



# Practicum: Performing ERP analysis in BCILAB

EEGLAB Workshop 2016, Track B

Christian A. Kothe  
Intheon

Questions welcome any time! 😊



# Outline

1. Preparations
2. Offline ERP Analysis
3. Online ERP Analysis
4. Customizing Approaches
5. Optional analyses: Using an Advanced ERP Paradigm



# 1 Preparations



# Preparations

- ... please start unzipping your BCILAB distribution file to some directory on your disk if you haven't done so already.
- Should take 5-10 minutes (if you're on Windows, try to use 7-zip or WinZip/WinRAR).
- Don't put it inside the EEGLAB folder (BCILAB includes an EEGLAB distribution).

# System requirements

- MATLAB 2008a+
- 1GB+ RAM (better: 2GB+)
- Windows, Linux, or Mac
- For smooth workshop: **No** toolboxes in MATLAB path other than Mathworks toolboxes (or EEGLAB)
- To use certain additional features (not covered today):  
Signal Processing Toolbox, Statistics Toolbox
- To use certain advanced features (also not covered today):  
Correct MEX compiler setting (this requires Microsoft Visual C++ Express under Win64 and Xcode/gcc under Mac)

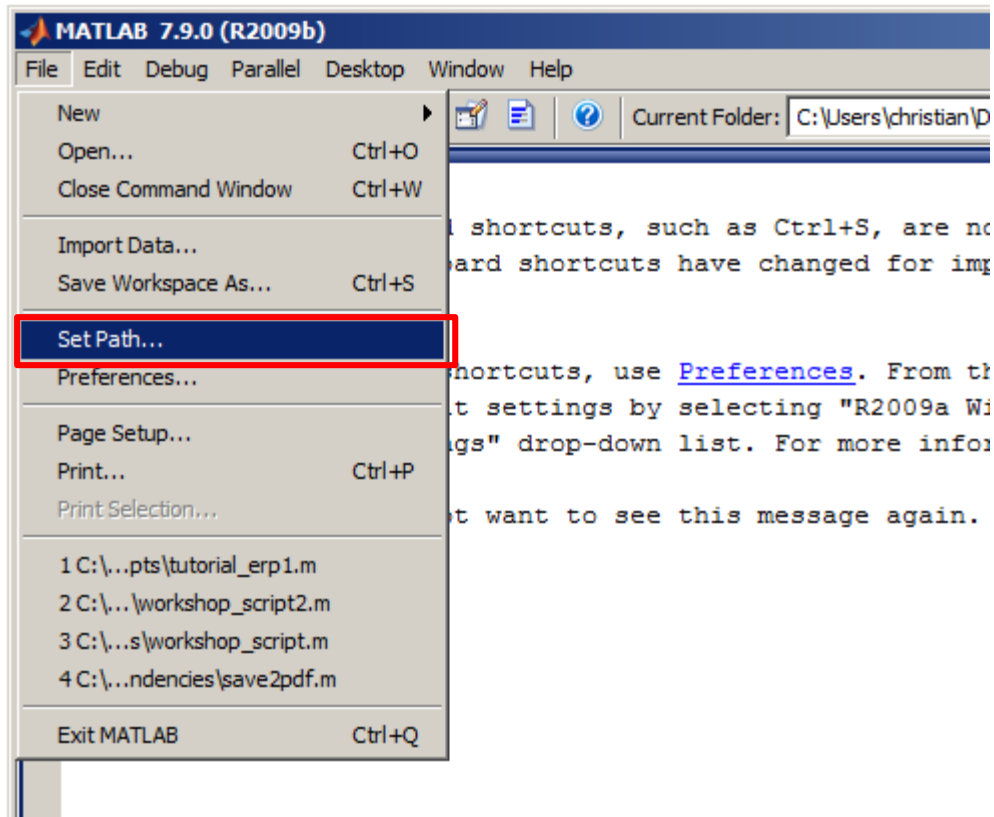


# Note: When Processing your own Data

- Note the following requirements:
    - You need proper channel labels (usually the 10-20 labels); 3d locations not necessary
    - You need event markers in your data at time points where the BCI should predict outputs
    - BCILAB needs raw (unprocessed) data
    - Make sure you have a file format supported by EEGLAB
- Rawr!
-

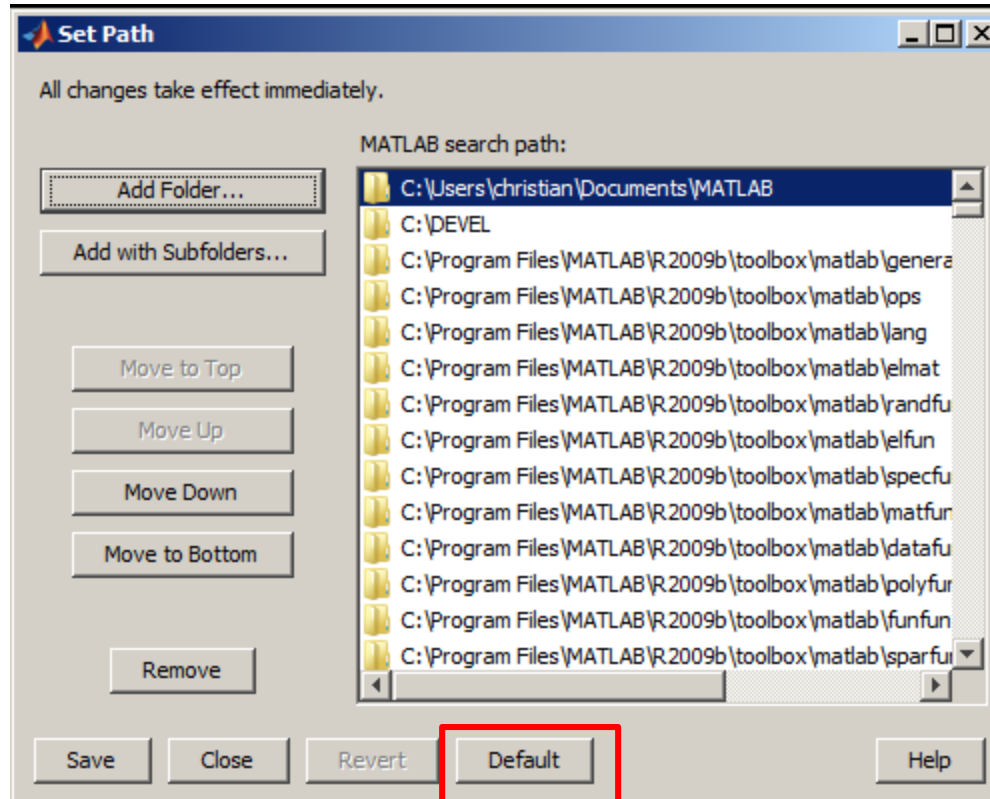


# Clearing the Path





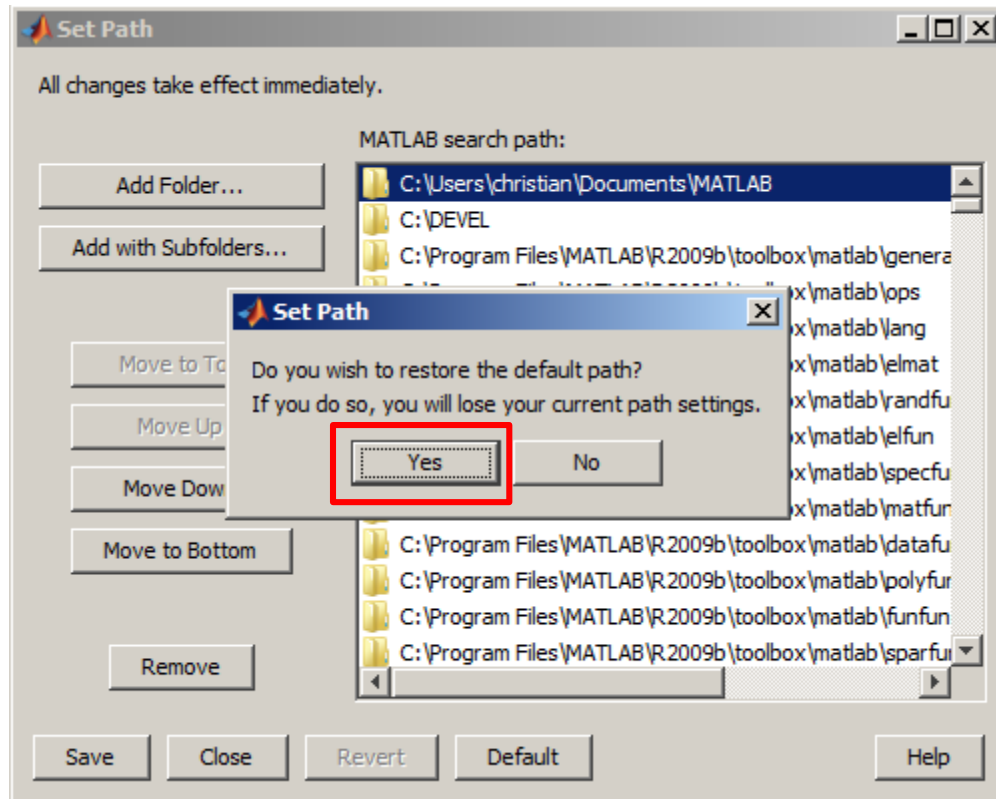
# Clearing the Path





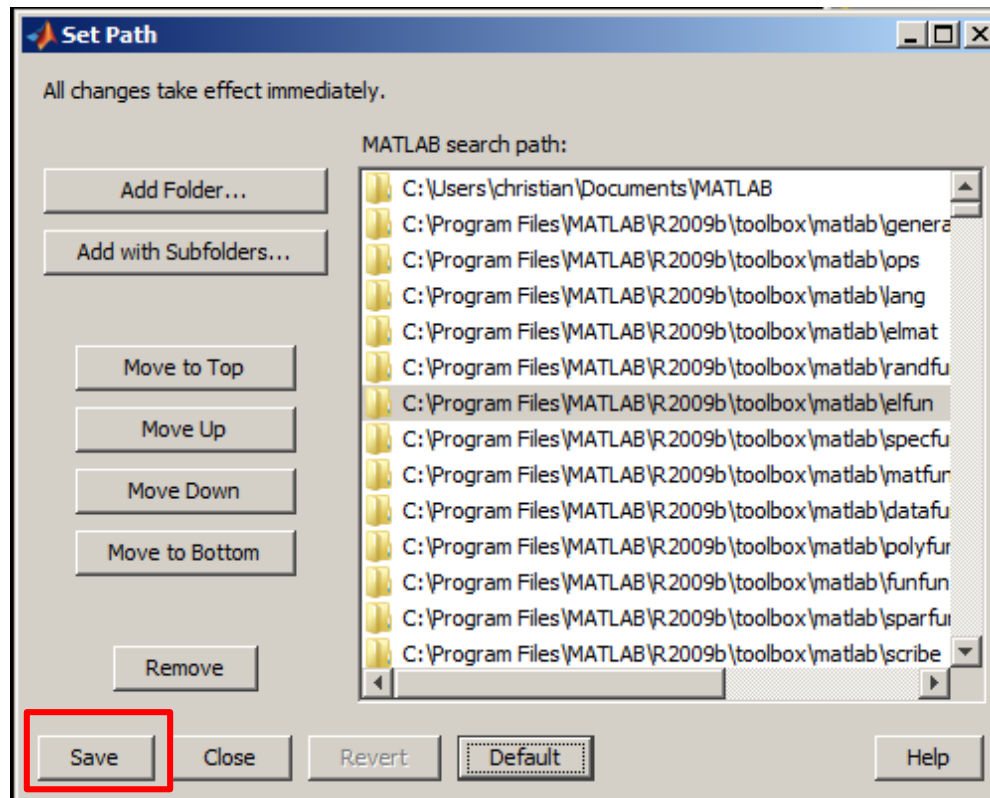


# Clearing the Path



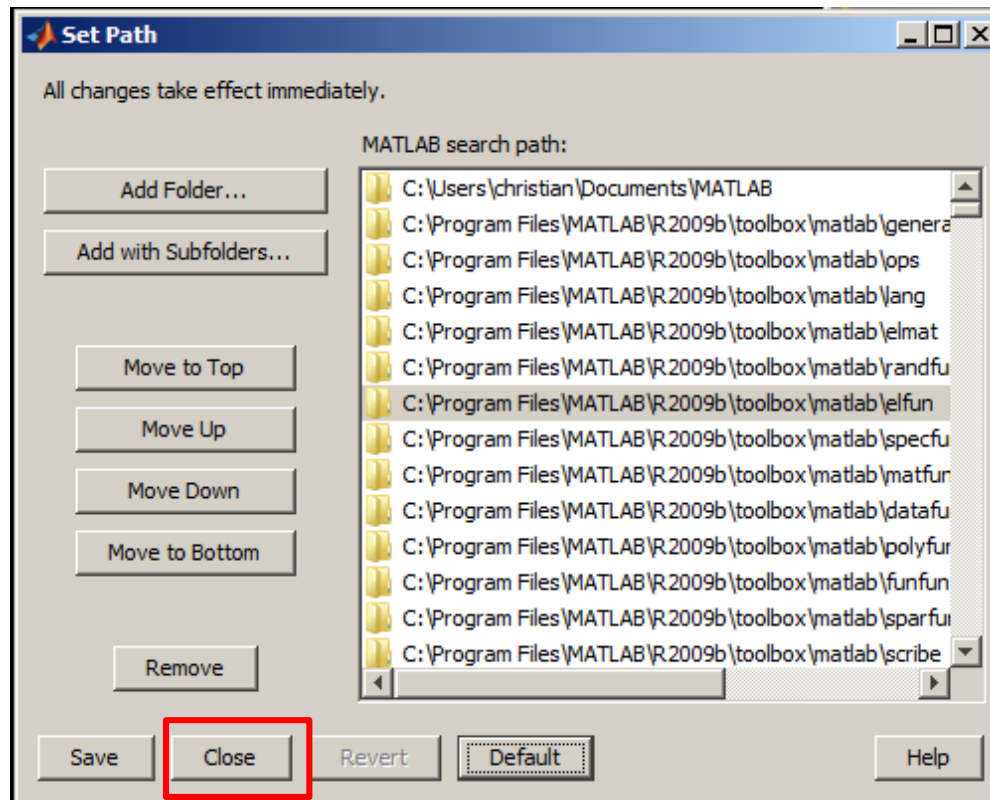


# Clearing the Path





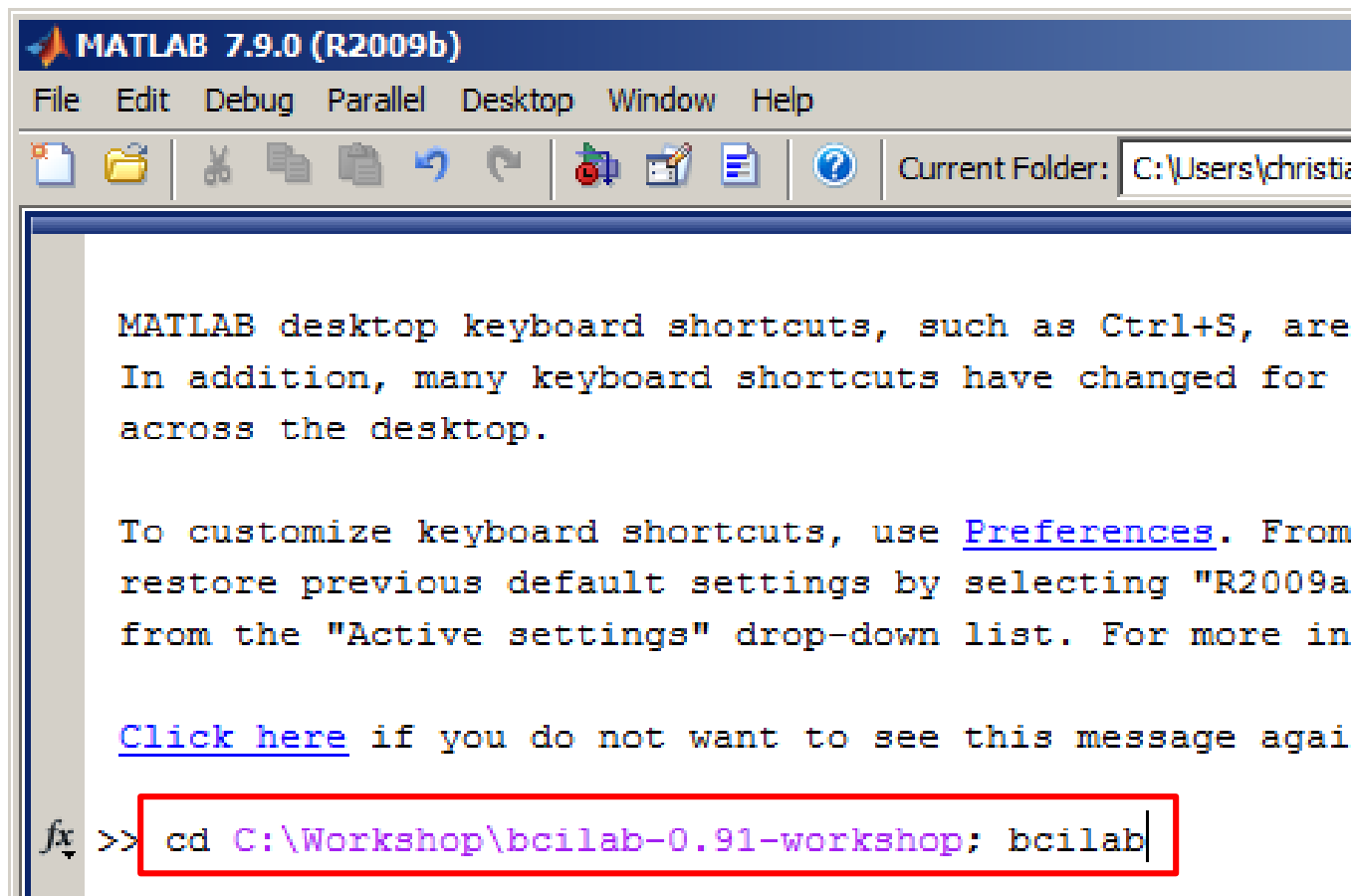
# Clearing the Path





# Starting the Toolbox

- Type: `cd C:\your\path\to\bcilab; bcilab`





# Starting the Toolbox

- Or if your path contains spaces, type:  
`cd('C:\your\path\to\bcilab'); bcilab`



