

Standard data format MEG/EEG

ATR Neural Information Analysis Laboratories (June 29, 2012)

Contents

Revision History	3
1. Introduction.....	4
1.1. Conventions in this document	4
2. MEG	5
2.1. Minimum MEG-MAT format.....	5
2.1.1. Primary variables.....	5
2.1.2. Measurement information variables	5
2.1.3. Calculation of the leadfield and <i>sensor_weight</i>	6
2.2. Standard MEG-MAT format.....	6
2.2.1. Primary variables.....	6
2.2.2. Measurement information variables	6
2.3. Device-dependent variables and fields.....	8
2.3.1. <i>MEGinfo.device_info</i>	9
3. EEG.....	10
3.1. Minimum EEG-MAT format.....	10
3.1.1. Primary variables.....	10
3.1.2. Measurement information variables	10
3.2. Standard EEG-MAT format	10
3.2.1. Primary variables.....	11
3.2.2. Measurement information variables	11
3.3. Device dependent variables and fields.....	13
3.3.1. <i>EEGinfo.device_info</i>	13
4. Load functions	15
4.1. <i>vb_load_meg_data()</i>	15
4.2. <i>vb_load_meg_info()</i>	15
4.3. <i>vb_load_sensor()</i>	15
4.4. <i>vb_load_channel_info()</i>	16
4.5. <i>vb_load_channel_pos()</i>	16

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)

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

2.1.2. Measurement information variables

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

2.1.2.1. MEGinfo

field	commentary
<i>Measurement</i>	Measurement type code 'MEG' or 'EEG'
<i>device</i>	Device name E.g. 'YOKOGAWA', 'BIOSEMI', or 'NEUROMAG'
<i>Nchannel</i>	Number of channels in MEG data
<i>Nsample</i>	Number of time samples in MEG data
<i>Nrepeat</i>	Number of trials in MEG data
<i>Pretrigger</i>	Number of samples before trigger onset
<i>SampleFreq</i>	Sampling frequency [Hz]
<i>sensor_weight</i>	Sensor weight for each channel [<i>Nchannel</i> × <i>Nsensor</i>]

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 $\text{pick}(m, :)$ and direction $\text{Qpick}(m, :)$. It is also necessary to know the j -th cortical vertex's position $V(j, :)$ and its current direction $\text{xx}(j, :)$ stored in a `.brain.mat` file. Then, the leadfield at the n -th channel is given as

$$G_{\text{channel}}(n, j) = \sum_m \text{sensor_weight}(n, m) \cdot G_{\text{sensor}}(m, j)$$

or

$$G_{\text{channel}} = \text{sensor_weight} \cdot G_{\text{sensor}}$$

in matrix notation.

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

$$\text{sensor_weight}(n, m) = \begin{cases} 1 & \text{for } n = m \\ 0 & \text{otherwise} \end{cases} .$$

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

$$\text{sensor_weight}(n, 2n-1) = -1 ,$$

$$\text{sensor_weight}(n, 2n) = 1 , \text{ and}$$

$$\text{sensor_weight}(n, m) = 0 \text{ for } m \neq 2n-1, 2n .$$

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
<i>bexp</i>	MEG measurement signals [$N_{\text{channel}} \times N_{\text{sample}} \times N_{\text{repeat}}$] [Tesla]
<i>bexp_ext</i>	External channel measurement signals [$N_{\text{channel_ext}} \times N_{\text{sample}} \times N_{\text{repeat}}$]
<i>pick</i>	(x, y, z) coordinates of each sensor [$N_{\text{sensor}} \times 3$] [m]
<i>Qpick</i>	(x, y, z) direction of each sensor [$N_{\text{sensor}} \times 3$] (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
<i>Measurement</i>	Measurement type code

	'MEG' or 'EEG'
<i>CoordType</i>	Coordinate system for sensor position 'SPM_Right_m'
<i>MEGinfo</i>	<struct> Measurement information of data

