	<struct> Information of a coordinate transformation
index_in_fiffile	(neuromag) Index number in fiffile
Comment	(neuromag) Comment of this data

2.3.1.1. MEGinfo.device_info.TransInfo

field	commentary

trans_mri	Coordinate transform matrix to MRI
coord_type_before	Coord type before transformation
coord_type_after	Coord type after transformation by the *trans_mri* matrix

2.3.1.2. MEGinfo.device_info.acq_info

field	**commentary**
data_file	Imported data file path
data_bit_len	Imported data bit length (16)
condition	<struct> Acquisition condition information

MEGinfo.device_info.acq_info.condition (YOKOGAWA)

This information depends on Yokogawa device.

3. EEG

VBMEG accepts two EEG file formats: the standard format and the minimum data format.

3.1. Minimum EEG-MAT format

The minimum format contains EEG signals and essential information to calculate the leadfield and requires the variables listed in the tables below. In order to execute current source estimation, it is necessary to import a set of sensor positions as well as channel data. EEG signals are stored in *eeg_data* and the positions of sensors are in *EEGinfo.Coord*. Leadfields which represent electric potentials in the EEG channels generated by cortical currents can be calculated by Legendre expansion or BEM method (boundary element method); both methods are implemented in VBMEG.

3.1.1. Primary variables

variable	**commentary**
eeg_data	EEG channel data [*Nchannel* x *Nsample* x *Nrepeat*] [V]

3.1.2. Measurement information variables

variable	**commentary**
Measurement	Measurement type code: 'MEG' or 'EEG'
EEGinfo	<struct> Measurement information of data

3.1.2.1. EEGinfo

field	**commentary**
Measurement	Measurement type code: 'MEG' or 'EEG'
Device	Device name = ‘BASIC’
Nchannel	Number of channels in EEG data
Nsample	Number of time samples in EEG data
Nrepeat	Number of trials in EEG data
Pretrigger	Number of time samples before trigger onset
SampleFrequency	Sampling frequency [Hz]
Coord	(*x*, *y*, *z*) coordinates of EEG channels [*Nchannel* x 3] [m]

3.2. Standard EEG-MAT format

The standard format includes additional information useful for preprocessing and requires the variables described in the tables below. Data files made by an import function of VBMEG, such as `vb_job_meg.m`, have this file format. A data file in the standard format has standard information that is independent of the device type and device-dependent information that is useful to know detailed measurement conditions. The device-dependent information is not required in general.

3.2.1. Primary variables

variable	commentary
eeg_data	EEG channel data [*Nchannel* x *Nsample* x *Nrepeat*] [V]

If there are external channel such as status signal, *eeg_data* has also these extra channel data. If *eeg_data* is empty, signal data are saved as binary files for each channel (see later).

3.2.2. Measurement information variables

variable	commentary
Measurement	Measurement type code: 'MEG' or 'EEG'
EEGinfo	<struct> Measurement information of data

3.2.2.1. EEGinfo

field	commentary
Measurement	Measurement type code: 'MEG' or 'EEG'
Device	Device name such as 'BIOSEMI'
Nchannel	Number of channels in EEG data
Nsample	Number of time samples in EEG data
Nrepeat	Number of trials in EEG data
Pretrigger	Number of time samples before trigger onset
SampleFrequency	Sampling frequency [Hz]
ChannelID	List of original index of channel [*Nchannel* x 1]
ChannelName	List of original label of channel {*Nchannel* x 1}
ActiveChannel	Active channel list (boolean array) [*Nchannel* x 1]
ChannelInfo	<struct> Information of EEG channels
ExtraChannelInfo	<struct> Information of extra channels
DataType	Data type list of each channel {(*Nchannel* + *Nchannel_ext*) x 1}
ActiveTrial	Active trial list (boolean array) [*Nrepeat* x 1]
Trial	<struct array> Information of each trial [*Nrepeat* x 1]
Coord	(*x*, *y*, *z*) coordinates of each channel [*Nchannel* x 3]
CoordType	Coordinate system for channel position 'SPM_Right_m'
Vcenter	Center coordinate of a spherical model [1 x 3] [m]
Vradius	Radius of a spherical model [m]
MRI_ID	ID for MRI image on which the channel position is registered
File	<struct> File information

Binary data file

In some cases, the size of EEG data become so large size that it can not be held in one variable *eeg_data*. Therefore, imported data are saved as binary files in the default setting of `vb_job_meg.m`. In this case, each channel's data are saved in a separate binary file and *eeg_data* is set to be empty. On the other hand, if EEG data is set to *eeg_data*, the structure containing the path to a binary files folder *EEGinfo.File* should be empty.

The binary file path and data precision are described in *File* field of *EEGinfo*. EEG data are saved in the float32 format in sequence of time. If there are multiple trials, the first trial data are saved first, then the second trial data are saved after that, and so on. The file name is the label name of each channel and the extension of binary file is `.ch.eeg.dat`. Label names are defined in *EEGinfo.ChannelName* or in *EEGinfo.ExtraChannelInfo.Channel_name*.