# Starting the Toolbox

- If you have an unsupported OS/MATLAB combination, BCILAB might ask you some question about compiling functions
  - Just type n (for no) to continue
- If you have things on your MATLAB path that override BCILAB function names, you will get some warnings about it (it's best to remove them from the path)



# Starting the Toolbox

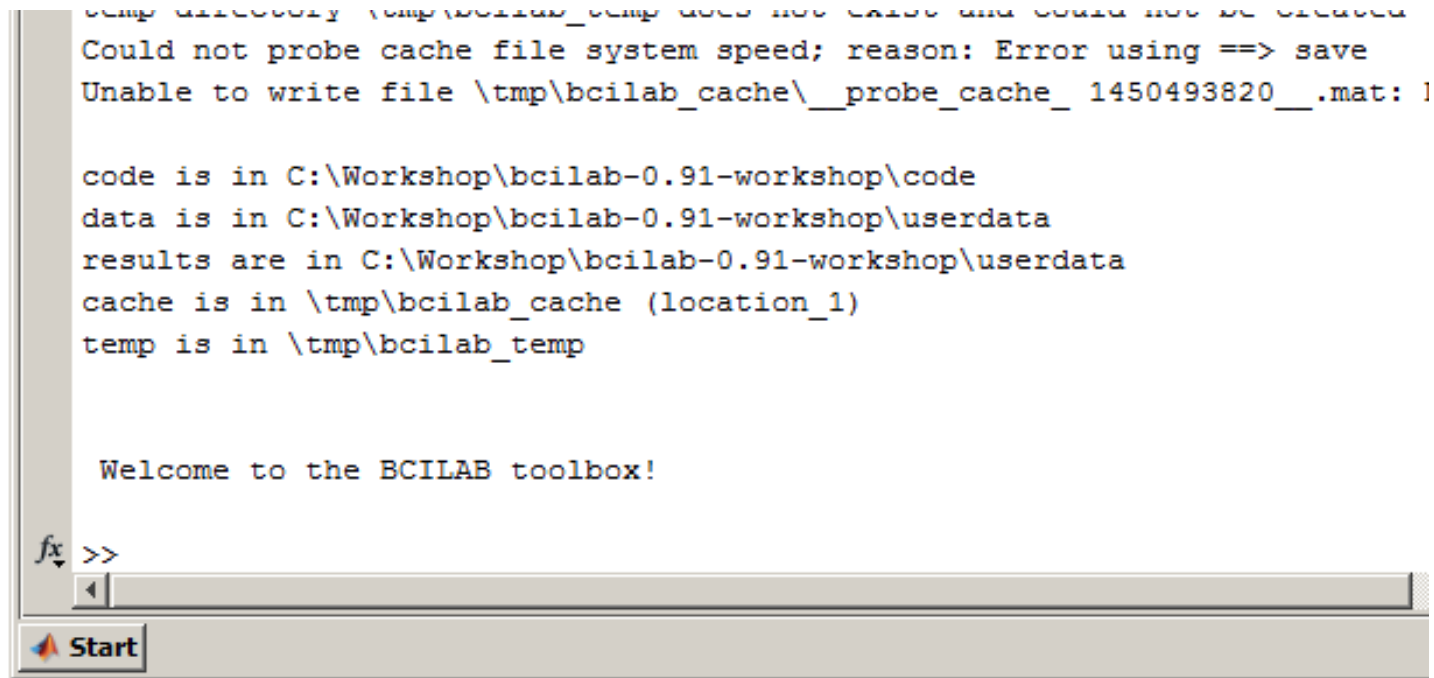
- You should now see the welcome message

```
temp directory \tmp\bcilab_temp does not exist and could not be created
Could not probe cache file system speed; reason: Error using ==> save
Unable to write file \tmp\bcilab_cache\__probe_cache_1450493820__.mat: 1

code is in C:\Workshop\bcilab-0.91-workshop\code
data is in C:\Workshop\bcilab-0.91-workshop\userdata
results are in C:\Workshop\bcilab-0.91-workshop\userdata
cache is in \tmp\bcilab_cache (location_1)
temp is in \tmp\bcilab_temp

Welcome to the BCILAB toolbox!

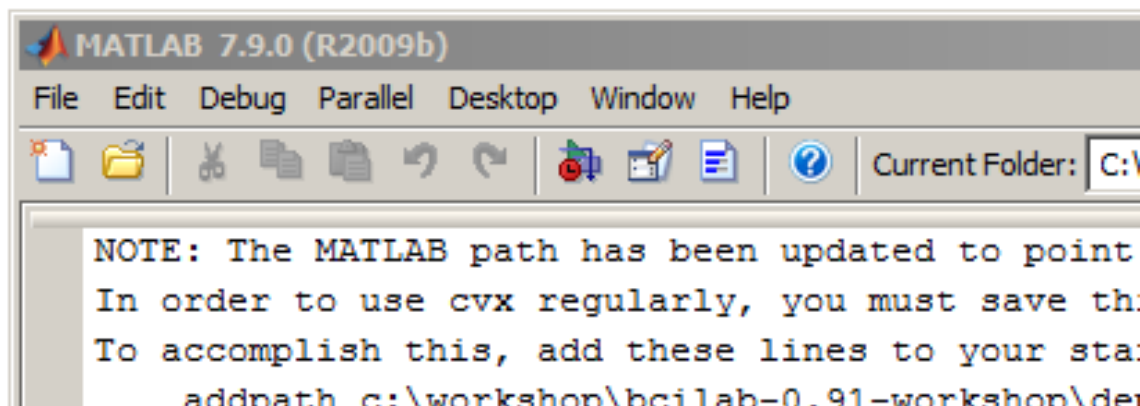
fx >>
```





# Starting the Toolbox

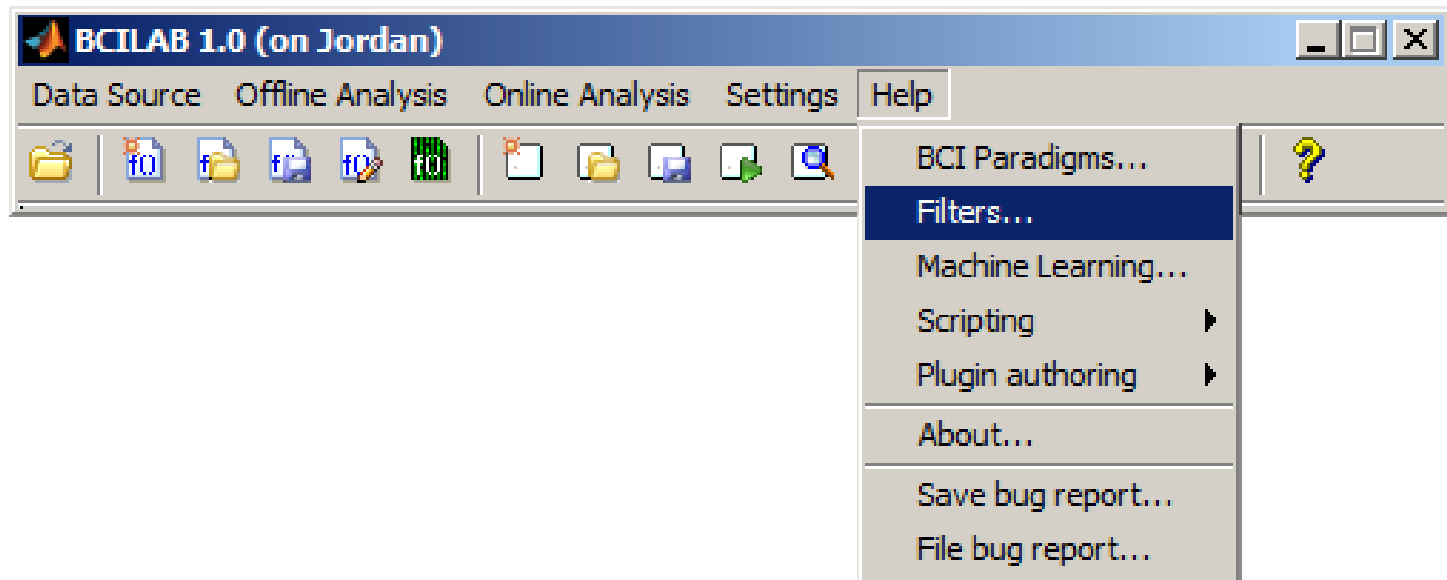
- ... and the main menu







# Getting help (if needed)





# Getting help (if needed)

The screenshot shows the MATLAB File Help window. The title bar reads 'Help'. The menu bar includes 'File', 'Edit', 'View', 'Go', 'Favorites', 'Desktop', 'Window', and 'Help'. The address bar shows 'MATLAB File Help: code/filters'. The left pane has a 'Contents' tab selected, showing a tree view with the following items: Release Notes, Installation, MATLAB, Signal Processing Toolbox, Statistics Toolbox, and Simulink 3D Animation. The main pane displays the 'code/filters' section. It has a sub-header 'code/filters' in red and a link 'Default Topics' on the right. Below the sub-header is the text 'Contents of filters:'. A list of filter functions is shown on the left, each with a blue underlined link: `flt_bandpower`, `flt_clean_channels`, `flt_clean_peaks`, `flt_clean_settings`, `flt_clean_spikes`, `flt_clean_windows`, `flt_coherence`, `flt_eog`, `flt_epochica`, `flt_epochpca`, `flt_fft`, `flt_fir`, `flt_fourier`, `flt_ica`, `flt_iir`, `flt_laplace`, and `flt_pipeline`. To the right of this list is a list of descriptions for each function, each preceded by a hyphen. The descriptions are: 'Compute logarithmic bandpower features.', 'Remove channels with abnormal data from a continu...', 'Project local peaks out of the data (blinks, musc...', 'Clean EEG data according to a particular cleaning...', 'Set outlier samples in the data to zero.', 'Remove periods of abnormal data from continuous d...', 'Calculate between-channel / component coherence.', 'Remove EOG artifacts from EEG using EOG reference...', 'Apply an independent component decomposition acro...', 'Apply a principal component decomposition across...', 'Apply an FFT to each epoch of an epoched signal (...', 'Filter a continuous data set by a digital FIR fil...', 'Transform an epoched data set into a fourier repr...', 'Annotate the Signal with a spatial decomposition', 'Filter a continuous data set by a digital IIR low...', 'Applies a simple Hjorth-style surface laplacian f...', and 'Configurable preprocessing pipeline for most ECI...'. The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

Help

File Edit View Go Favorites Desktop Window Help

MATLAB File Help: code/filters

MATLAB File Help: code/filters [Default Topics](#)

## code/filters

Contents of filters:

- [flt\\_bandpower](#)
- [flt\\_clean\\_channels](#)
- [flt\\_clean\\_peaks](#)
- [flt\\_clean\\_settings](#)
- [flt\\_clean\\_spikes](#)
- [flt\\_clean\\_windows](#)
- [flt\\_coherence](#)
- [flt\\_eog](#)
- [flt\\_epochica](#)
- [flt\\_epochpca](#)
- [flt\\_fft](#)
- [flt\\_fir](#)
- [flt\\_fourier](#)
- [flt\\_ica](#)
- [flt\\_iir](#)
- [flt\\_laplace](#)
- [flt\\_pipeline](#)

- Compute logarithmic bandpower features.
- Remove channels with abnormal data from a continu...
- Project local peaks out of the data (blinks, musc...
- Clean EEG data according to a particular cleaning...
- Set outlier samples in the data to zero.
- Remove periods of abnormal data from continuous d...
- Calculate between-channel / component coherence.
- Remove EOG artifacts from EEG using EOG reference...
- Apply an independent component decomposition acro...
- Apply a principal component decomposition across...
- Apply an FFT to each epoch of an epoched signal (...
- Filter a continuous data set by a digital FIR fil...
- Transform an epoched data set into a fourier repr...
- Annotate the Signal with a spatial decomposition
- Filter a continuous data set by a digital IIR low...
- Applies a simple Hjorth-style surface laplacian f...
- Configurable preprocessing pipeline for most ECI...



# Getting help (if needed)

The screenshot shows the MATLAB File Help window for the `fit_bandpower` function. The window has a menu bar (File, Edit, View, Go, Favorites, Desktop, Window, Help) and a search bar. The left sidebar shows a tree view of the help content, including Release Notes, Installation, MATLAB, Signal Processing Toolbox, Statistics Toolbox, and Simulink 3D Animation. The main content area displays the function name `fit_bandpower` in red, followed by a description: "Compute logarithmic bandpower features." and the function signature: `[Signal, State] = flt_bandpower(Signal, Bands, Smoothing, State)`. Below this, there is a "TODO: detailed description" section. The "In:" section lists the inputs: `Signal` (continuous data set to be filtered) and `Bands` (bands specification). The `Bands` section includes two bullet points: one for a cell array of frequency bands (e.g., `{[8, 10], [12, 16], [22, 30]}`) and another for a two-dimensional cell array containing the frequency bands (second dimension) of each channel (first dimension), e.g., `{{[7, 11], [13, 18]}, {[6, 35]}, {[12, 15], [20, 22]}}`.

Help

File Edit View Go Favorites Desktop Window Help

Search

Contents Search Results

- Release Notes
- Installation
- MATLAB
- Signal Processing Toolbox
- Statistics Toolbox
- Simulink 3D Animation

MATLAB File Help: `fit_bandpower` [View code for flt\\_bandpower](#) [Default Topics](#)

## `fit_bandpower`

Compute logarithmic bandpower features.

`[Signal, State] = flt_bandpower(Signal, Bands, Smoothing, State)`

TODO: detailed description

In:

`Signal` : continuous data set to be filtered

`Bands` : bands specification:

- \* if all channels have the same bands, use a cell array containing frequency bands, e.g. `{[8, 10], [12, 16], [22, 30]}` (in Hz) this example creates 3 identical bands for each channel
- \* if you want individual bands for each channel, use a two-dimensional cell array containing the frequency bands (second dimension) of each channel (first dimension), e.g. `{{[7, 11], [13, 18]}, {[6, 35]}, {[12, 15], [20, 22]}}`, this example creates 2 bands for the first channel, 1 band for the second channel, and 3 bands for the third channel



## 2 Offline ERP Analysis



# The Data

- Provided by Grainne McLoughlin
- Contains data from a Flanker task
- Two groups of markers:
  - S101, S102: person presses a button and **commits no error**
  - S201, S202: person presses a button and **commits an error**



# Experimental Task

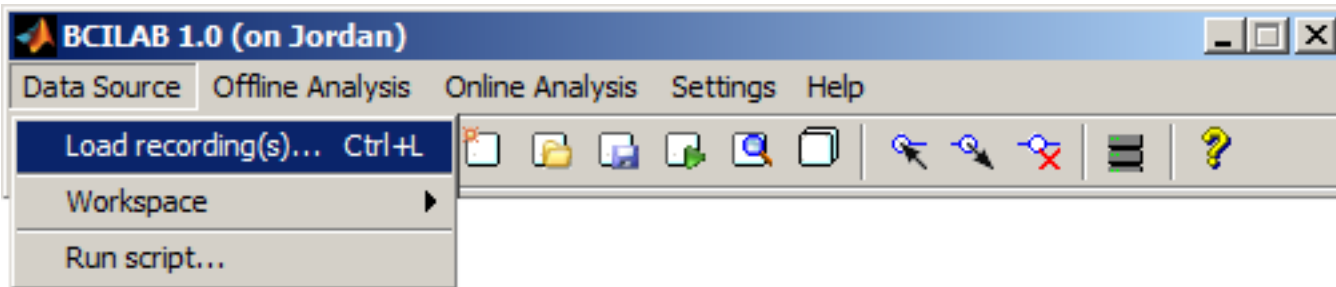
- **Flanker Task:** The experiment consists of a sequence of ca. 330 trials with inter-trial interval of 2s +/- 1.5s
- At the beginning of each trial, an arrow is presented centrally (pointing either left or right)
- The arrow is flanked by congruent or incongruent “flanker” arrows:



- The subject is asked to press the left/right button, according to the central arrow, and makes frequent errors (25%)

