

# User Reference:EEGLabImport

From BCI2000 Wiki

## Contents

- 1 Synopsis
- 2 Prerequisites
- 3 Installing EEGLabImport
- 4 Using EEGLabImport
- 5 Tutorial
  - 5.1 Load the TestData.dat Data Set
  - 5.2 Import the Channel Location File
  - 5.3 Extracting Epochs
  - 5.4 Extracting Task-Specific Conditions
  - 5.5 Analyzing the Data
    - 5.5.1 Channel Spectra and Maps
    - 5.5.2 Channel ERP Analysis
  - 5.6 Further Information

## Synopsis

EEGLabImport is a plugin for EEGLAB that allows importing data, including the recorded signal and event markers, in the BCI2000 data format. Data from multiple runs can be concatenated and imported simultaneously, greatly extending and simplifying the analysis of EEG and ECoG based data sets. Further information about EEGLAB can be found at: <http://scn.ucsd.edu/eeglab/>.

## Prerequisites

BCI2000import requires Matlab 7.0 or greater, with a current version of EEGLAB. Depending on the size of the data sets, a large amount of RAM may also be required, due to resource requirements of Matlab. 512 MB is probably the minimum that allows the system to be usable.

## Installing EEGLabImport

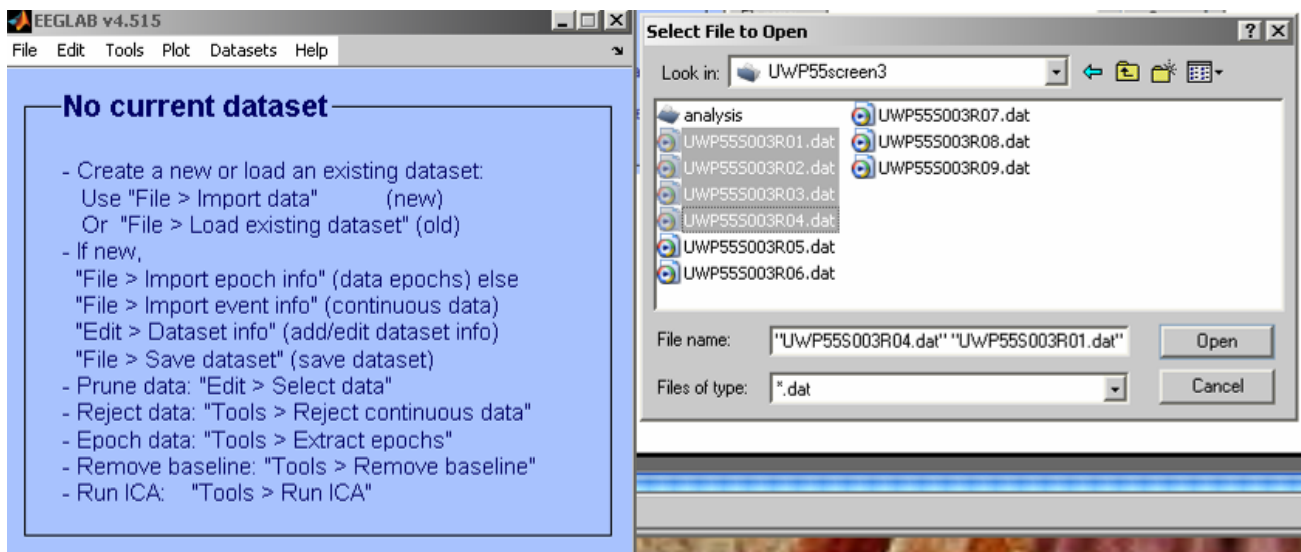
Installation of the plugin is straightforward, but may require compiling the `load_bcidat` MEX function for your platform.