2.2.2.1. MEGinfo

field	commentary
<i>Measurement</i>	Measurement type code 'MEG' or 'EEG'
<i>device</i>	Device name such as 'YOKOGAWA', 'BIOSEMI', and 'NEUROMAG'
<i>Nchannel</i>	Number of channels in MEG data
<i>Nsample</i>	Number of time samples in MEG data
<i>Nrepeat</i>	Number of trials in MEG data
<i>Pretrigger</i>	Number of time samples before trigger onset
<i>SampleFreq</i>	Sampling frequency [Hz]
<i>sensor_weight</i>	Sensor weight for each channel [$Nchannel \times Nsensor$]
<i>MEGch_id</i>	List of original index of channel [$Nchannel \times 1$]
<i>MEGch_name</i>	List of original label of channel [$Nchannel \times 1$]
<i>ActiveChannel</i>	Active channel list (boolean array) [$Nchannel \times 1$]
<i>ActiveTrial</i>	Active trial list (boolean array) [$Nrepeat \times 1$]
<i>Vcenter</i>	Center coordinate of a spherical model [1×3] [m]
<i>Vradius</i>	Radius of a spherical model [m]
<i>MEG_ID</i>	MEG data ID
<i>MRI_ID</i>	ID for MRI image on which the sensor position is registered
<i>Trial</i>	<struct array> Information of each trial [$Nrepeat \times 1$]
<i>ChannelInfo</i>	<struct> Information of MEG channels
<i>ExtraChannelInfo</i>	<struct> Information of extra channels
<i>saveman</i>	<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
<i>data_dir</i>	Relative path from a <code>.meg.mat</code> file to a directory where binary data file is stored
<i>precision</i>	Precision of a binary data: 'float64'

2.2.2.2. Substructure of *MEGinfo*

MEGinfo.ChannellInfo

field	commentary
<i>ID</i>	Original index list for each channel [$N_{channel} \times 1$]
<i>Name</i>	Label list for each channel [$N_{channel} \times 1$] cell array
<i>Type</i>	Type list for each channel [$N_{channel} \times 1$] cell array
<i>Active</i>	Boolean (0 or 1) list of active channel [$N_{channel} \times 1$]

MEGinfo.ExtraChannellInfo

field	commentary
<i>gain</i>	<struct> Struct array that has two fields, name and value
<i>Channel_id</i>	Original index list for each channel [$N_{channel_ext} \times 1$]
<i>Channel_name</i>	Label list for each channel [$N_{channel_ext} \times 1$] cell array
<i>Channel_type</i>	Type list for each channel [$N_{channel_ext} \times 1$] cell array
<i>Channel_active</i>	Boolean (0 or 1) list of active channel [$N_{channel_ext} \times 1$]

MEGinfo.Trial

This field is a struct array. Its size is [$N_{repeat} \times 1$]. It has following fields.

field	commentary
<i>number</i>	Index number for this trial
<i>sample</i>	List of time sample index for this trial
<i>Active</i>	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
<i>ref_pick</i>	coordinates of each reference sensor [$n_{ref} \times 3$]
<i>ref_Qpick</i>	normal vectors of each reference sensor [$n_{ref} \times 3$]
<i>PositionFile</i>	File path for the sensor position registration on MRI by our positioning program (.pos.mat) This file is used for importing sensor position by <code>vb_job_meg.m</code> . If this variable is empty, there is no such file.
<i>MEGinfo.device_info</i>	Device-dependent variables in <i>MEGinfo</i> are included into <i>device_info</i> struct.