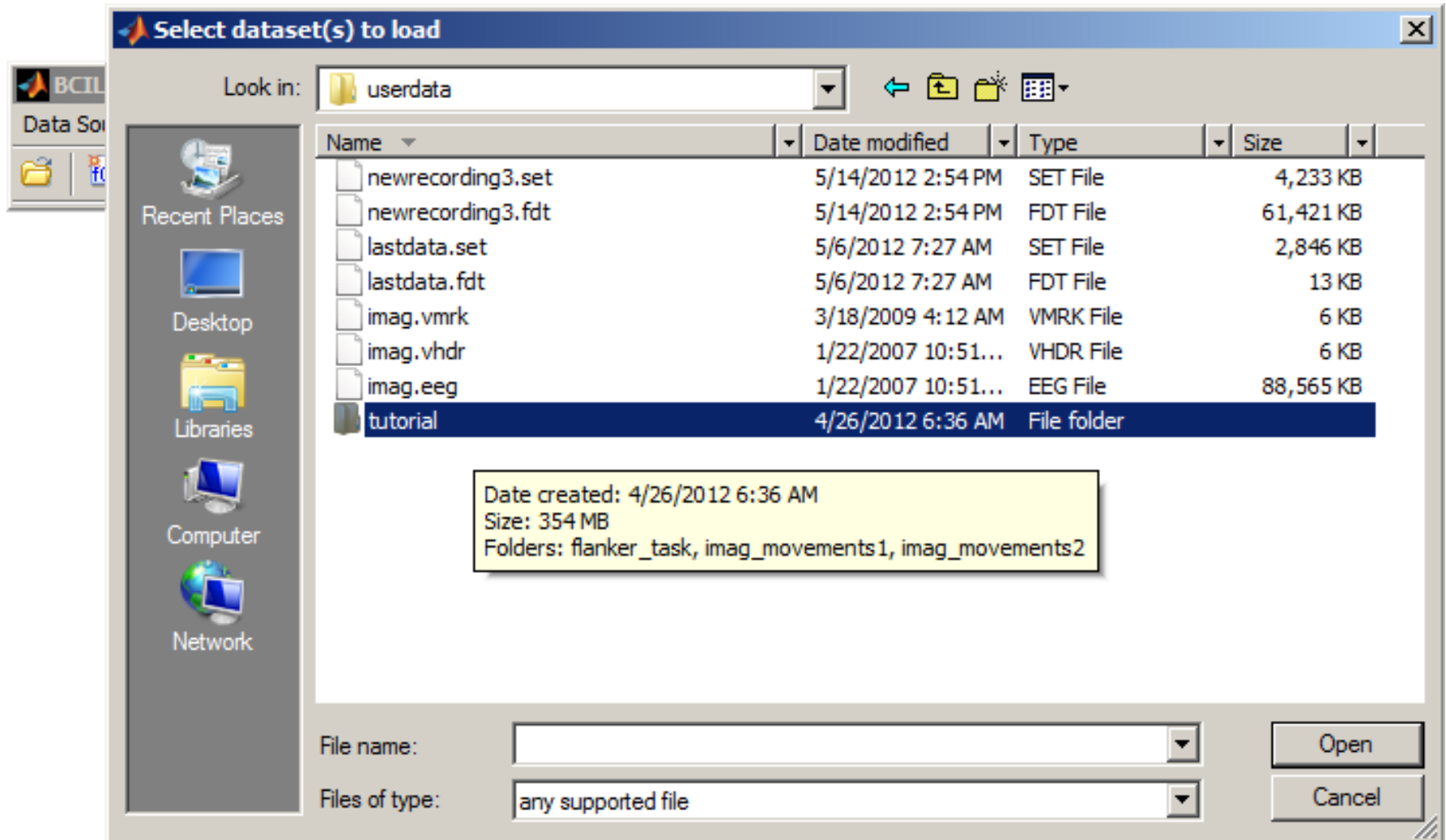


# Loading the Data





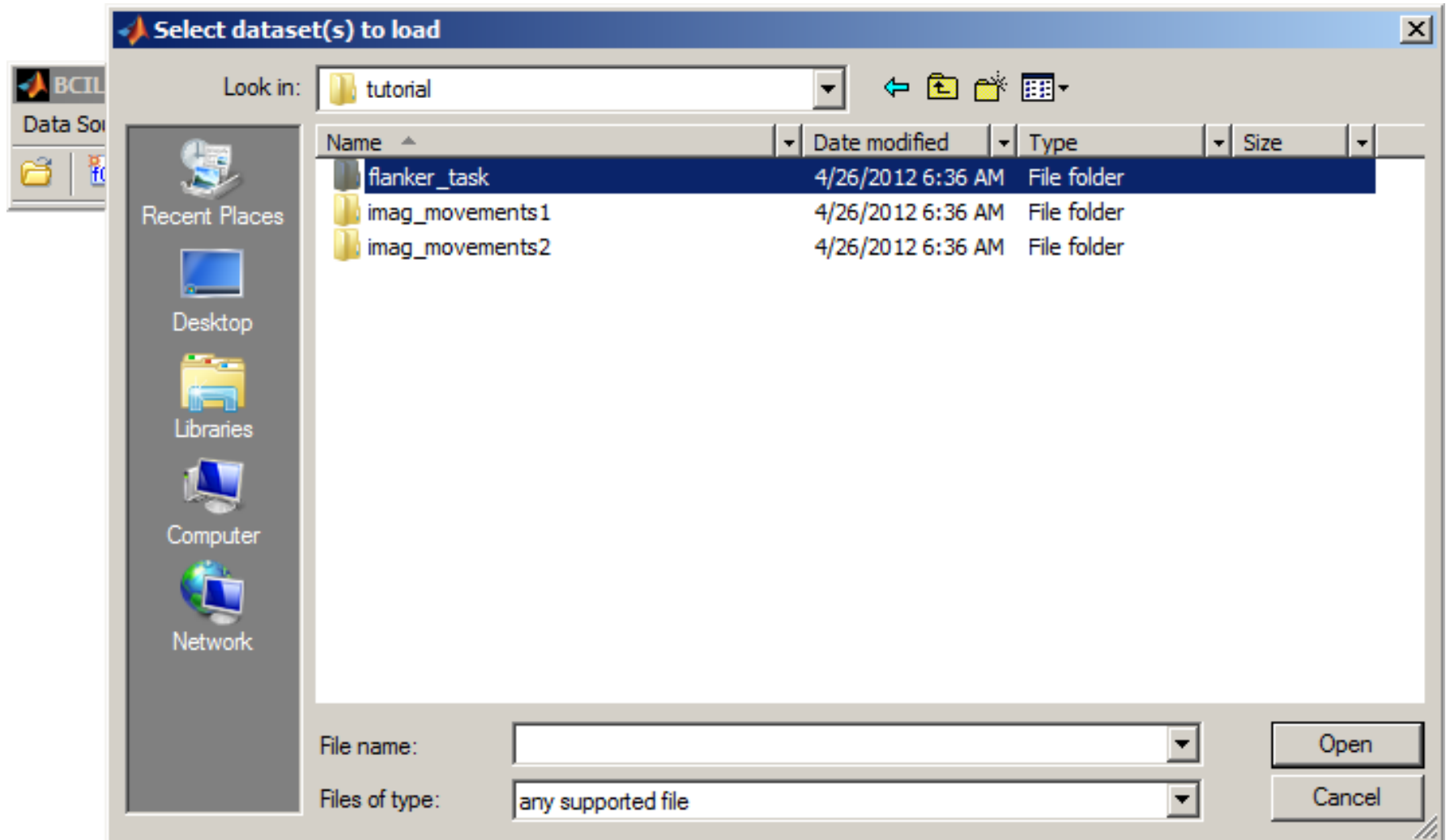
# Loading the Data





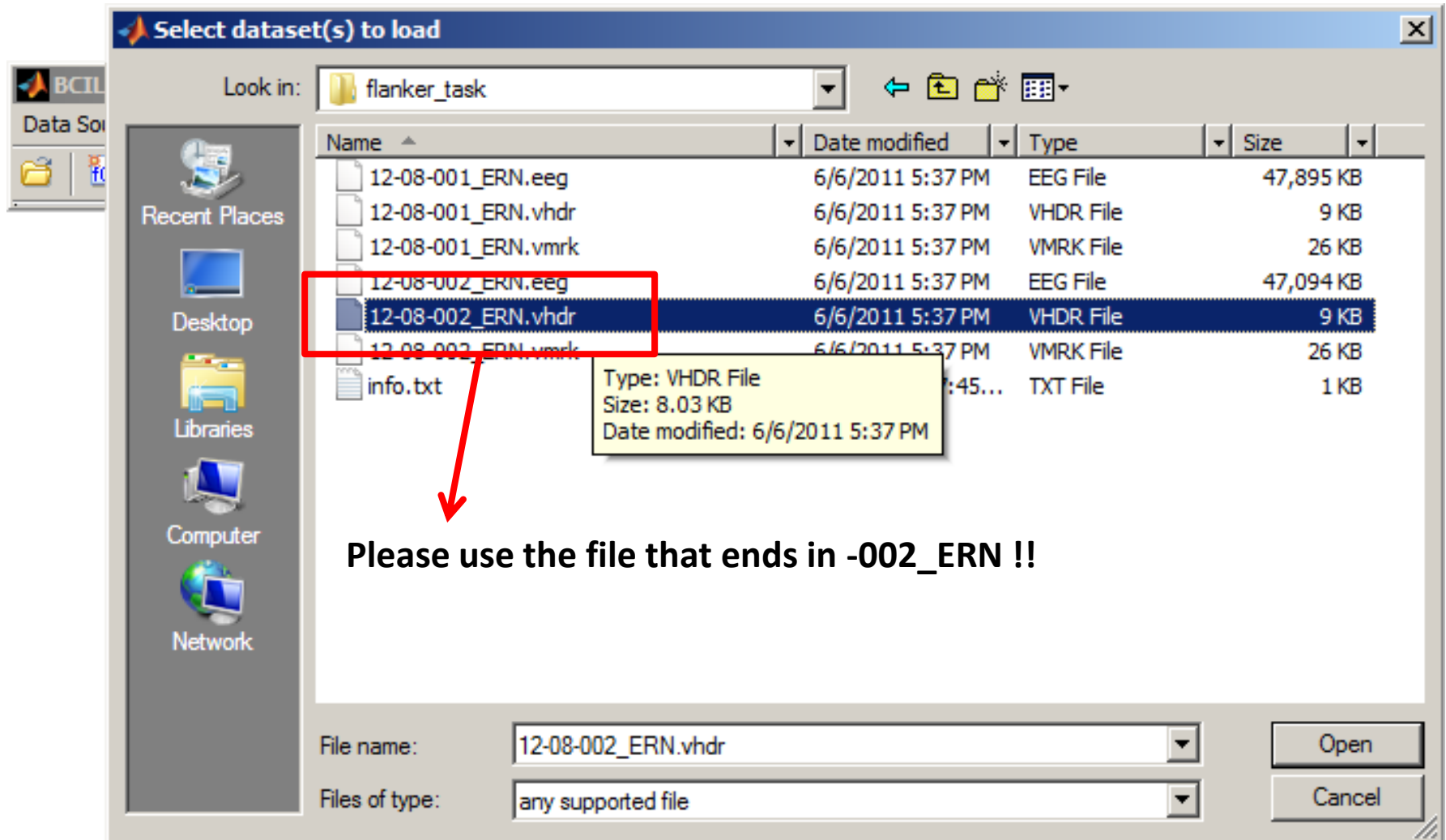


# Loading the Data



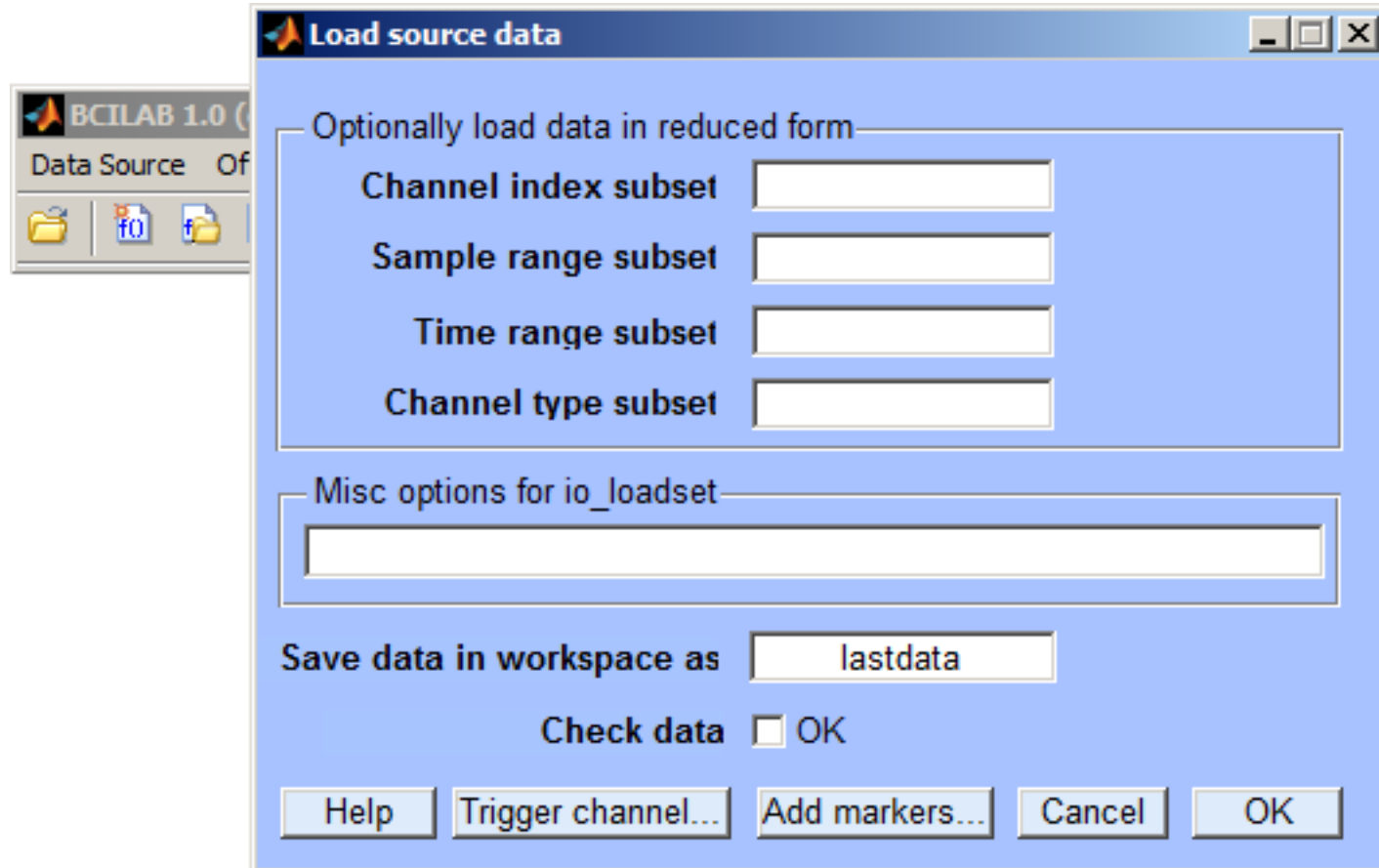


# Loading the Data





# Confirming Import Options



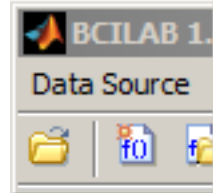


# Creating a New Approach





# Creating a New Approach



**Define a new approach**

Select approach

log-Bandpower (ParadigmBandpower)

Description

Basic paradigm for oscillatory processes, via per-channel logarithmic bandpower (note: fairly primitive by modern standards)

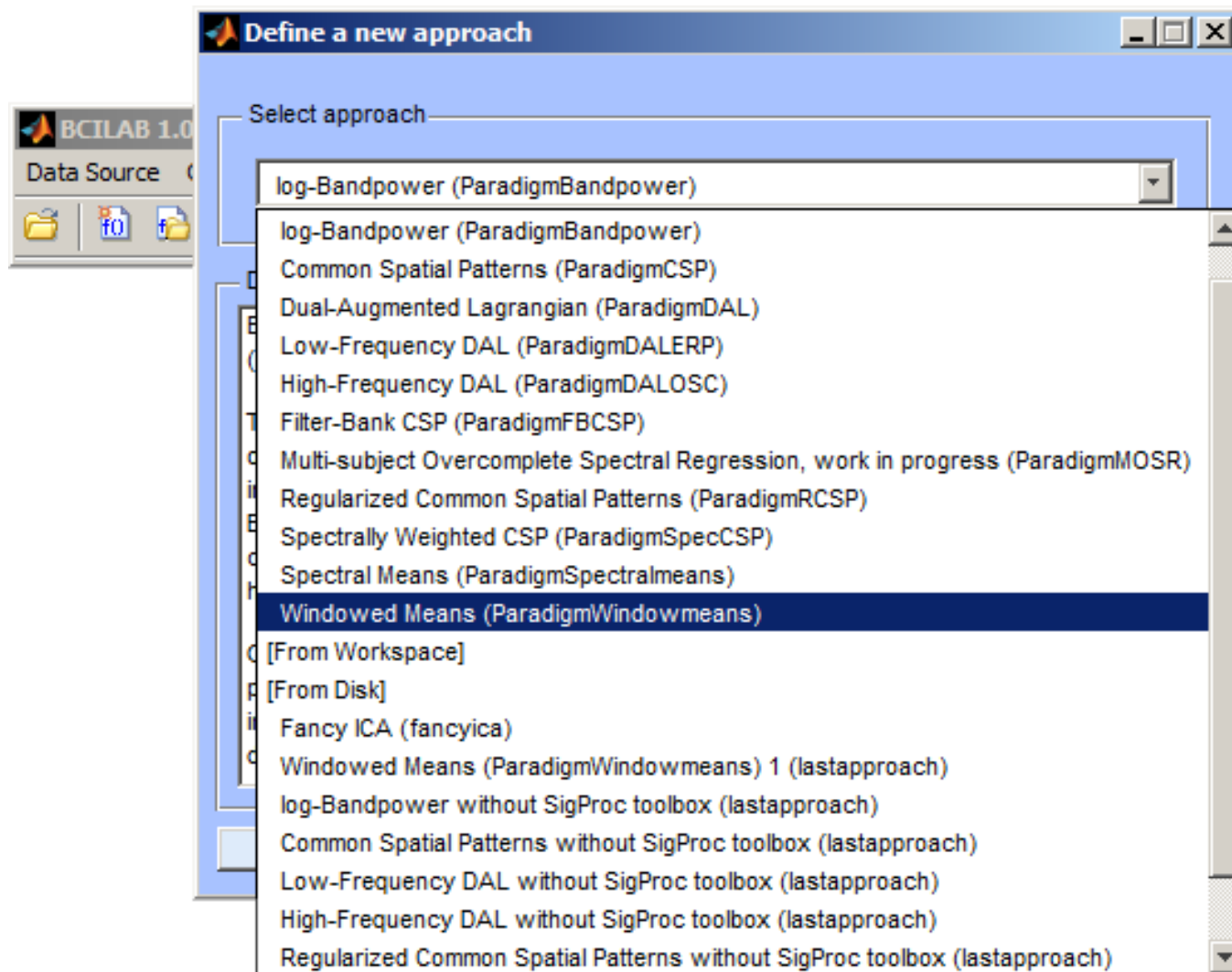
The logarithmic Bandpower estimates ("log-BP") paradigm is based on the design of the original Graz Brain-Computer Interface [1][5], which used lateralized motor imagery for control. The features exploited by this paradigm in its original form are Event-Related Synchronization and Desynchronization [2] localized in the motor cortex, but the paradigm is not restricted to these applications. Similar measures have also been used in [4] although without machine learning.