1. If you are using a 32-bit version of Windows, `load_bcidat.mexw32` is likely already compiled for your platform, and can be located in the `BCI2000/tools/mex/` folder. Copy this file to the `BCI2000/tools/EEGLabImport` folder, and skip to 5.
2. If you are using a platform for which a version of `load_bcidat` does not exist in the `/tools/mex` folder, the `load_bcidat` MEX file must be compiled.
3. Open Matlab, and navigate to `BCI2000/src/core/Tools/mex/`. Type `buildmex` at the command line; this will build the BCI2000 MEX functions for your platform.
4. Once the build process is complete, locate the `load_bcidat` file for your platform based on its extension (see the Matlab MEX documentation for more details), and copy it to `BCI2000/tools/EEGLabImport`.
5. Copy the folder `BCI2000/tools/EEGLabImport` to the EEGLab plugin folder, located at `eeglab/plugins`.

When EEGLAB is started, the plugin will be automatically detected and installed.

## Using EEGLabImport

Begin a Matlab session, and start EEGLAB by typing `eeglab` at the command prompt. EEGLabImport is detected and added to the menus automatically. Go to `File->Import Data->BCI2000 Data` to start the data import program. A file-select dialog is presented, in which multiple BCI2000 data files can be selected (see figure below). The default file extension for BCI2000 data files is `*.dat`. Select one or more files from the desired folder, and press Open.



After pressing Open, the program loads the selected data, including the signal, states, and testing parameters such as the sampling rate. Next, a window will appear asking which BCI2000 states should be imported as EEGLAB events. Several important and common states will automatically be selected, but it is not required to import these. These events are used to generate signal epochs, which are then used for signal averaging and other more complex analyses. Each event relates to a different experimental event, some of which are described below.

- **TargetCode** – This state gives the target number that was displayed to the subject during each trial. In cases where only 2 targets are presented, this will have a value of either 1 or 2. EEGLAB stores the TargetCode as a single event, and does not discriminate between different values. Therefore, once the data is imported, it is necessary to create new, separate events for each TargetCode value in EEGLAB using the Edit->Select Epochs/Events menu. Refer to the EEGLAB documentation for more information.
- **ResultCode** – This is related to the TargetCode state, but tells how the user actually performed on the task. If the user was able to move the cursor to the correct Target, then the ResultCode is the same as the TargetCode for that trial. Otherwise, it will be different. This is useful if you only want to select trials that result in a hit.
- **Feedback** – This event gives the time at which the user is first presented with visual feedback of the cursor. This often does not coincide with the appearance of the target, and you may want to average on the Feedback state instead of TargetCode.
- **IntertrialInterval** – This event gives the period of time in between trials, and may be useful to determine a baseline state of the signal.

Once the desired events are selected, EEGLABImport removes the undesired states, and finishes importing the data into EEGLAB.

The final step is to give the dataset a name, and to optionally save the file in the EEGLAB format for future analysis:



## Tutorial

This tutorial covers some of the basics of using EEGLAB to analyze data from a BCI experiment. The sample data set used for this is included in the EEGLABImport folder, and is called TestData.dat. This data set consists of a BCI cursor control session in which the 10-Hz component of one channel was modulated with a function generator to simulate alpha brain activity during a BCI task. The amplitude of the 10-Hz component was translated into vertical cursor velocity, as in a real BCI cursor task.