In BIOSEMI case, binary files will have names such as `FP1.ch.eeg.dat`, `CP1.ch.eeg.dat`. The precision of a status channel in BIOSEMI is bit24 (`Status.ch.eeg.dat`).

3.2.2.2. Substructure of *EEGinfo*

EEGinfo.ChannelInfo

field	commentary
Active	Boolean list that shows each channel is active or not
Name	Label list for each channel
Type	Type list for each channel
ID	Original index list for each channel
PhysicalUnit	List of physical units

EEGinfo.ExtraChannelInfo

field	commentary
Channel_active	Boolean list that shows each channel is active or not
Channel_name	Label list for each channel
Channel_type	Type list for each channel
Channel_id	Original index list for each channel
PhysicalUnit	List of physical units

EEGinfo.File

field	commentary
BaseFile	EEG file to be imported
OutputDir	Directory that an output `.eeg.mat` file is made
EEGFile	`.eeg.mat` file that was created by importing data
DataDir	Relative path from the EEGFile for the imported data directory

EEGinfo.Trial (array struct)

field	commentary
number	Index number of this trial
sample	List of sample index number for this trial
Active	Active flag

3.3. Device dependent variables and fields

In general, the following variables and fields are not necessary.

For specific devices such as 'BIOSEMI', the import program adds the following device dependent variables and fields.

3.3.1. EEGinfo.device_info

field	commentary
Reference	Reference method to decide coordinates
RecordTime	Recording time [sec]
TransInfo	<struct> Information of a coordinate transformation
Header	<struct> Biosemi header information
History	List of histories of this file
Version	Version of this struct

3.3.1.1. EEGinfo.device_info.TransInfo

field	commentary
trans_mri	Matrix to transform coordinates
coord_type_before	Coordinate type before transforming
coord_type_after	Coordinate type after transforming

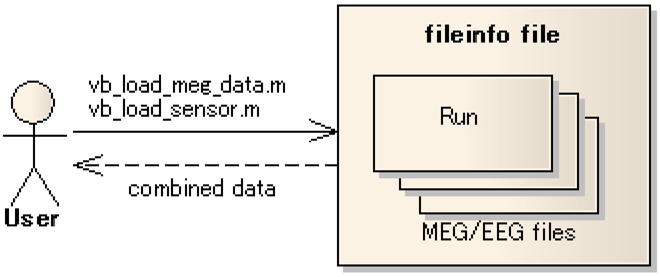
3.3.1.2. EEGinfo.device_info.Header

field	commentary
FILE	<struct> File information that was made by Biosemi
FileName	Biosemi file name
VERSION	Version code ('BIOSEMI')
PID	Local patient identification
RID	Local recording identification
T0	List of parameters to made Y2K compatible
HeadLen	Size of header [byte]
NRec	Recording time [sec]

Dur	Duration of data record
NS	Number of signals
Label	List of signal labels {*NS* x 1}
Transducer	List of type of each signal ('Active Electrode' etc.)
PhysDim	List of parameters for calibration [*NS* x 1]
PhysMin	List of parameters for calibration [*NS* x 1]
PhysMax	List of parameters for calibration [*NS* x 1]
DigMin	List of parameters for calibration [*NS* x 1]
DigMax	List of parameters for calibration [*NS* x 1]
PreFilt	List of pre-filters ('HP: DC; LP: 208 Hz' etc.)
SPR	Samples per data record
Cal	List of values to calibrate [*NS* x 1]
Off	List of offset to calibrate [*NS* x 1]
Calib	Matrix for calibration
SampleRate	List of sampling frequencies [*NS* x 1] (*SPR* / *Dur*) [Hz]
Chan_Select	List of flags to be selected channels
ChanTyp	List of channel type
AS	<struct>
AS.spb	Samples per block, AS.IDX2 : a list of index

4. Fileinfo file

Normally, when measuring data of multiple runs in one experiment, data is divided into multiple files. In such a case, when analyzing it, we will create one large file that actually combines the runs. However, this method consumes a lot of storage, so Fileinfo file provides a way to do it virtually. **Sensor information is averaged through the multiple runs. Please check the sensor information of the MEG / EEG file individually before combining. Do not combine run data whose sensor information is largely different**.



4.1. How to create fileinfo file.

Fileinfo file is created using vb_get_multifile_info.m. Specify the filename of MEG/EEG file imported with VBMEG by cell array as datafiles. The extension of fileinfo file should be '.meg.mat' for MEG or '.eeg.mat' for EEG.

```
vb_get_multi_fileinfo(datafiles, fileinfofile);
```

datafiles : cell array of multiple data file names

fileinfofile: output file name to save multiple file information 'fileinfo'

4.2. Exclude Bad channels and Trials

Measurement data may contain bad data. It is caused by hardware deficiencies, subject's body movements, picking up externally generated noise, and so on. I will explain the procedure to exclude such a bad channels / bad trial data from current source estimation.