Generally, log-BP can be used as a simple method to operate on oscillatory processes, either in relation to events, or asynchronously. The paradigm is simple in that it does not capture any complex time variations in the oscillations detected, does not capture interactions between multiple frequency bands, and does not

Help ☐ Full edit when done Cancel OK

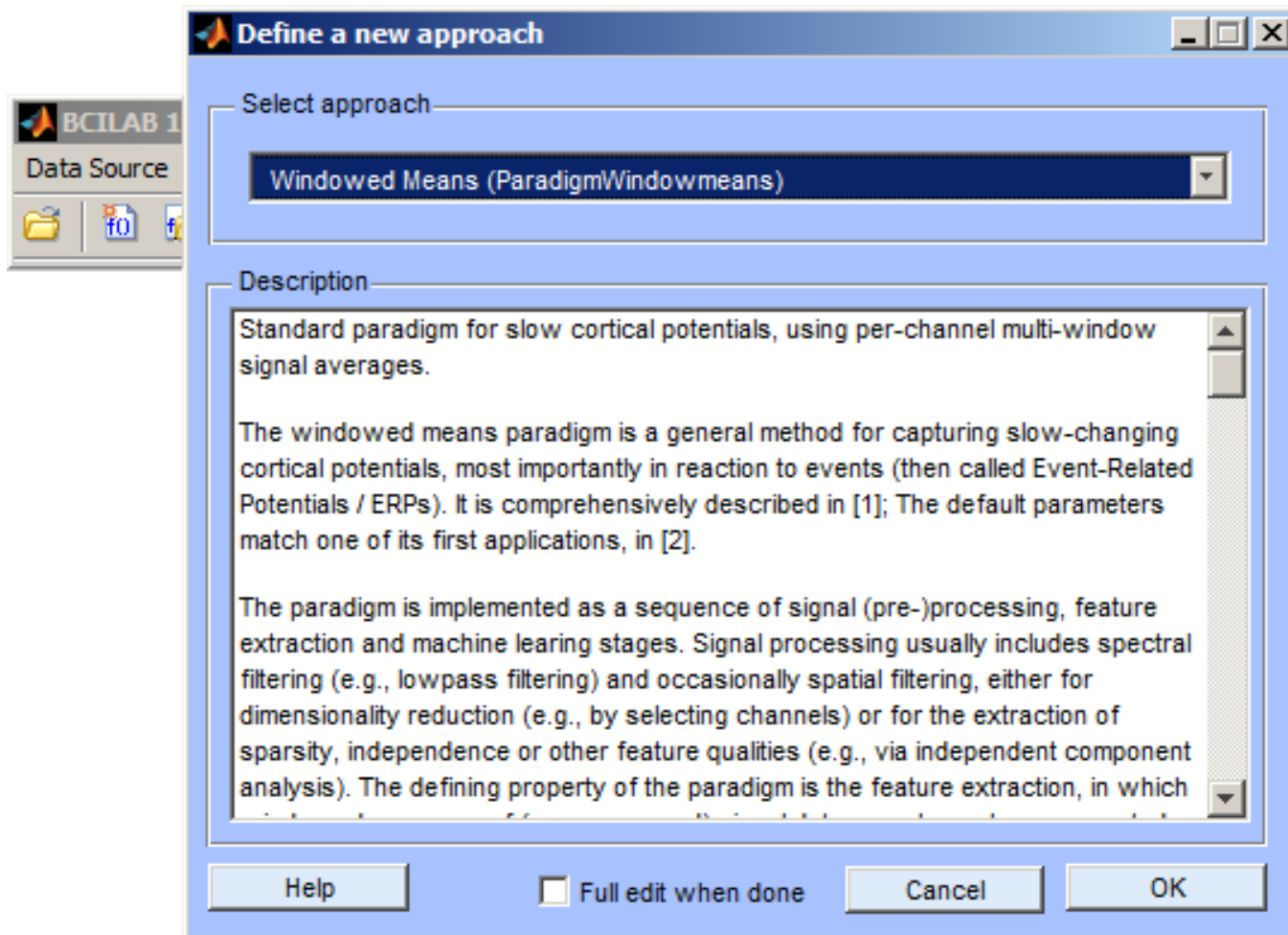


# Select an ERP Paradigm





# Select an ERP Paradigm





# Configuring the Approach

BCILAB: Configure approach

New sampling rate of the data: 100

Epoch time window relative to the target markers: [-0.2 0.8]

Frequency-domain selection: [0.1 15]

Epoch intervals to take as features: [45; 0.45 0.5; 0.5 0.55; 0.55 0.6]

Machine learning function: lda

Buttons: Help, Cancel, Ok

Type into the lowest of the 3 highlighted fields:

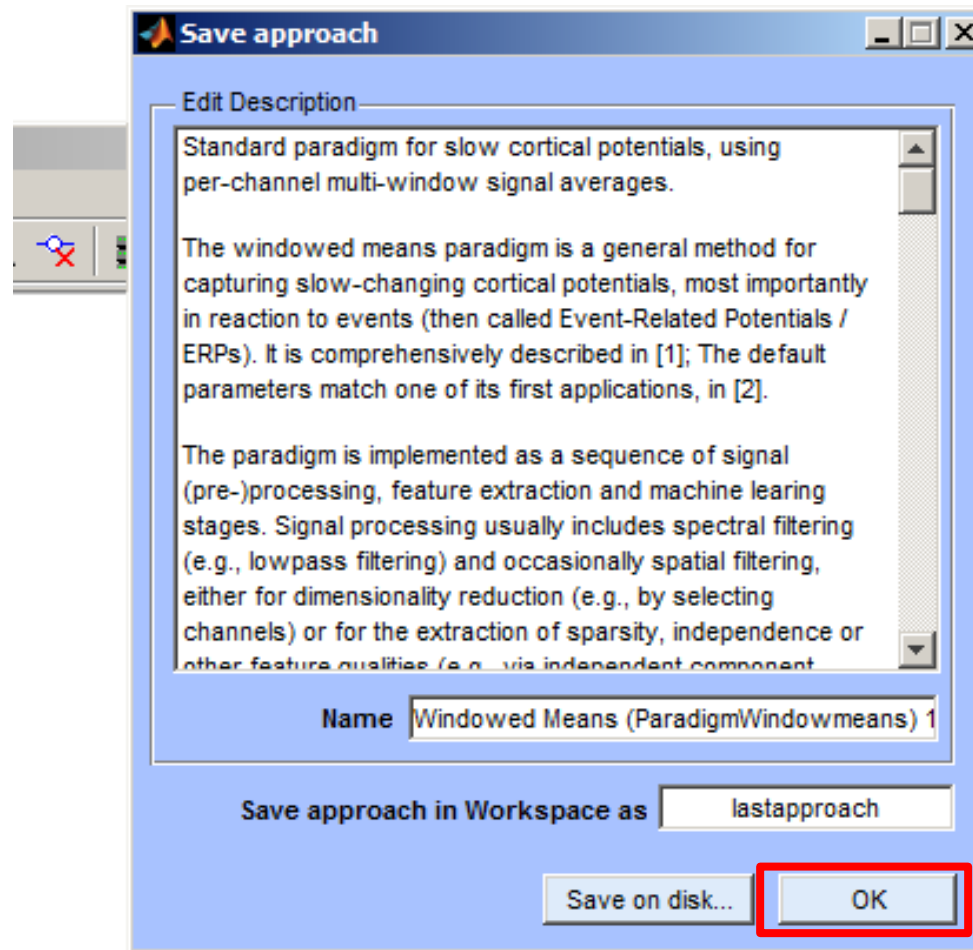
[0.25 0.3; 0.3 0.35; 0.35 0.4; 0.4 0.45; 0.45 0.5; 0.5 0.55; 0.55 0.6]

**Note:** On macOS, the latest MATLAB version likes to pop up this window in full screen; For the time being it is best to resize it to something similar to the above



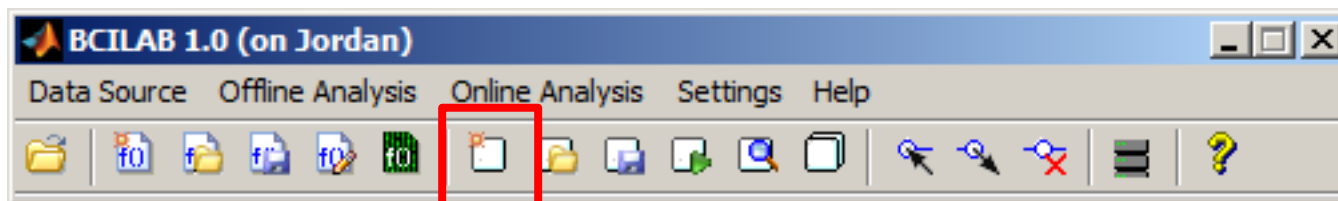


# Saving to the Workspace





# Calibrating a New Model





# Calibrating a New Model

**Calibrate a model**

Selected approach: ans ("Windowed Means (Paradig...)

Calibration data source: lastdata ("12-08-002\_EPN.vhdr")

Target markers: **`{'S102'}, {'S201'}, {'S202'}}`**

Inspect data...

Parameter Search

Loss/Performance Metric: Automatically chosen

Cross-validation folds: 5

Spacing around test trials: 5

Performance estimates

☒ Compute performance estimates

Cross-validation folds: 5

Spacing around test trials: 5

Computing resources

☐ Run on a computer cluster

Node pool: (use current config)

Save model in workspace as: lastmodel

Save stats in workspace as: laststats

Help Cancel OK

This is the set of marker labels that determine our two possible error conditions. For each of the two conditions, there is a group of multiple markers (different types of errors and non-errors).

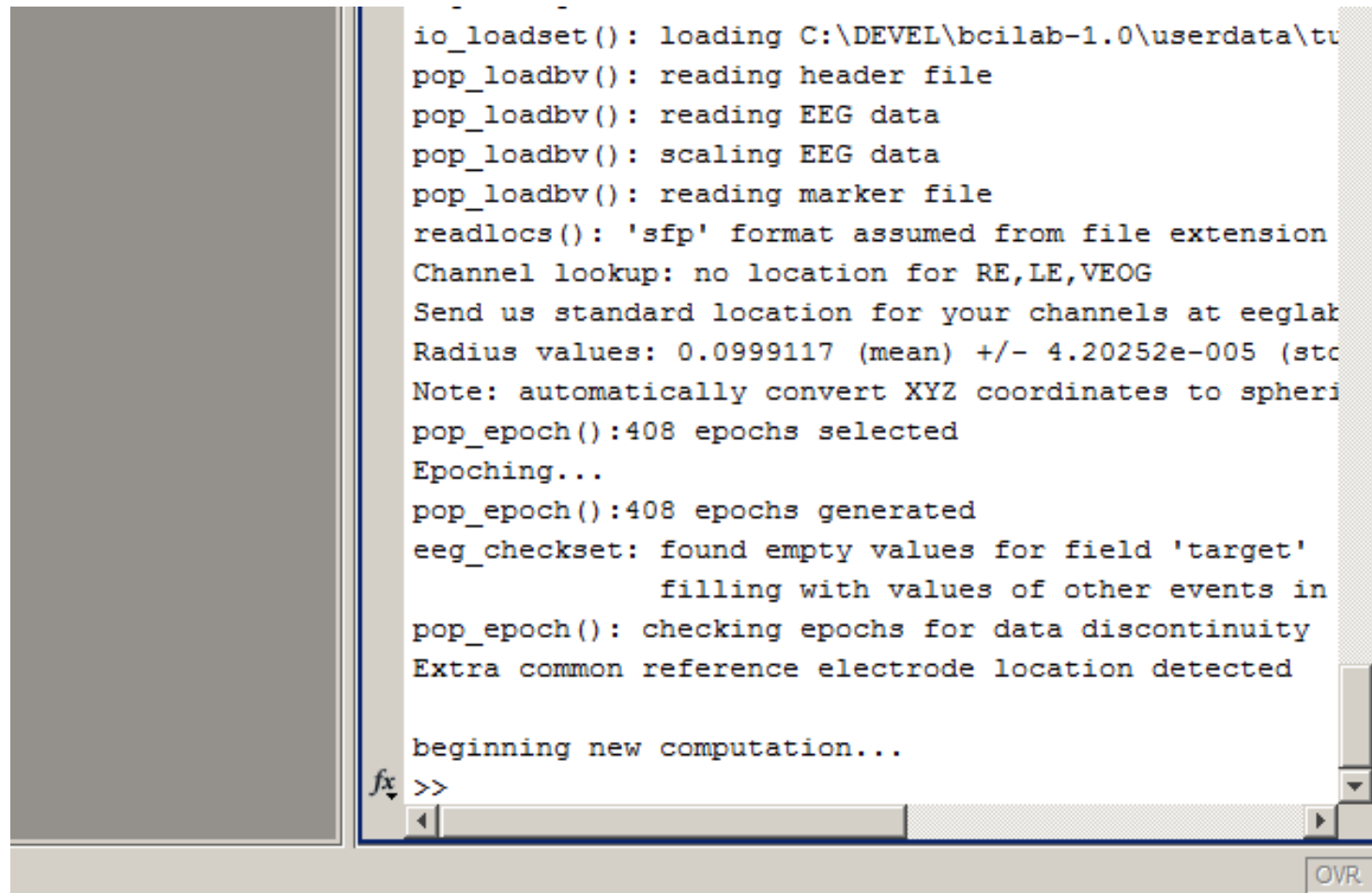
Type the following here:  
**`{{'S101','S102'}, {'S201','S202'}}`**



# Watching the Computation...

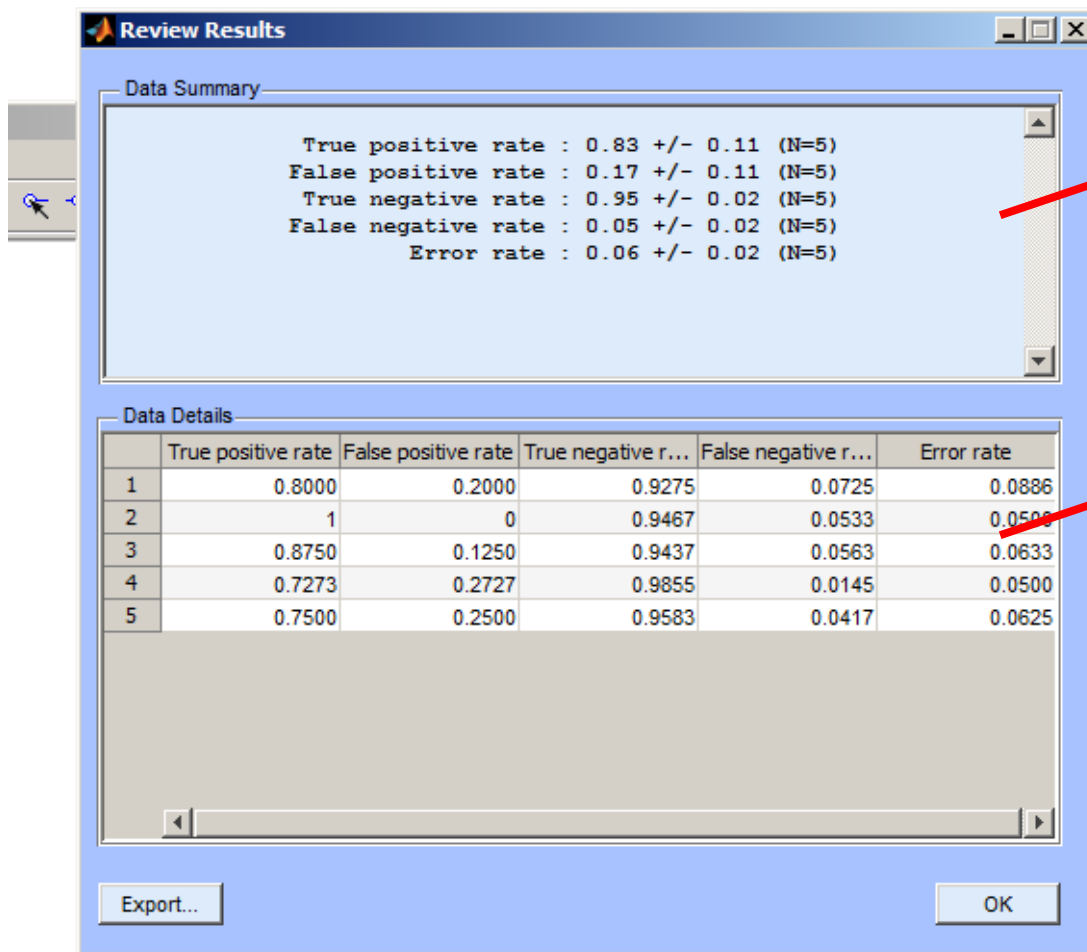
```
io_loadset(): loading C:\DEVEL\bcilab-1.0\userdata\tv
pop_loadbv(): reading header file
pop_loadbv(): reading EEG data
pop_loadbv(): scaling EEG data
pop_loadbv(): reading marker file
readlocs(): 'sfp' format assumed from file extension
Channel lookup: no location for RE,LE,VEOG
Send us standard location for your channels at eeglab
Radius values: 0.0999117 (mean) +/- 4.20252e-005 (std)
Note: automatically convert XYZ coordinates to spherical
pop_epoch():408 epochs selected
Epoching...
pop_epoch():408 epochs generated
eeg_checkset: found empty values for field 'target'
              filling with values of other events in
pop_epoch(): checking epochs for data discontinuity
Extra common reference electrode location detected

beginning new computation...
fx >>
```