### Load the TestData.dat Data Set

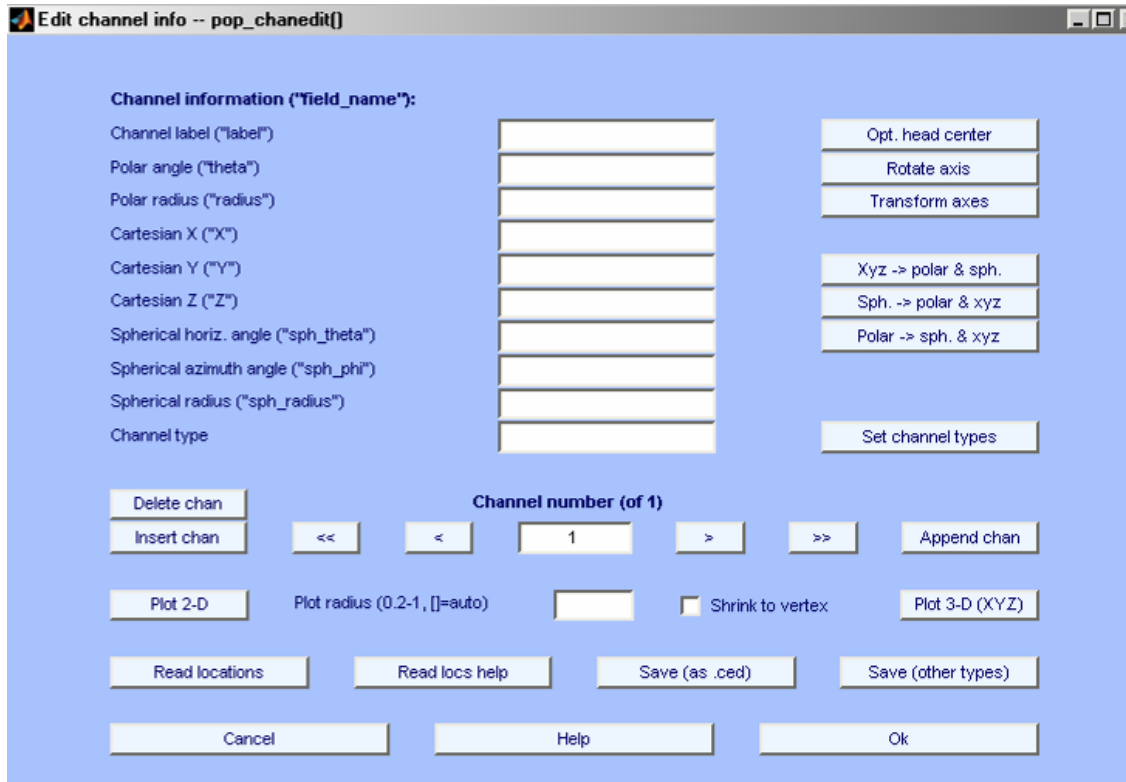
Start EEGLAB in Matlab. Go to File->Import Data->BCI2000 Data, and find the TestData.dat file (which should be in the EEGLABImport folder, or the eeglab/sample\_data folder). Select the following BCI events to import:

- TargetCode
- ResultCode
- Feedback
- IntertrialInterval

These events are needed to create epochs in EEGLAB. When asked, give the dataset the name "data1." (Any name can be used, but the tutorial will refer to data1 from here on).

### Import the Channel Location File

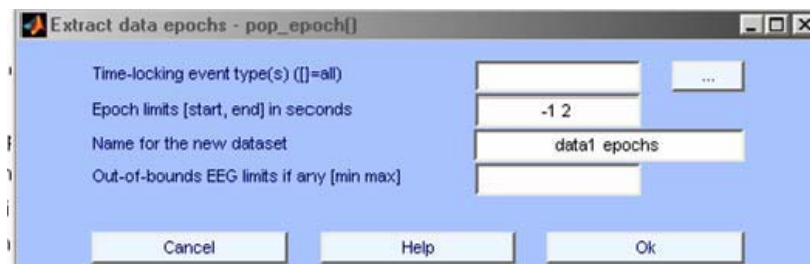
In order to utilize many of EEGLAB's functions, it needs to know where each channel is located physically on the head. This is accomplished by loading a channel location file. An example file with locations for 16 channels is included with the sample data set, and is called Cap17.locs. To import this file, in EEGLAB go to Edit->Channel Locations, which brings up the channel location edit window:



Press the Read Locations button in the lower-left hand corner, and select the Cap17.locs file in the eeglab/plugins/EEGLabImport/ folder. A window appears asking for the File Format; select "autodetect", and press OK. The channel locations are imported, and you can view the information for each electrode. The important electrode for this data set is number 5, which was imported as the C3 electrode. Press the OK button to close the channel edit window.