4.3. Active channel information(fileinfo.ActiveChannel)

4.3.1. Load MEG/EEG data

data = vb_load_meg_data(fileinfofile);

4.3.2. Check Channel data

data(n, :, :); % n-channel

4.3.3. Load fileinfo and set Active channel (0:Invalid, 1:Valid)

load(fileinfofile, 'fileinfo');

fileinfo.ActiveChannel(n) = 0 or 1;

4.3.4. Update fileinfo file

vb_save(fileinfofile, ‘fileinfo’);

4.4. Active Trial information(fileinfo.ActiveTrial)

4.4.1. Load MEG/EEG data

data = vb_load_meg_data(fileinfofile);

4.4.2. Check trial data

data(:, :, n); % n-trial

4.4.3. Load fileinfo and set Active Trial flag(0:Invalid, 1:Valid)

load(fileinfofile, ‘fileinfo’);

fileinfo.ActiveTrial(n) = 0 or 1;

4.4.4. Update fileinfo file

vb_save(fileinfofile, ‘fileinfo’);

4.5. Fileinfo Format

Measurement	Measurement type code : ‘INFO’
fileinfo	Structure with the fileinformation for multiple files.
fileinfo.filename	List of session data filename {1xNsession}
fileinfo.Nchannel	The number of total channels.
fileinfo.Nsample	The number of samples in one trial.
fileinfo.Ntotal	The number of all trials.
fileinfo.Ntrial	The number of trials for each session [1 x Nsession]
fileinfo.session_id	Session index for each trial [1xNtotal]
fileinfo.cond_id	Condition number for each trial [1xNtotal]
fileinfo.ActiveChannel	Active channel list(boolean array) [Nchannel x 1]
fileinfo.ActiveTrial	Active trial list (boolean array) [Ntotal x 1]

5. Load functions

Usages of these functions are essentially the same for the MEG and EEG data files. Here, only basic usages for load functions are explained. These usages can be applied for both the minimum and standard formats. Please see help information of each function for advanced usages, which require the standard format in some cases.

All load functions below have one input argument, which is a file name of MEG/EEG data (`.meg.mat` or `.eeg.mat` file).

5.1. vb_load_meg_data()

```
[data, ch_info] = vb_load_meg_data(data_file);
```

field	commentary
Active	Boolean list that shows each channel is active or not
Name	Label list for each channel
Type	Type list for each channel
ID	Original index list for each channel

The first return value *data* is a MEG/EEG channel data whose size is [*Nchannel* x *Nsample* x *Nrepeat*].

The second return value *ch_info* is a channel information struct with the following fields.

5.2. vb_load_meg_info()

```
[info] = vb_load_meg_info(data_file);
```

The return value *info* is a minimum set of measurement information.

Fields are as follows.

field	commentary
SampleFreq	Sampling frequency [Hz]
Nchannel	Number of channels
Nsample	Number of time samples
Nrepeat	Number of trials
Pretrigger	Number of time samples before trigger onset
Measurement	Measurement type code
device	Device name ('YOKOGAWA', 'BIOSEMI', or 'NEUROMAG')
sensor_weight	(MEG) Sensor weight for each channel [*Nchannel*×*Nsensor*]
Coord	(EEG) (*x*, *y*, *z*) coordinates of each channel [*Nchannel*×3] [m]

5.3. vb_load_sensor()

```
[pick, Qpick, CoilWeight, Vcenter] = vb_load_sensor(data_file);
```

Meaning of each return value is as follows.

return value	commentary
pick	(*x*, *y*, *z*) coordinates of each sensor [*Nsensor*×3] [m]

Qpick	(*x*, *y*, *z*) direction of each sensor [*Nsensor*×3] (vector with unit norm) It is empty for EEG data
CoilWeight	Sensor weight for each channel [*Nchannel*×*Nsensor*] It is empty for EEG data
Vcenter	Center coordinate of a spherical model [1×3] [m] If this information is not stored in the data file, it is empty.

5.4. *vb_load_channel_info()*

```
[ch_info] = vb_load_channel_info(data_file);
```

The return value *ch_info* is a channel information struct.

This is the same as the second return value of *vb_load_meg_data()*.

field	**commentary**
Active	Boolean list that shows each channel is active or not
Name	Label list for each channel
Type	Type list for each channel
ID	Original index list for each channel
PhysicalUnit	(EEG) List of physical units

5.5. *vb_load_channel_pos()*

```
[pos] = vb_load_channel_pos(data_file);
```

The return value *pos* is (*x*, *y*, *z*) coordinates of each channel [*Nchannel*×3].

For EEG data, this is the same as *pick* returned by *vb_load_sensor()*.

For MEG data, one of the sensor positions is selected for each channel.

This function is used for plotting purpose. To calculate leadfield, you should use *vb_load_sensor()*.