# Reviewing Results



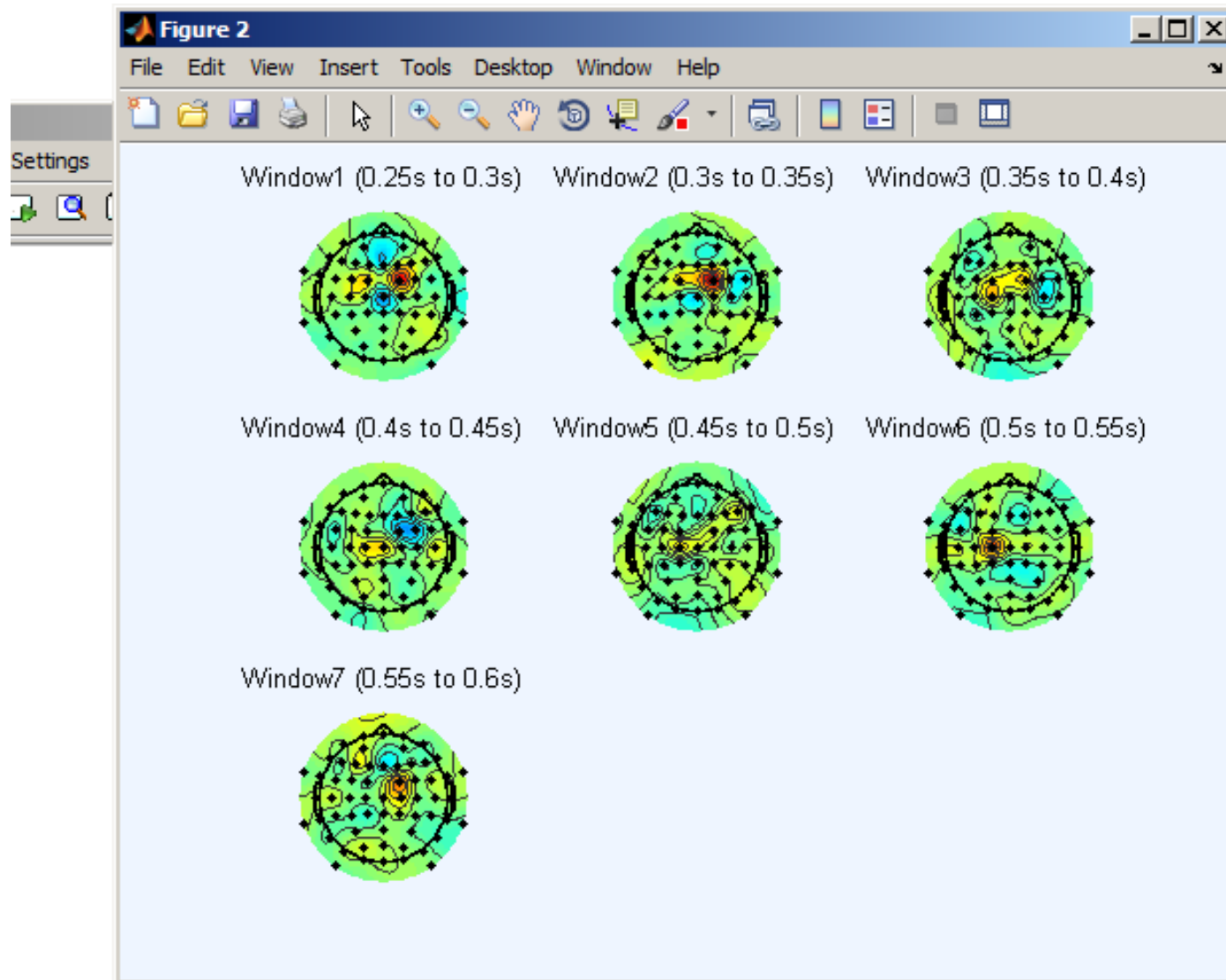
Summary Statistics

Statistics for each fold of the Cross-Validation (here 5x).



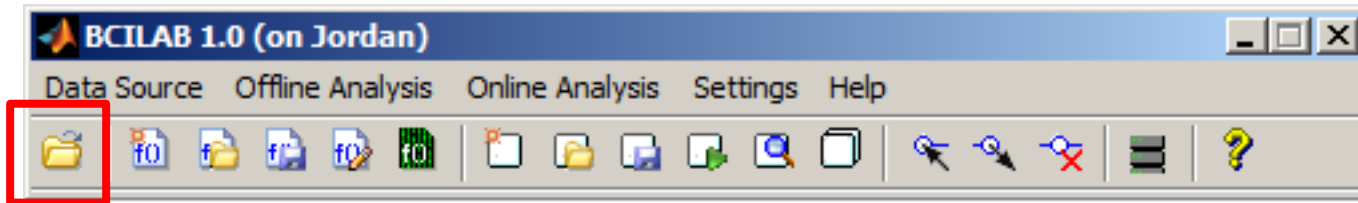


# Visualizing the Model





# Loading a Separate Test Set

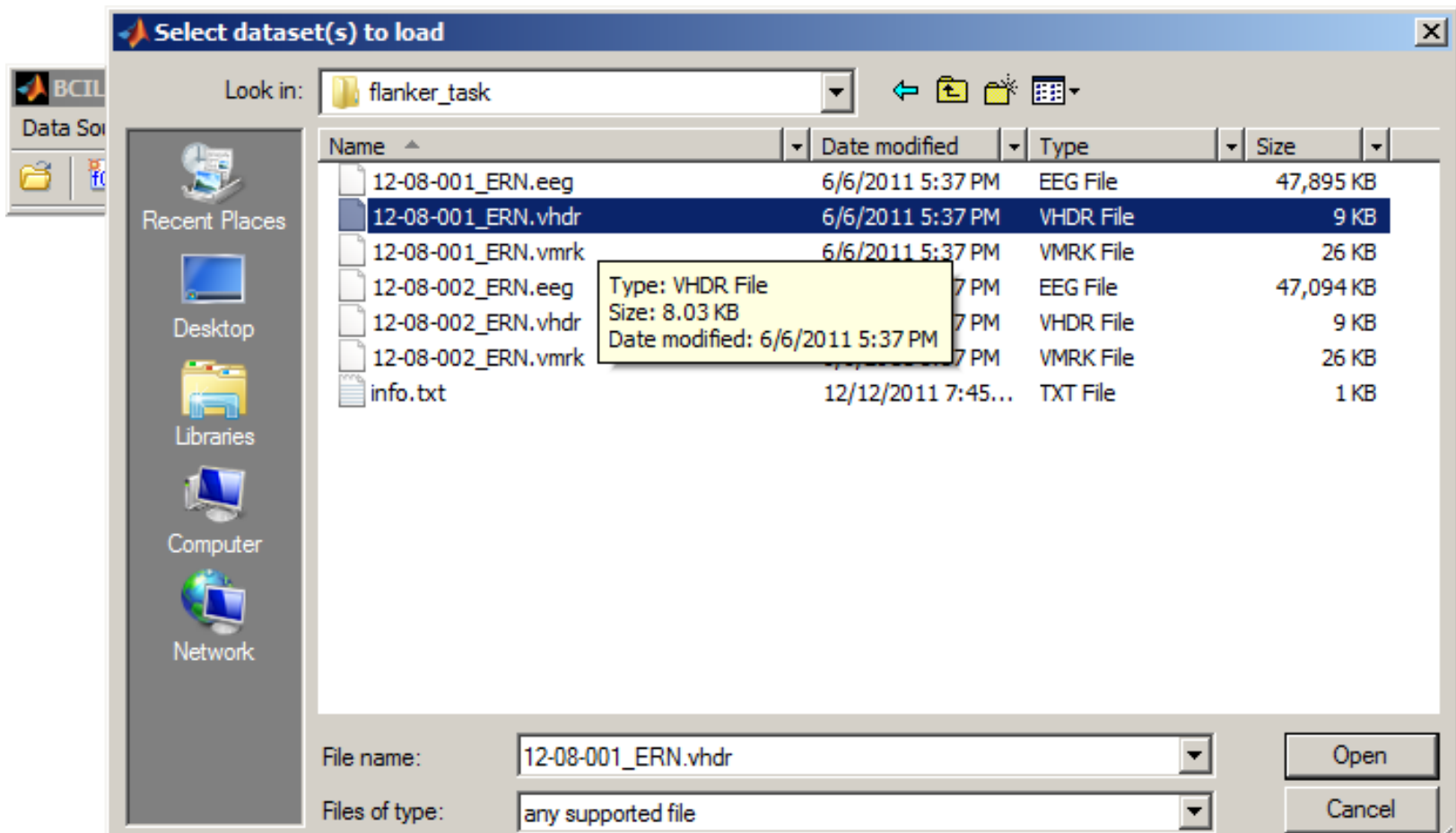






# Loading a Separate Test Set

Note: This data set is from an identical twin doing the same task.