## Extracting Epochs

The next step is to break the data set down into epochs based on one or more events that were imported. The BCI task can be epoched in many ways, but the way it will be done here is by using the TargetCode as the event marker. To begin, in EEGLAB, go to Tools->Extract Epochs. There are several important values to be set in this window:

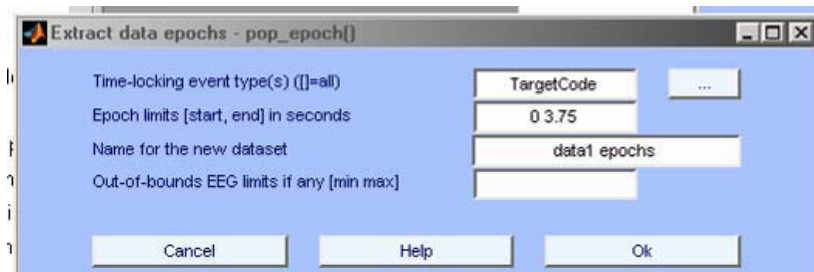


The first and most important value is the time-locking event, which is the event used to create epochs. In this example, we are using the TargetCode event. This can be typed in, or pressing the [...] button will show all available events.

The next value to set is the epoch limits. This determines the time ranges to keep before and after each event occurs.

**Note:** These values *will* be different depending on how the experiment was setup, based on the pretrial pause and the duration of each trial. In this data set, a pretrial pause of 1.25 s was used, and each trial lasted 2.5 s. Therefore, a time range of 0 to 3.75 seconds (1.25 s + 2.5 s) will be used; the time from 0 to 1.25 s will later be specified as the baseline range. Type 0 3.75 in the Epochs limits box.

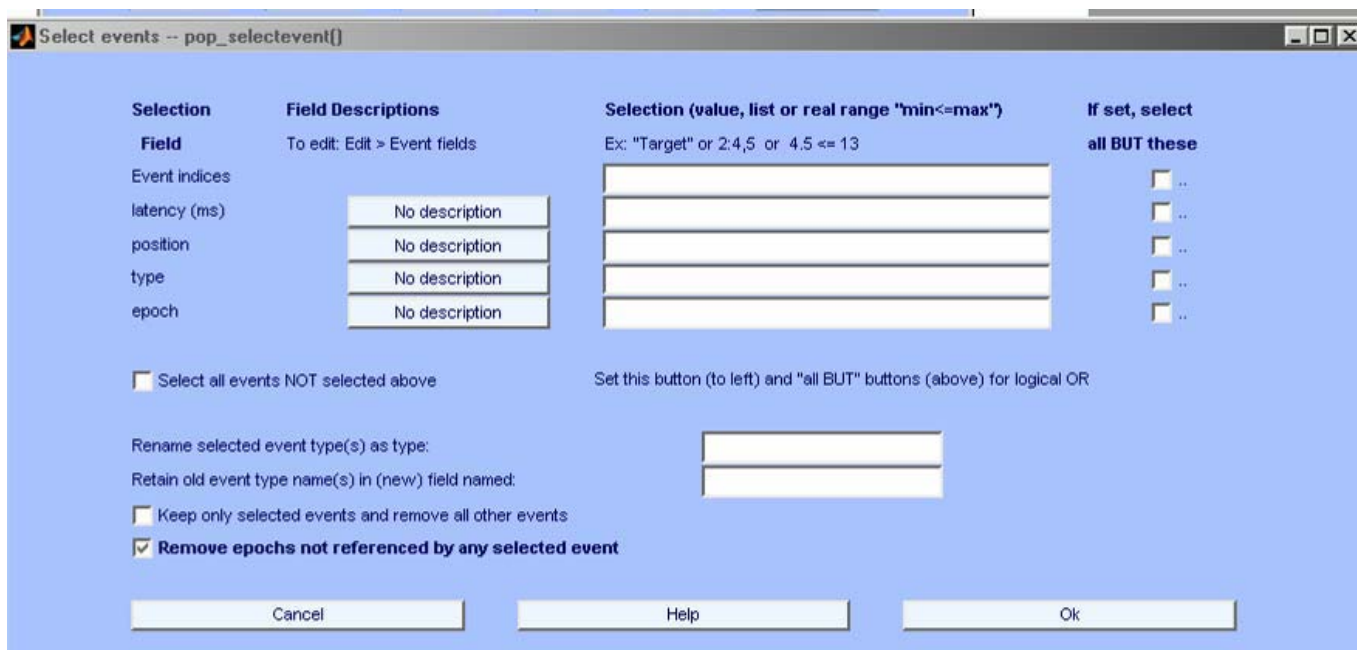
Finally, the name of the new dataset should be specified in the 3rd box. The default value of data1 epochs is fine. The window should appear like this now:



Press the OK button to finish. When asked for a new dataset name, make sure it is "data1 epochs", and press OK. A new dataset with that name is created and activated. An important note to make here is that EEGLAB does not write over old datasets when operations are performed, but creates an entirely new set. Therefore, the set we just created is given an index of 2; the currently selected dataset can be changed in EEGLAB in the Datasets menu. The final box presented is the baseline removal box, which specifies the region of the epoch which is baseline. For this, enter 0 1.25, to specify the baseline from time 0 to 1.25 s.