2.3.1. *MEGinfo.device_info*

field	commentary
<i>sensor_weight_ref</i>	List of weight of each reference sensor [$n_{ref} \times n_{ref}$]
<i>acq_type</i>	Type of data acquisition 'Continuous_Raw'
<i>acq_info</i>	<struct> Data acquisition information
<i>TransInfo</i>	<struct> Information of a coordinate transformation
<i>index_in_fiffile</i>	(neuromag) Index number in fiffile
<i>Comment</i>	(neuromag) Comment of this data

2.3.1.1. *MEGinfo.device_info.TransInfo*

field	commentary
<i>trans_mri</i>	Coordinate transform matrix to MRI
<i>coord_type_before</i>	Coord type before transformation
<i>coord_type_after</i>	Coord type after transformation by the <i>trans_mri</i> matrix

2.3.1.2. *MEGinfo.device_info.acq_info*

field	commentary
<i>data_file</i>	Imported data file path
<i>data_bit_len</i>	Imported data bit length (16)
<i>condition</i>	<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
<i>eeg_data</i>	EEG channel data [<i>Nchannel</i> x <i>Nsample</i> x <i>Nrepeat</i>] [V]

3.1.2. Measurement information variables

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

3.1.2.1. *EEGinfo*

field	commentary
<i>Measurement</i>	Measurement type code: 'MEG' or 'EEG'
<i>Device</i>	Device name such as 'BIOSEMI'
<i>Nchannel</i>	Number of channels in EEG data
<i>Nsample</i>	Number of time samples in EEG data
<i>Nrepeat</i>	Number of trials in EEG data
<i>Pretrigger</i>	Number of time samples before trigger onset
<i>SampleFrequency</i>	Sampling frequency [Hz]
<i>Coord</i>	(<i>x</i> , <i>y</i> , <i>z</i>) coordinates of EEG channels [<i>Nchannel</i> 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
<i>eeg_data</i>	EEG channel data [<i>Nchannel</i> x <i>Nsample</i> x <i>Nrepeat</i>] [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
<i>Measurement</i>	Measurement type code: 'MEG' or 'EEG'
<i>EEGinfo</i>	<struct> Measurement information of data

3.2.2.1. EEGinfo

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

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

EEGinfo.ExtraChannellInfo

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

EEGinfo.File

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

EEGinfo.Trial (array struct)

field	commentary
<i>number</i>	Index number of this trial
<i>sample</i>	List of sample index number for this trial
<i>Active</i>	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
<i>Reference</i>	Reference method to decide coordinates
<i>RecordTime</i>	Recording time [sec]
<i>TransInfo</i>	<struct> Information of a coordinate transformation
<i>Header</i>	<struct> Biosemi header information
<i>History</i>	List of histories of this file
<i>Version</i>	Version of this struct

3.3.1.1. *EEGinfo.device_info.TransInfo*

field	commentary
<i>trans_mri</i>	Matrix to transform coordinates
<i>coord_type_before</i>	Coordinate type before transforming
<i>coord_type_after</i>	Coordinate type after transforming

3.3.1.2. *EEGinfo.device_info.Header*

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

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

4. 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).

4.1. *vb_load_meg_data()*

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

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.

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

4.2. *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
<i>SampleFreq</i>	Sampling frequency [Hz]
<i>Nchannel</i>	Number of channels
<i>Nsample</i>	Number of time samples
<i>Nrepeat</i>	Number of trials
<i>Pretrigger</i>	Number of time samples before trigger onset
<i>Measurement</i>	Measurement type code
<i>device</i>	Device name ('YOKOGAWA', 'BIOSEMI', or 'NEUROMAG')
<i>sensor_weight</i>	(MEG) Sensor weight for each channel [<i>Nchannel</i> × <i>Nsensor</i>]
<i>Coord</i>	(EEG) (<i>x</i> , <i>y</i> , <i>z</i>) coordinates of each channel [<i>Nchannel</i> ×3] [m]

4.3. *vb_load_sensor()*

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

Meaning of each return value is as follows.

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

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

4.4. *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
<i>Active</i>	Boolean list that shows each channel is active or not
<i>Name</i>	Label list for each channel
<i>Type</i>	Type list for each channel
<i>ID</i>	Original index list for each channel
<i>PhysicalUnit</i>	(EEG) List of physical units

4.5. *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()*.