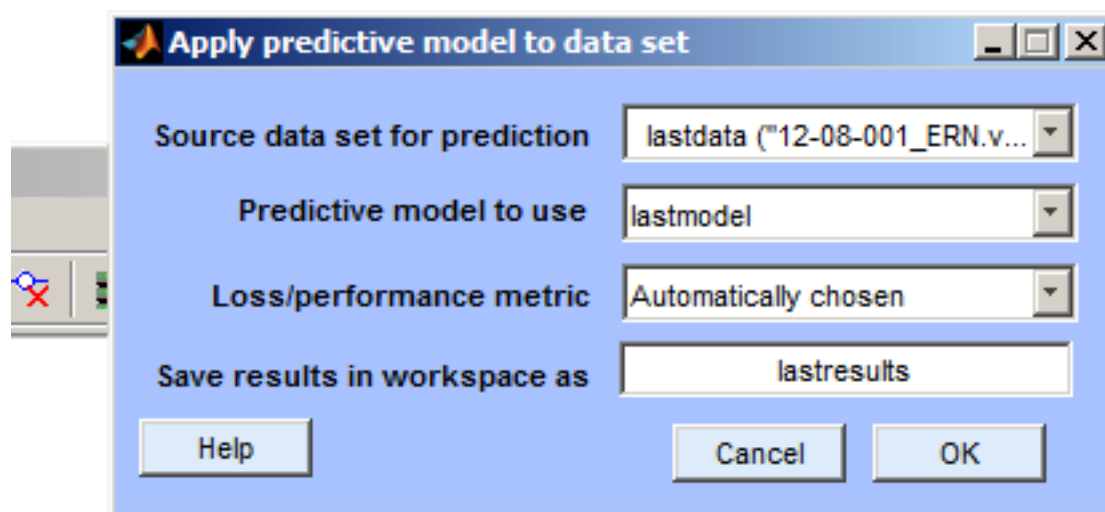


# Applying the Model to Test Data



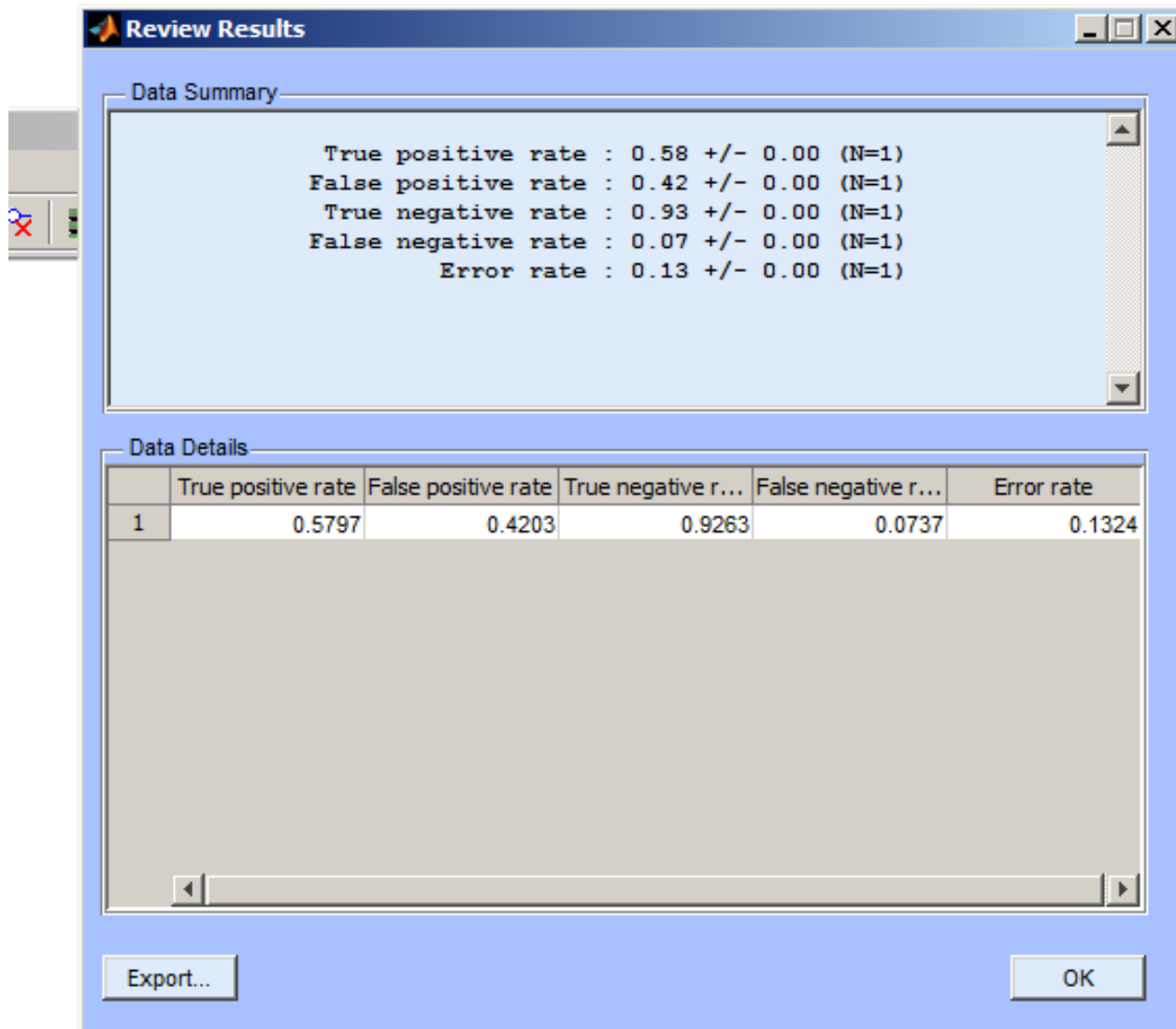


# Applying the Model to Test Data



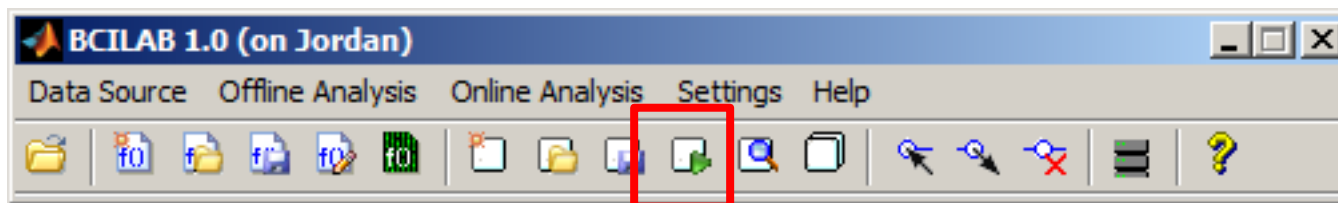


# Reviewing Statistics



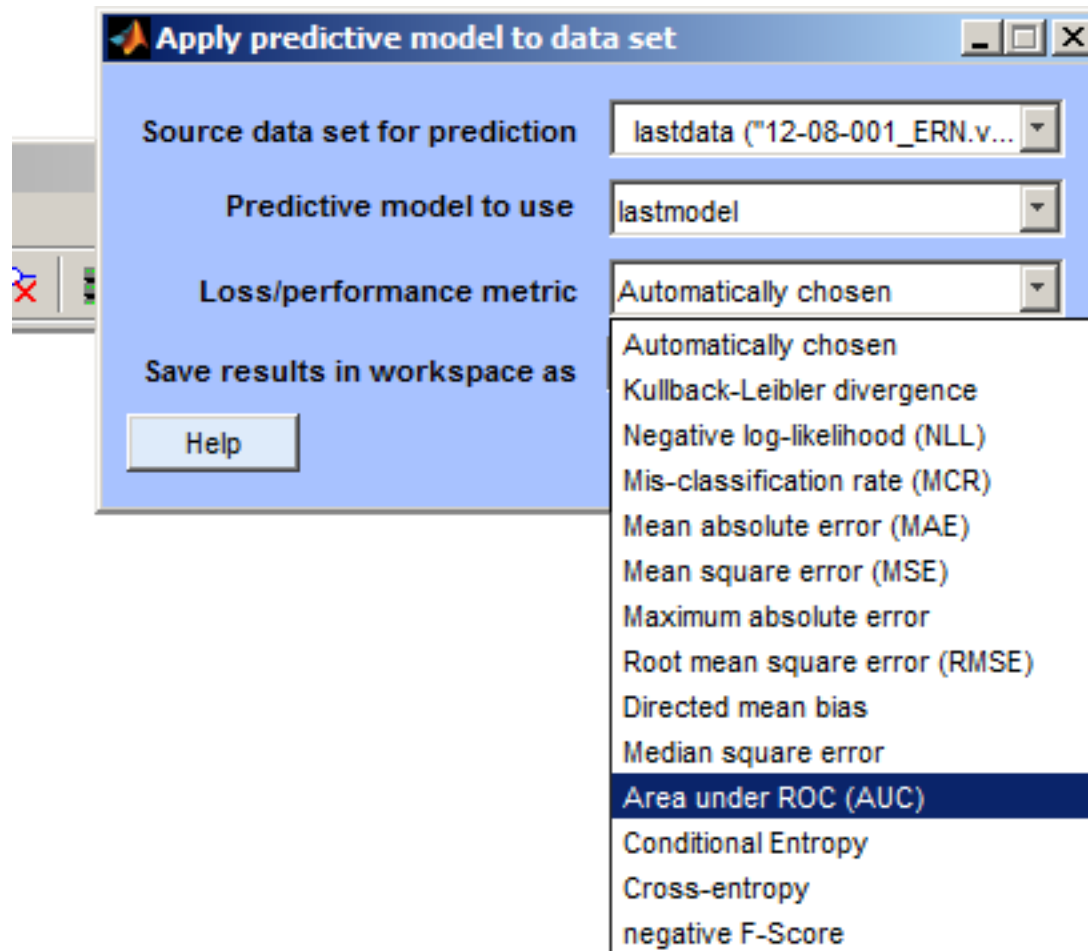


# Using Another Loss Measure



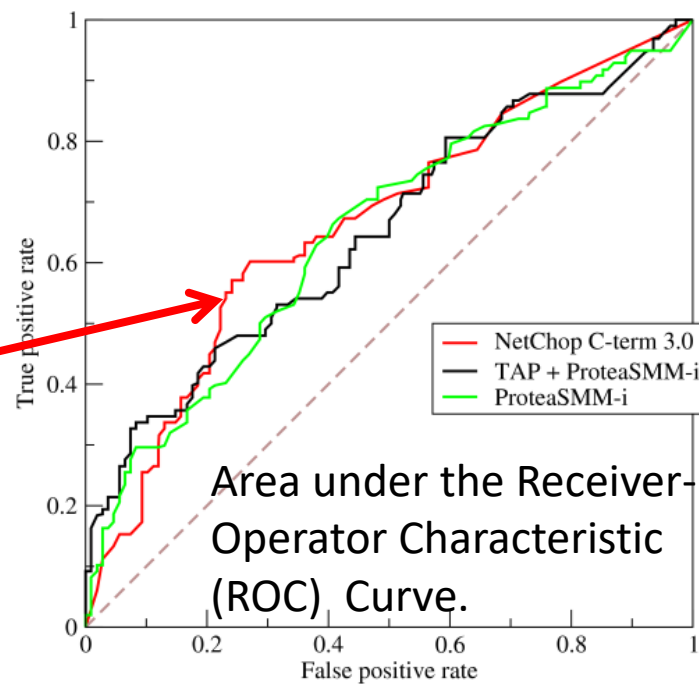
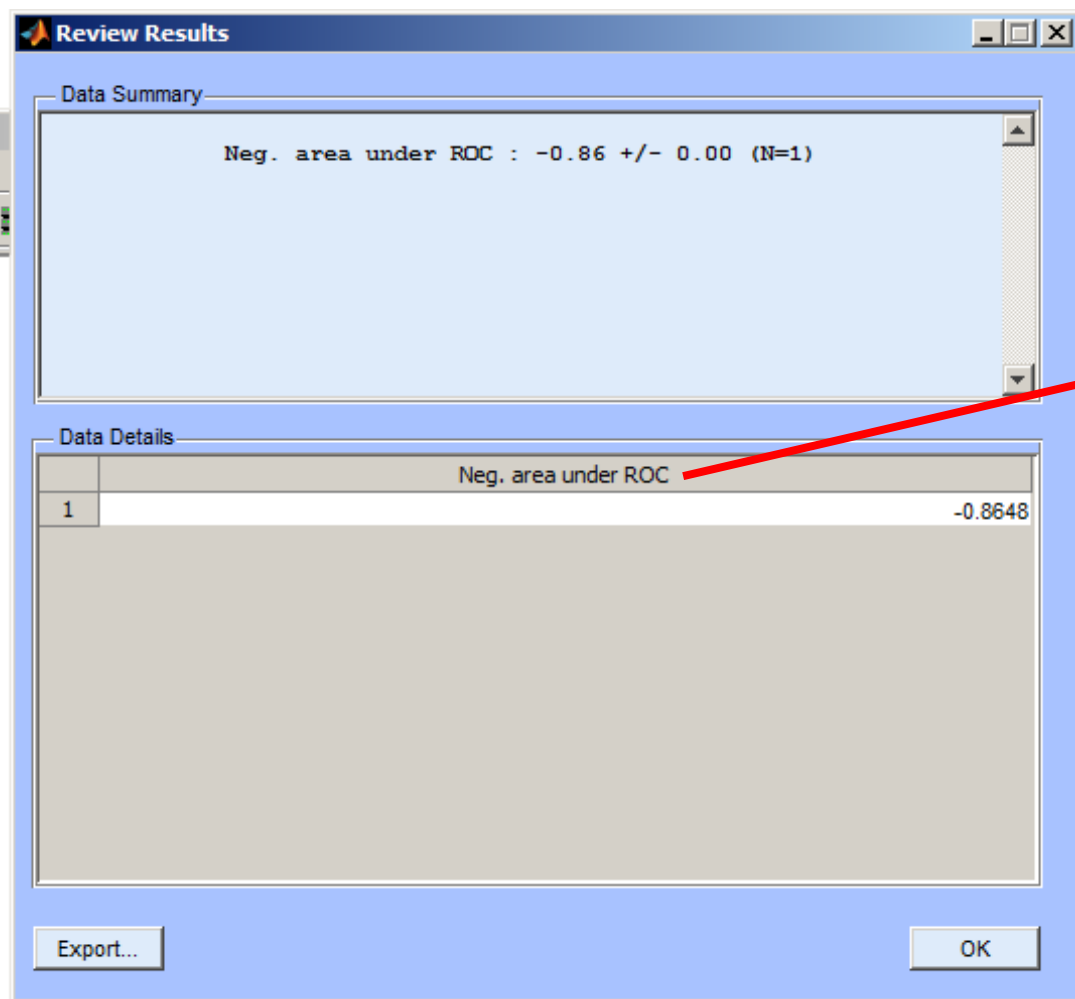


# Using Another Loss Measure





# Using Another Loss Measure



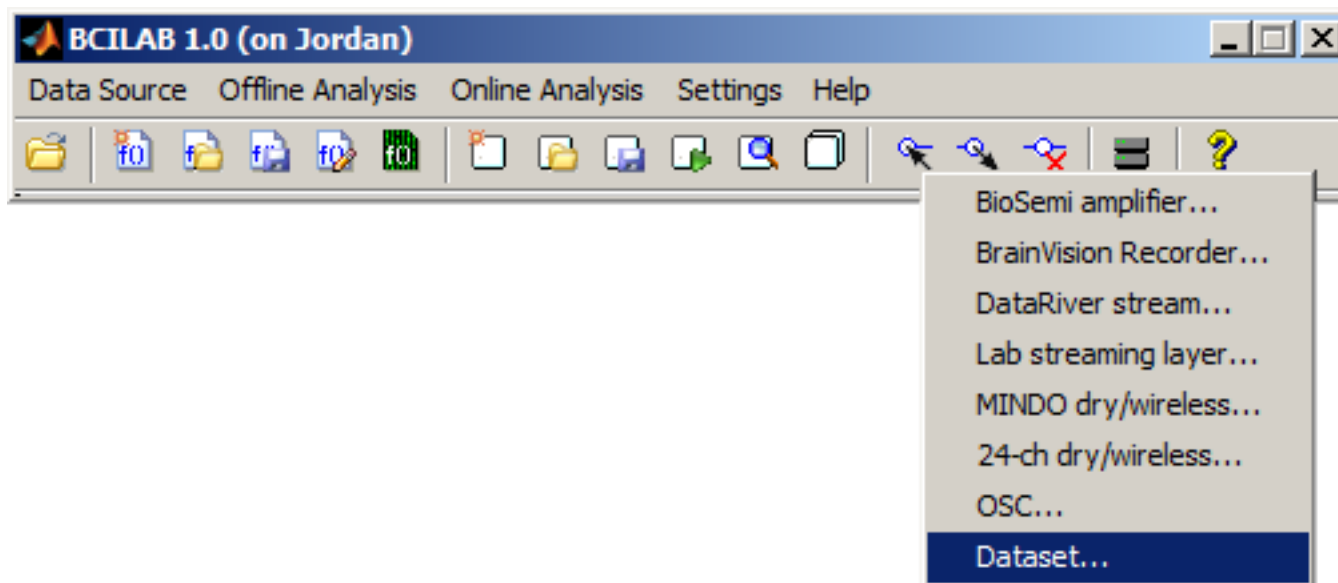


## 3 Online ERP Analysis





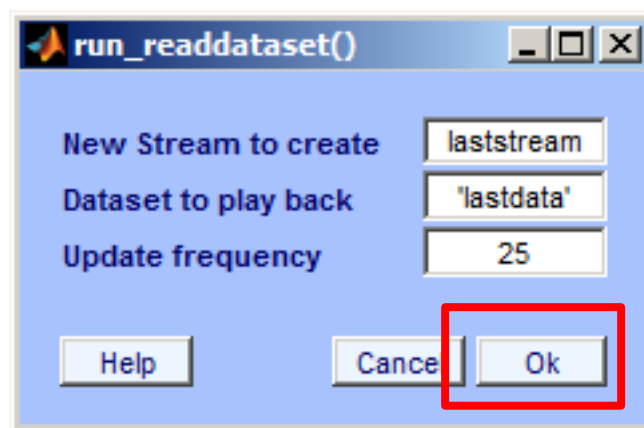
# Starting an Online Data Stream



The selected stream will be played back in the background.  
In this course we'll be playing back the test data set in real time  
(instead of reading from an actual device).



# Configuring the Online Stream





# Meanwhile in the MATLAB Workspace...

If you type **whos** you could see the data structure (laststream) that is updated in the background.

```
beginning new computation...
beginning evaluation...
Extra common reference electrode location detected
>> whos
```

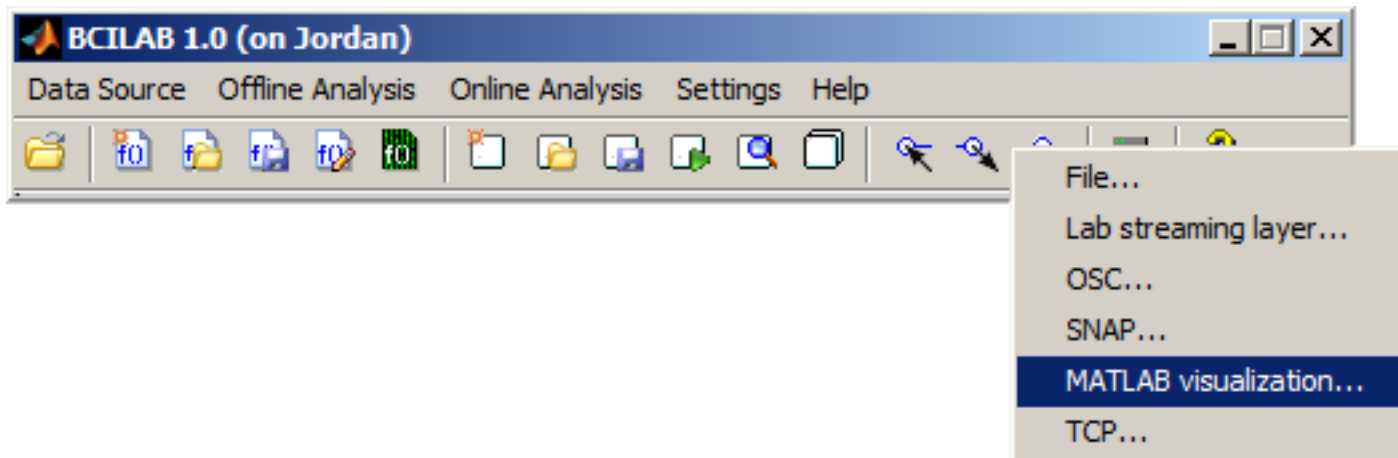
Name	Size	Bytes	Class
ans	1x1	38164	struct
f	1x1	8	double
lastapproach	1x1	38164	struct
lastchunk	64x15	3840	single
lastdata	1x1	874	struct
lastmodel	1x1	356657	struct
lastresults	1x1	12543	struct
laststats	1x1	418464	struct
laststream	1x1	7804880	struct
y	1x2	16	double

```
fx >> |
```

OVR



# Selecting the Destination for BCI Outputs





# Selecting the Destination for BCI Outputs

The dialog box 'run\_writevisualization()' contains the following settings:

Parameter	Value
Predictive model	'lastmodel'
Input Matlab stream	laststream
Visualization function	bar(y)
Update frequency	10
Form of the produced output values	distribution
Create a figure	<input checked="" type="checkbox"/> (set)
Start-up delay	1
Name of new predictor	lastpredictor

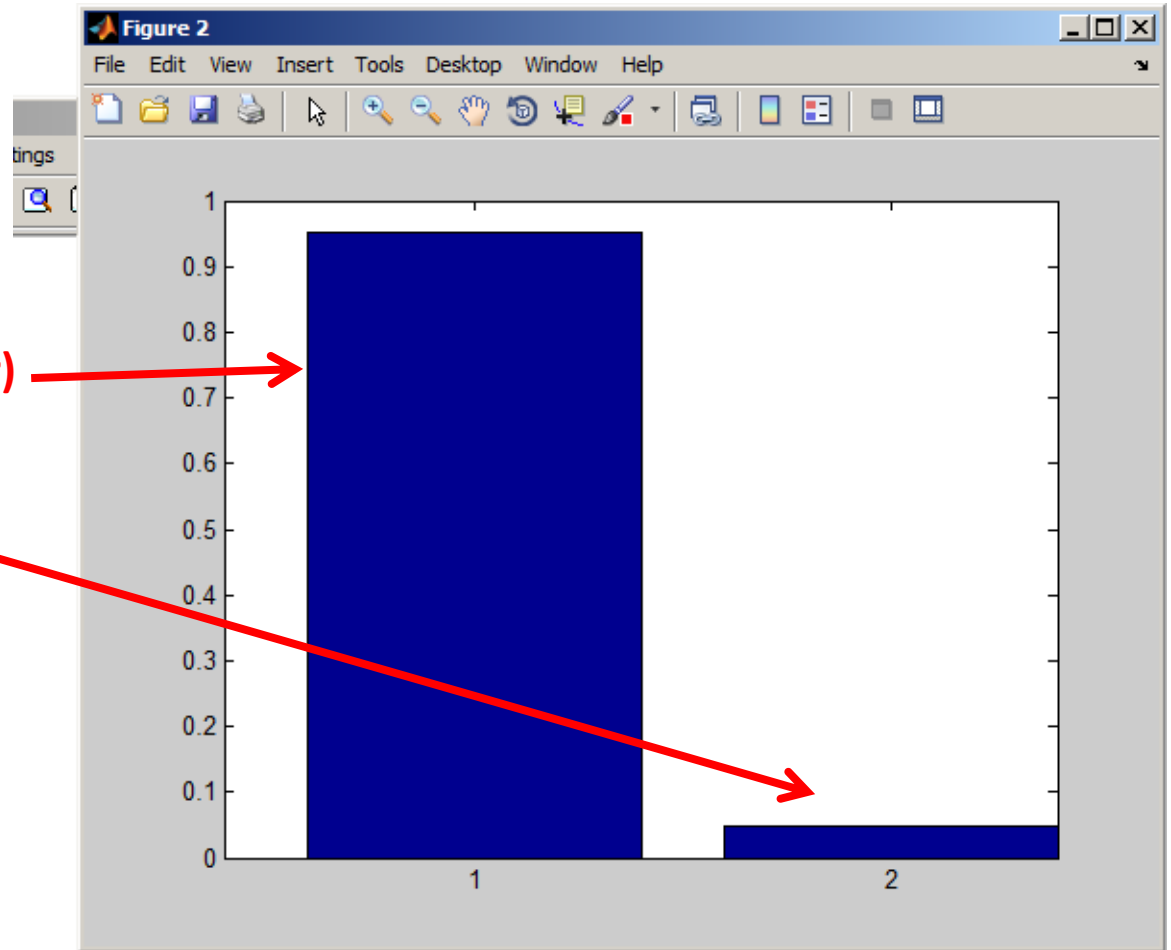
Buttons: Help, Cancel, Ok (highlighted with a red box)



# Visualized Real-Time Outputs

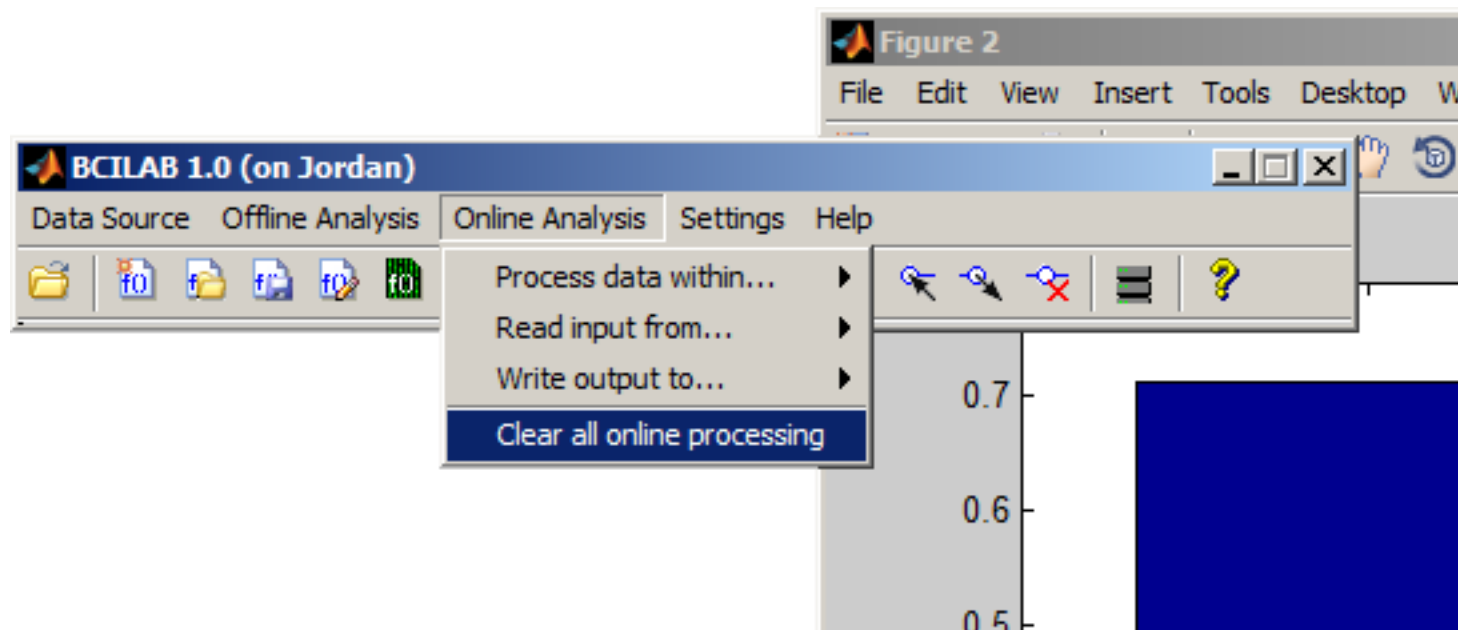
Probability of class 1 (no error)