## Extracting Task-Specific Conditions

In the BCI cursor control task, there are actually several conditions that can be observed based on the value of the TargetCode. Extracting the epochs in Step 3 created a dataset based on all values of TargetCode, while we actually want to separate the different conditions. This is done easily in EEGLAB by going to the Edit->Select Epochs/Events menu:



There are two important fields for us to use: the position and the type fields. The type field is where you specify the event that you want to select, which is the TargetCode (enter this into the type field). The position is the value of the TargetCode that you want to keep; we will be creating data sets for both TargetCodes used in the BCI task, 1 and 2. For this step, enter 1 into the position field, and press OK. It will ask you to confirm deleting some of the epochs, which is what we want, so press OK. Next, a new window appears asking for the Dataset name; enter data1 epochs T1, since the TargetCode with a value of 1 was kept. This creates yet another new dataset, this with an index of 3.

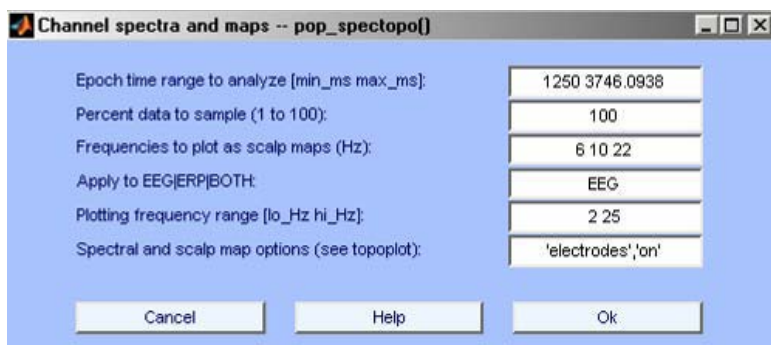
Now, we need to create another dataset that contains the information for TargetCode 2. Remember, though, that the currently selected dataset removed all of this information, so we need to select a previous data set with ALL info. Go to Datasets->Dataset2: Data1 Epochs in eeglab. This selects the old epoched data. Now, repeat the above steps, going to Edit Select Epochs/Events, and enter TargetCode for type and 2 for position. Press Ok, confirm deleting the other epochs, and give the Dataset the name data1 epochs T2. We now have separate datasets for each condition, which can be analyzed separately, or compared to each other.

## Analyzing the Data

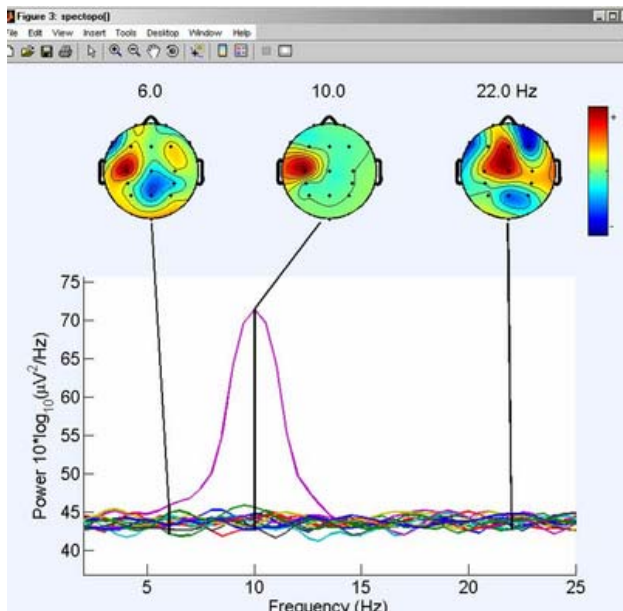
It would be impossible to go through every aspect of EEGLAB in this tutorial, so I will cover a few of the basic ones to get started.

### Channel Spectra and Maps

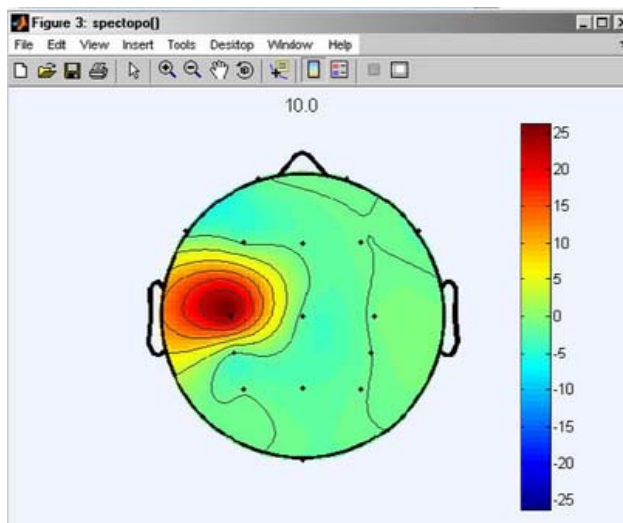
One of the simplest functions to use in EEGLAB and one of the most useful for BCI research is analyzing the spectral content of the data during different conditions. To look at the channel spectra for each condition, start by selecting the dataset for TargetCode1 (Datasets->Dataset3...). Go to Plot->Channel Spectra and Maps. This brings up a window with several options:



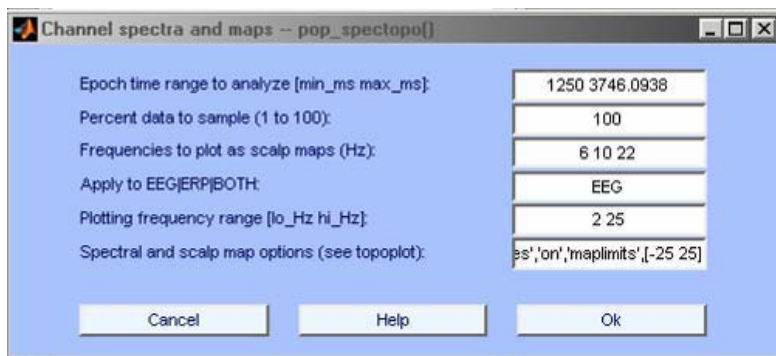
Because we are interested in what happened during the trial, change the start time to 1250 ms, and leave the end time the same; this omits the baseline activity before the cursor appears. Since this is a small dataset, set Percent data to sample to 100. Set the Frequencies to plot as scalp maps to '6 10 22' (this will become more clear shortly). Finally, in the last box, change the 'electrodes' option to 'on.' Press OK. The following plot is produced:



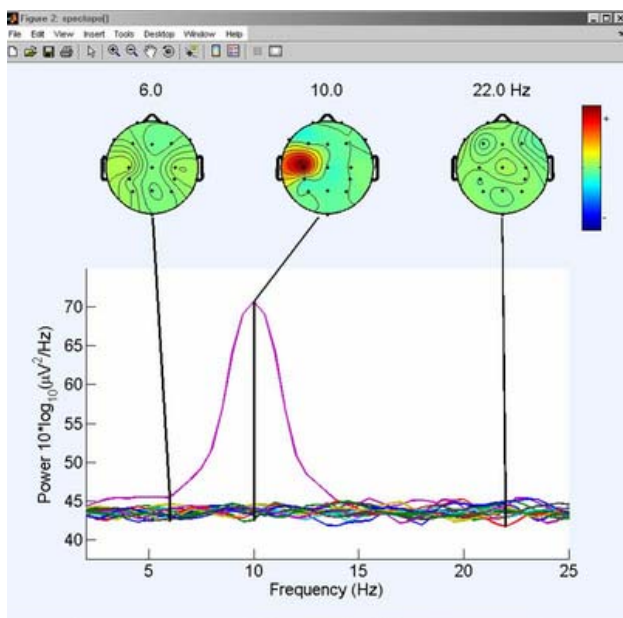
Note that in this figure, the scalp maps are **not** normalized to each other; the red in the 10 Hz figure is much larger than the red in the other two figures. To compensate for this, find out what the maximum range is by clicking on the scalp map, which will open a new figure:



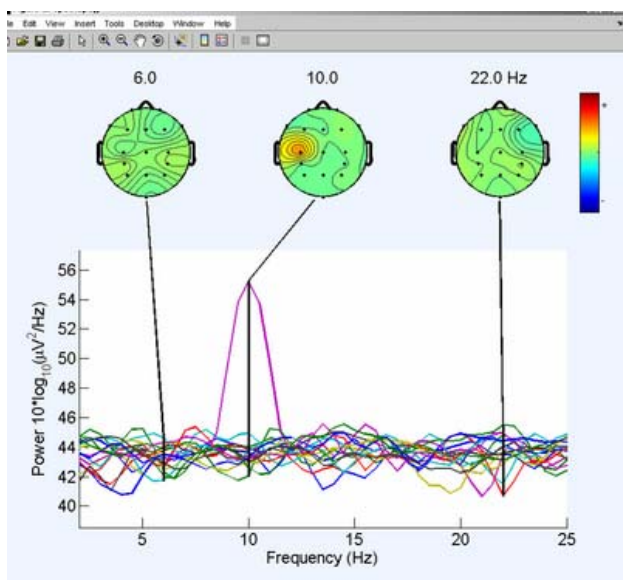
Press the Colorbar button (under the Window and Help menus) to find out what the range is in this figure (it is about -25 to +25). Now, go to Plot->Channel Spectra and Maps again, and enter the same options as before. Now, next to 'electrodes','on', add 'maplimits',[-25 25]:



Press OK, and the new normalized figure is produced:



One method of comparing between the two conditions would be look at the TargetCode 2 dataset using the same analysis and the same range (-25 to +25) as before, and seeing how the activation strength compares. The dataset4 was selected and analyzed exactly the same as before:



Notice that the 10 Hz activation is much weaker for TargetCode = 2 than for TargetCode = 1.

### Channel ERP Analysis

Another set of useful functions in EEGLAB are the ERP image functions. The EEGLAB tutorial covers these in-depth, so I will give a few quick examples here of how they might be used; refer to the EEGLAB tutorial for more information. Select Dataset 3 (TargetCode = 1), and go to Plot->Channel ERP Image:



**Channel ERP image -- pop\_erpimage()**

<b>Channel</b>	<input type="text" value="5"/>	<b>Figure title</b>	<input type="text"/>
<b>Smoothing</b>	<input type="text" value="1"/>	<input checked="" type="checkbox"/> Plot scalp map	
<b>Downsampling</b>	<input type="text" value="1"/>	<input checked="" type="checkbox"/> Plot ERP	ERP limits (uV) <input type="text"/>
<b>Time limits (ms)</b>	<input type="text" value="0"/> <input type="text" value="3746.0938"/>	<input checked="" type="checkbox"/> Plot colorbar	Color limits (see Help) <input type="text"/>

**Sort/align trials by epoch event values**

Epoch-sorting field <input type="text"/>	Event type(s) <input type="text"/>	Event time range <input type="text"/>	Rescale <input type="text" value="no"/>	Align <input type="text"/>	<input type="checkbox"/> Don't sort by value
					<input type="checkbox"/> Don't plot values

**Sort trials by phase**

Frequency (Hz   minHz maxHz)	Percent low-amp. trials to ignore	Window center (ms)	Wavelet cycles
<input type="text" value="10"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="3"/>