Probability of class 2 (error)





# Stopping the Online Processing



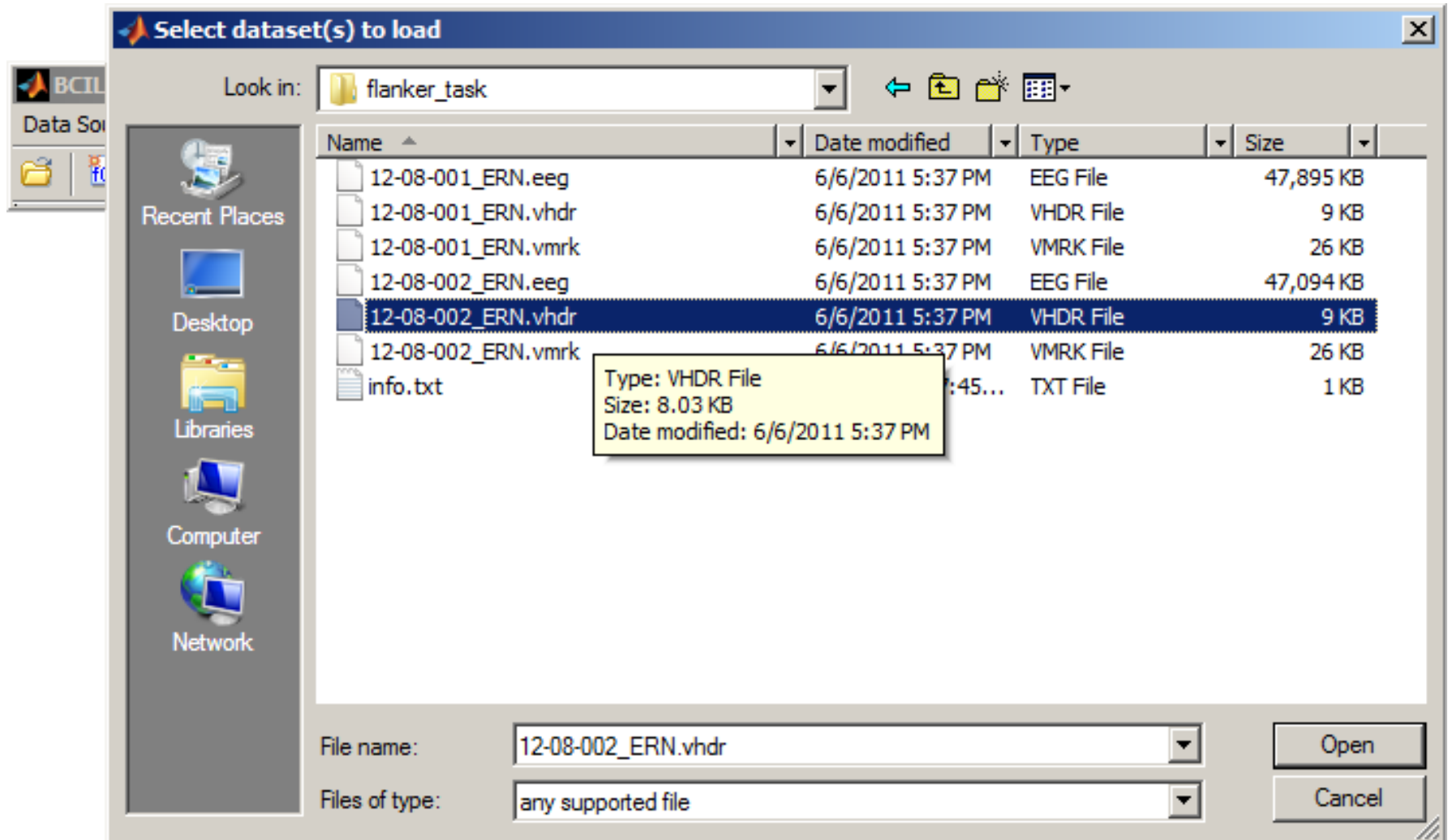


## 4 Customizing Approaches



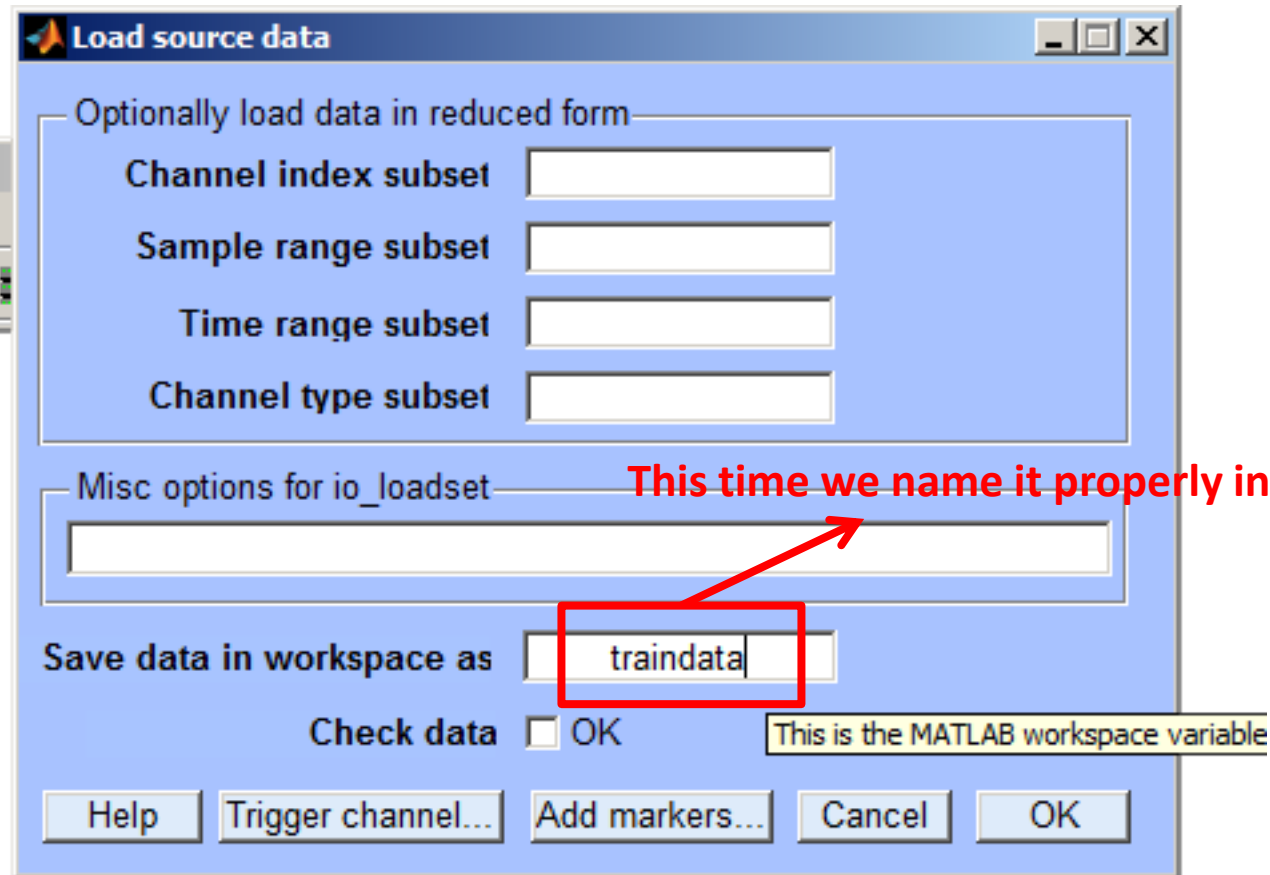


# Loading the Training Data Again



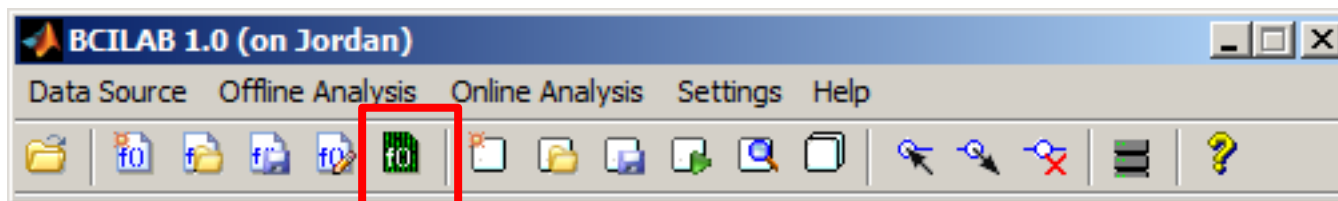


# Loading the Training Data Again





# Editing the Previous Approach In Detail





# Editing the Previous Approach In Detail

**Review/edit approach**

Approach properties

**Miscellaneous**

SignalProcessing	
FilterOrdering	
EOGRemoval	<input type="checkbox"/>
Resampling	<input checked="" type="checkbox"/>
SamplingRate	100
FilterLength	10
StopbandWeight	1
TypeSelection	<input type="checkbox"/>
MarkerInsertion	<input type="checkbox"/>
ChannelSelection	<input type="checkbox"/>
SurfaceLaplacian	<input type="checkbox"/>
Rereferencing	<input type="checkbox"/>
ICA	<input type="checkbox"/>
BandPower	<input type="checkbox"/>
DipoleFitting	<input type="checkbox"/>
IIRFilter	<input type="checkbox"/>
VolumeSelection	<input type="checkbox"/>
WindowCleaning	<input type="checkbox"/>
StationarySubspace	<input type="checkbox"/>
ChannelCleaning	<input type="checkbox"/>
ChannelRepair	<input type="checkbox"/>
BurstCleaning	<input type="checkbox"/>
Projection	<input type="checkbox"/>
FIRFilter	<input type="checkbox"/>
SparseReconstruction	<input type="checkbox"/>
Standardization	<input type="checkbox"/>
EpochExtraction	<input checked="" type="checkbox"/>
TimeWindow	[-0.2 0.8]
BaselineRemoval	<input type="checkbox"/>

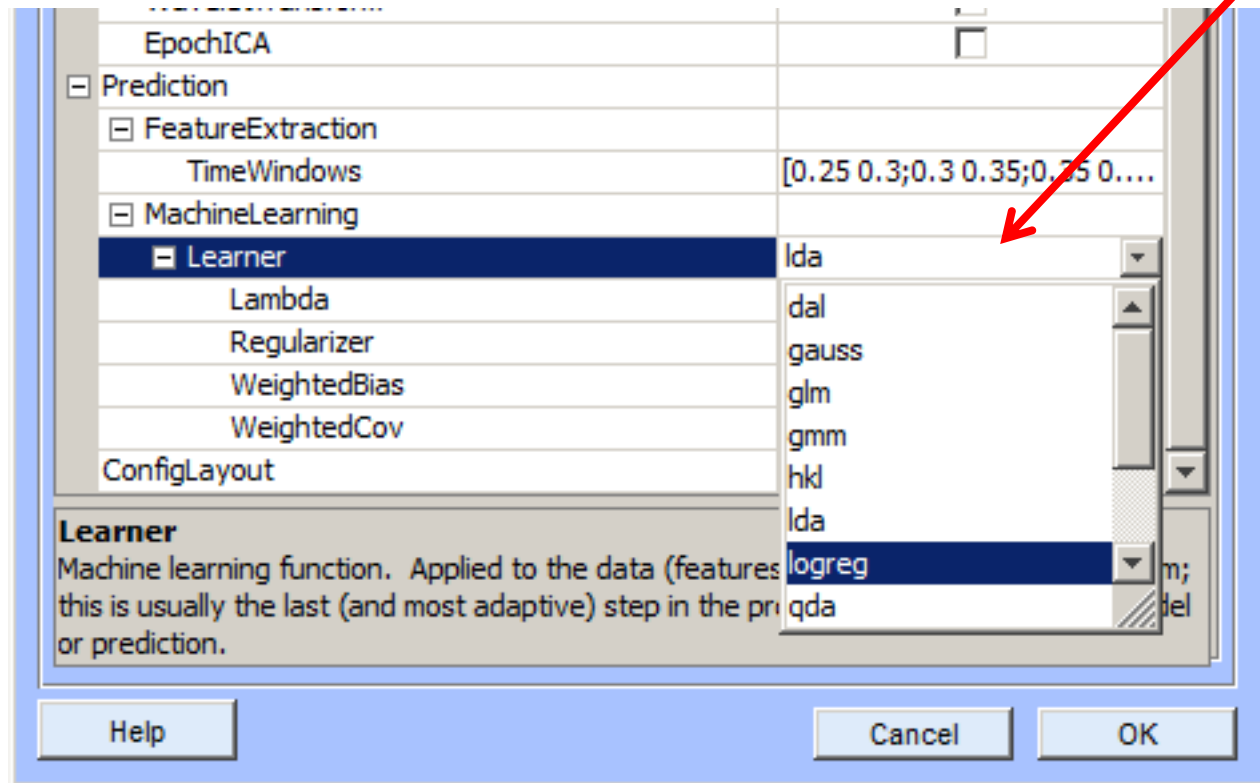
**(Name)**  
(Description)

Help Cancel OK



# Changing the Classifier

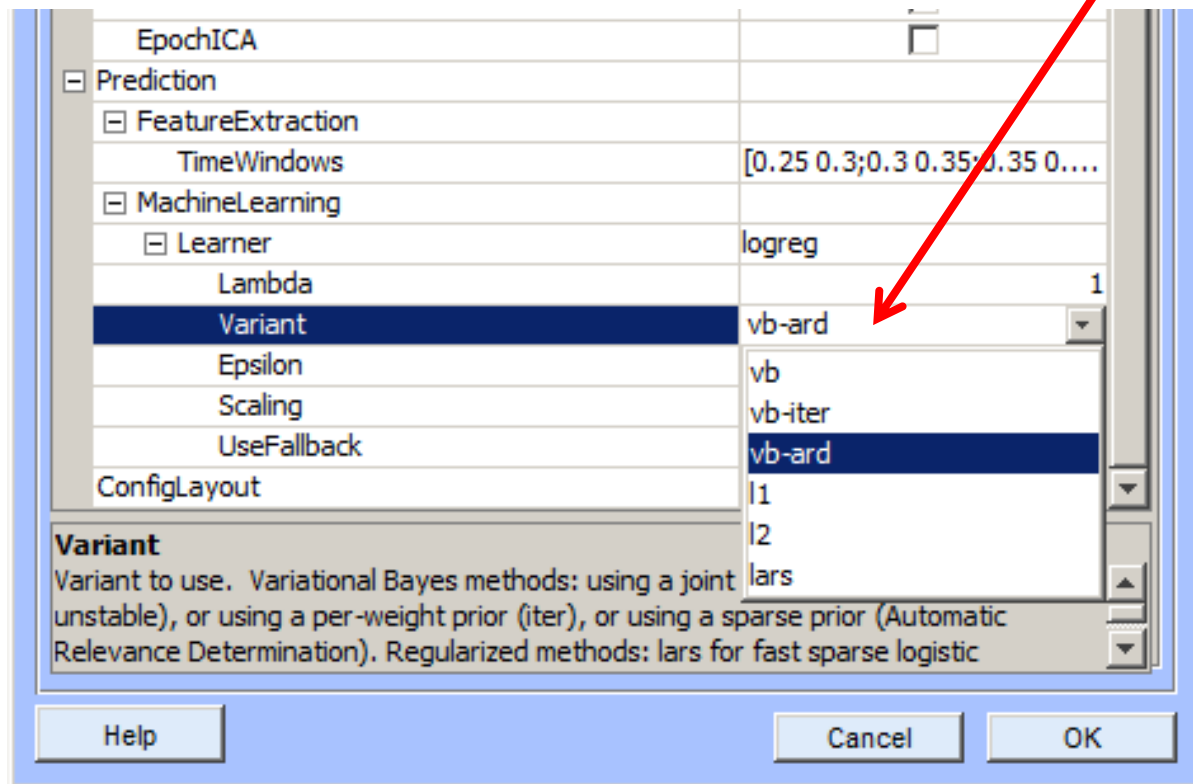
Instead of LDA we choose logistic regression.





# Changing the Classifier

We don't use the VB (Variational Bayes) variant but the sparse version (with Automatic Relevance Determination).