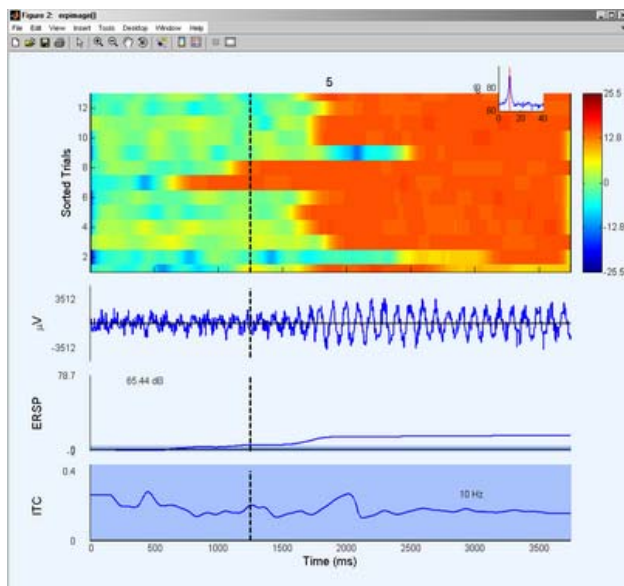
**Inter-trial coherence options**

Frequency (Hz   minHz maxHz)	Signif. level (<0.20)	Amplitude limits (dB)	Coher limits (<=1)	<input checked="" type="checkbox"/> Image amps (Requires signif.)
<input type="text" value="10"/> <input type="text" value="10"/>	<input type="text" value="0.01"/>	<input type="text" value="-3"/> <input type="text" value="3"/>	<input type="text"/>	

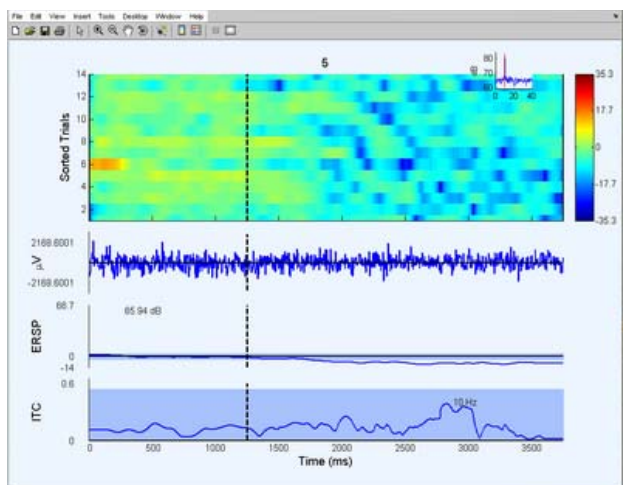
**Other options**

Plot spectrum (minHz maxHz)	Baseline ampl. (dB)	Mark times (ms)	More options (see >> help erpimage)
<input type="text" value="2"/> <input type="text" value="40"/>	<input type="text"/>	<input type="text" value="1250"/>	<input type="text"/>

To start, enter 5 into the channel box, 1 into smoothing, 1 into downsampling, and the default time range (0 to 3746...). To the right of these boxes, make sure that the Plot ERP and Plot Colorbar boxes are checked. Next, go to the Inter-trial coherence options, and under frequency enter 10; for the significance level enter 0.01, and on the right check the Image Amps box. Finally, at the bottom, enter 0 40 for the Plot Spectrum box, and 1250 for the Mark Times box. Press OK, and the following figure is produced:



This shows the amplitude for each trial, the mean response, the ERSP (event-related spectral power), and the inter-trial coherence for the condition that the TargetCode = 1. Compare this to TargetCode = 2 (change the dataset to #4, and enter the same parameters for the analysis in the ERP Image) as in this figure:



There is a very noticeable change in signal amplitude for each condition after the baseline condition, when the cursor appears and the control phase of the task begins.

### Further Information

This tutorial aimed at getting you started using EEGLAB for analyzing BCI2000 data. Detailed information on using the EEGLAB software is provided by the EEGLAB tutorial, available at the EEGLAB (<http://www.sccn.ucsd.edu/eeglab/>) homepage.

Retrieved from "[http://bci2000.org/wiki/index.php/User\\_Reference:EEGLabImport](http://bci2000.org/wiki/index.php/User_Reference:EEGLabImport)"

Categories: Matlab | External Interfaces

- This page was last modified 19:36, 5 March 2010.
- This page has been accessed 2,659 times.
- Privacy policy
- About BCI2000 Wiki
- Disclaimers