# Learning a New Model...

**Calibrate a model**

**Selected approach** lastapproach ("Windowed Means ...")

**Calibration data source** lastdata ("12-08-001\_ERN.vhdr")

**Target markers** [Loaded via EEGLAB]  
[Loaded via BCILAB]  
lastdata ("12-08-001\_ERN.vhdr")  
testdata ("12-08-001\_ERN.vhdr")  
traindata ("12-08-002\_ERN.vhdr")

**Parameter Search**

**Loss/Performance Metric**

**Cross-validation folds**

**Spacing around test trials** 5

**Performance estimates**

☒ **Compute performance estimates**

**Cross-validation folds** 5

**Spacing around test trials** 5

**Computing resources**

☐ **Run on a computer cluster**

**Node pool** (use current config)

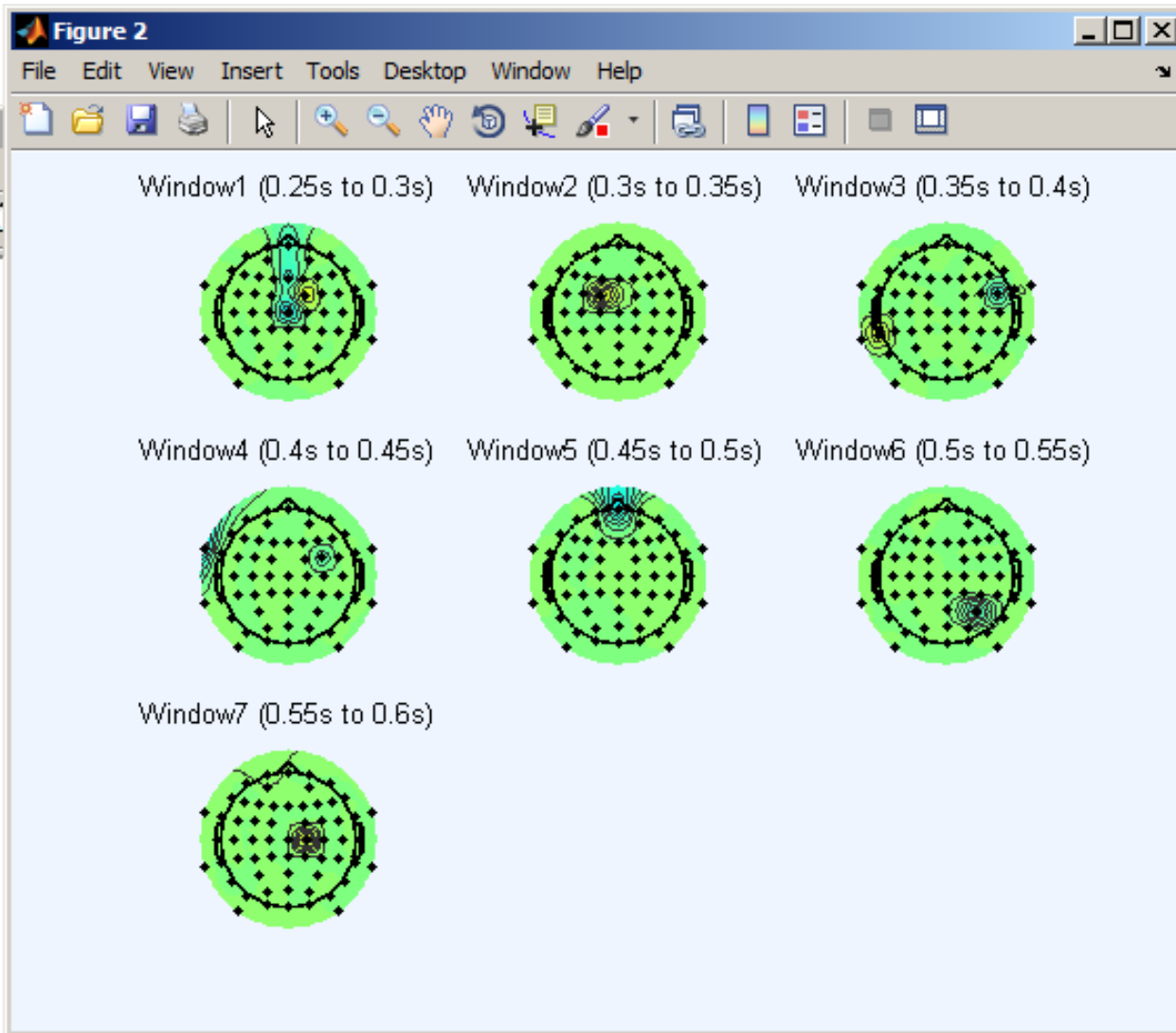
**Save model in workspace as** lastmodel

**Save stats in workspace as** laststats

**Help** **Cancel** **OK**



# Visualizing The Model



**This model uses  
a minimal  
subset of  
channels.**



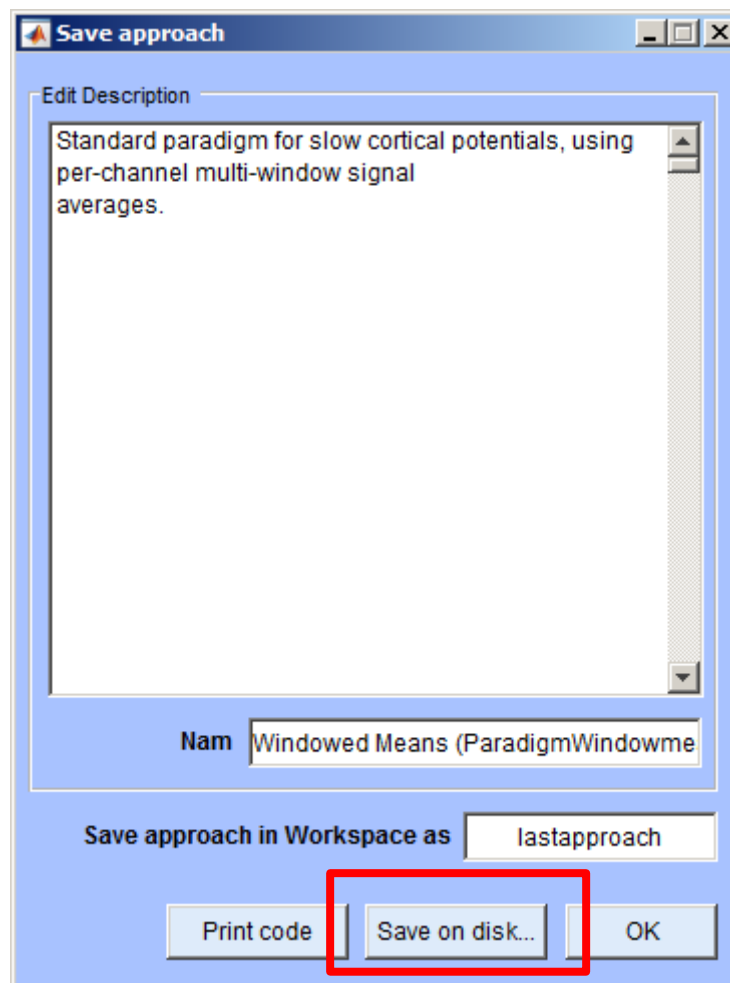


# Saving the Approach for Later



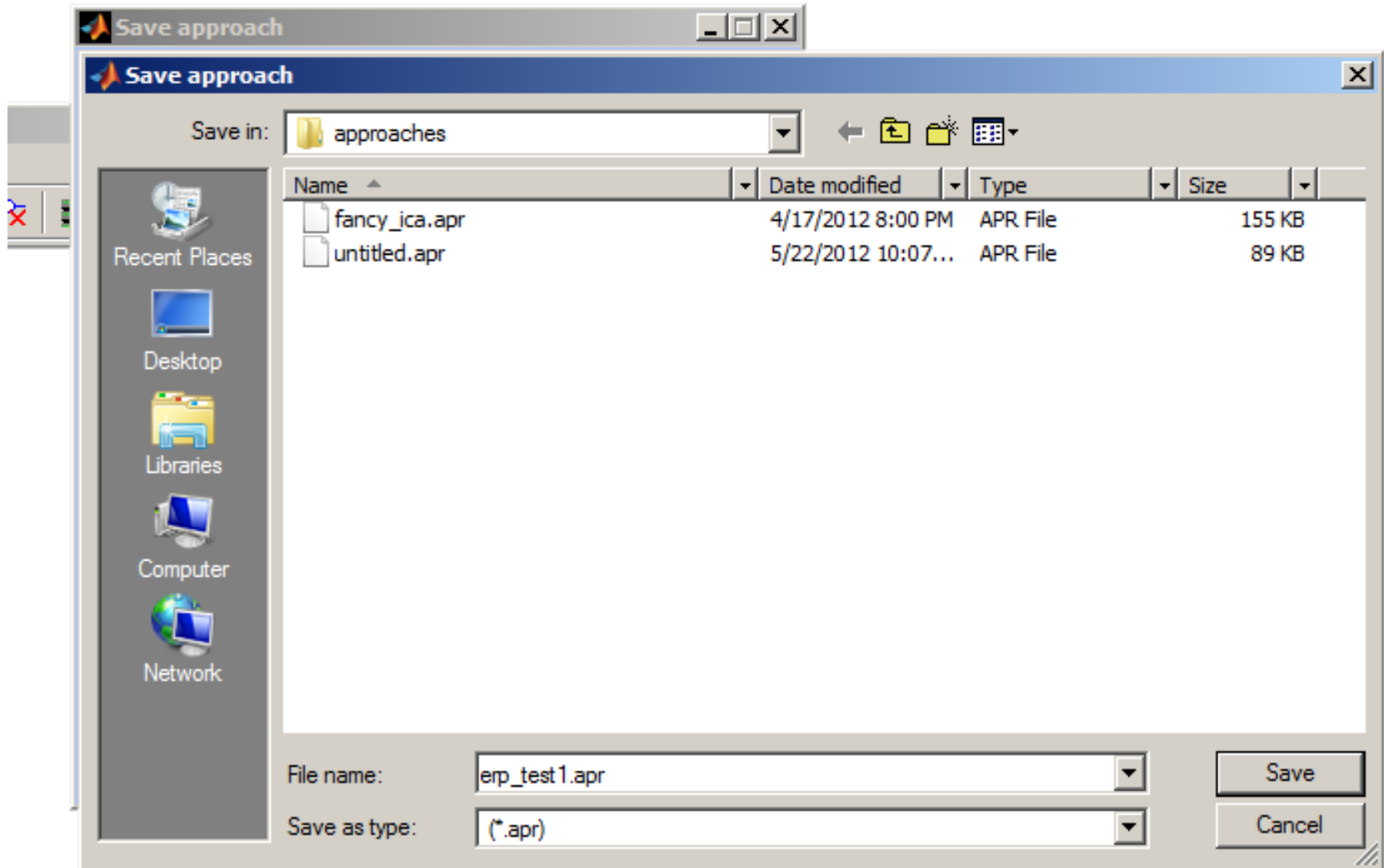


# Saving the Approach for Later





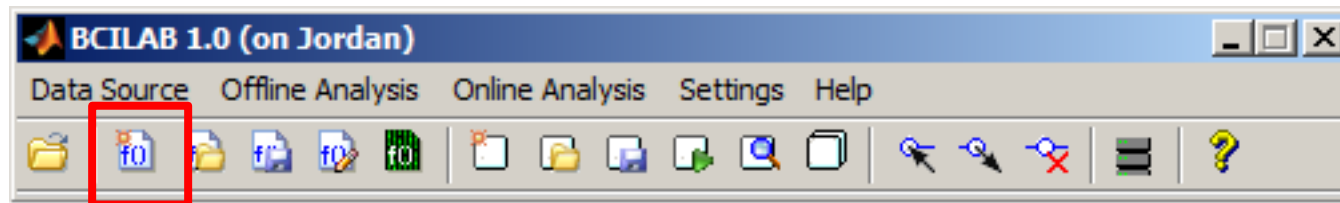
# Saving the Approach for Later





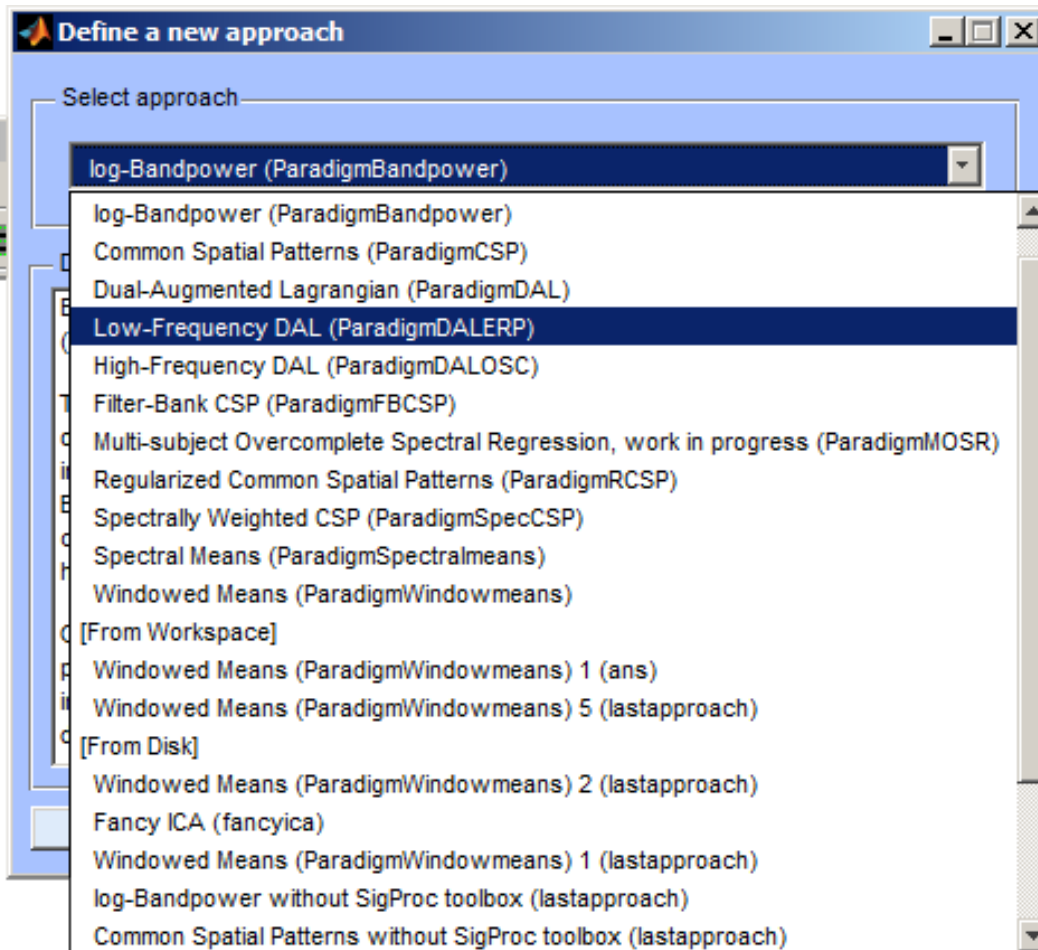
## 5 More Analyses: Using an Advanced ERP Paradigm

Note: this computation uses a method that will run for up to 15 minutes.





# Selecting DAL-ERP



This is one of the best known approaches for ERP-based BCIs. It assumes that there is a small set of latent spatial sources with their own characteristic time course weights, and learns *both simultaneously*.



# Configuring DAL-ERP

**BCILAB: Configure approach**

New sampling rate of the data	60
Frequency specification of the filter	[0.1 0.5]
Filter type	butterworth
Epoch time window relative to the target markers	[-0.2 0.8]
Frequency-domain selection	[0.1 15]
Regularisation parameters	[1024 861.077929219804 724.0]
Loss function to be used	logistic
Type of regularisation to use	dual-spectral

Help Cancel **Ok**



# Calibrate Model...

**Calibrate a model**

**Selected approach** lastapproach ("Low-Frequency DA...")

**Calibration data source** traindata ("12-08-002\_ERN.vhdr")

**Target markers** {'S101','S102'},{'S201'}

**Parameter Search**

**Loss/Performance Metric** Automatically chosen

**Cross-validation folds** 5

**Spacing around test trials** 5





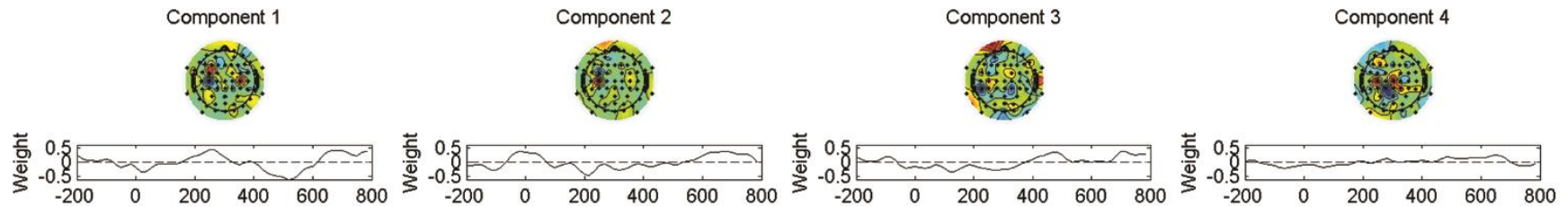
# Wait for a Few Minutes...

```
beginning evaluation...

beginning new computation...
pop_epoch():398 epochs selected
Epoching...
pop_epoch():398 epochs generated
eeg_checkset: found empty values for field 'target'
              filling with values of other events in
pop_epoch(): checking epochs for data discontinuity
learning ensemble...
  scanning lambda = 1024.000000... model rank = 0
  scanning lambda = 861.077929... model rank = 0
  scanning lambda = 724.077344... model rank = 0
  scanning lambda = 608.874043... model rank = 0
  scanning lambda = 512.000000... model rank = 0
  scanning lambda = 430.538965... model rank = 1
  scanning lambda = 362.038672... model rank = 1
  scanning lambda = 304.437021... model rank = 1
  scanning lambda = 256.000000... model rank = 1
  scanning lambda = 215.269482... model rank = 1
  scanning lambda = 181.019336... model rank = 1
  scanning lambda = 152.218511... model rank = 1
  scanning lambda = 128.000000... model rank = 2
  scanning lambda = 107.634741... model rank = 3
  scanning lambda = 90.509668... model rank = 3
  scanning lambda = 76.109255... model rank = 3
  scanning lambda = 64.000000... model rank = 3
  scanning lambda = 53.817371... model rank = 3
  scanning lambda = 45.254834... model rank = 4
  scanning lambda = 38.054628... model rank = 6
  scanning lambda = 32.000000...>> |
```



# Visualizing the Model



**Note that these are the spatial filters rather than the forward projections!**