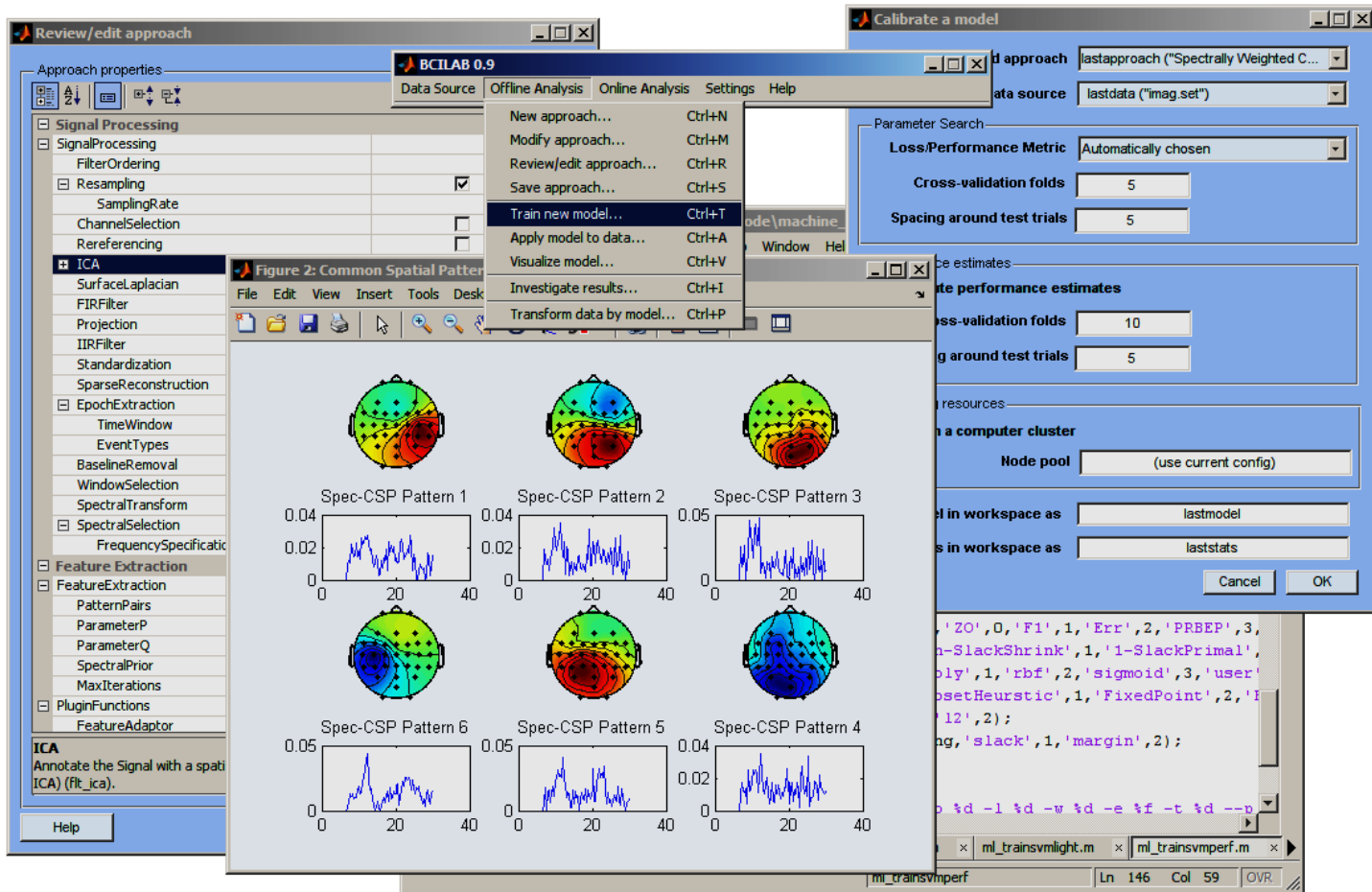


These slides are at:

<ftp://sccn.ucsd.edu/pub/bcilab/mallorca>

BCILAB Briefing



The screenshot displays the BCILAB 0.9 software interface with several windows open:

- Review/edit approach:** Shows approach properties such as Signal Processing, ICA, and Feature Extraction.
- Calibrate a model:** A dialog box for configuring model training, including parameters like Loss/Performance Metric, Cross-validation folds (set to 5), and Spacing around test trials (set to 5).
- Figure 2: Common Spatial Patterns:** A window displaying six topographic maps and corresponding time-frequency plots for Spec-CSP Patterns 1 through 6. Each plot shows power over time (0 to 40).
- Main Menu:** The BCILAB 0.9 menu is open, showing options like 'Train new model...' (Ctrl+T) and 'Apply model to data...' (Ctrl+A).
- Code Editor:** A window showing MATLAB code for model training, including parameters like 'rbf', 'sigmoid', and 'margin'.

Idea & Purpose

- Like EEGLAB, but for BCI (and/or cognitive state assessment)
 - Seeding a community
 - Strengthening links between BCI and Neuroscience
- SCCN's in-house tool for BCI problems
 - Main focus: Advanced cognitive monitoring
 - Part of a large US research program (CaN CTA)
 - Funded by ARL (and ONR, Swartz Foundation, ...)



BCILAB Specialty

- State of the art
- Largest collection of machine learning & signal processing components in any open-source BCI package
 - Many standard components (CSP, LDA, SVM, ...)
 - Many modern components (SBL, SSA, AMICA, HKL, DPGMM, LR-DAL, ...)
 - Some novel components (OSR, RSSD, SSB, ...)
- Next-generation framework
 - Fully probabilistic
 - Model inference from data corpora*
 - Anatomical priors, other neuroscience-aware features
 - Processing of parallel streams

(*: not yet in the current release)

BCILAB Components

Framework

GUI / Scripting Interfaces

Approach
Definition

Online
Execution

Offline
Evaluation

Visualization

Plugins

Signal Processing

ICA

SSA

FIR

IIR

FFT

...

Machine Learning

LDA

QDA

DAL

GMM

SVM

...

BCI Paradigms

CSP

Spec-CSP

ERP

RSSD

...

Devices

TCP

OSC

BCI2000

...

Infrastructure

GUI
generation

cluster
computing

disk
caching

helper
functions

environment
services

Dependencies

CVX

BNT

EEGLAB

GUI utils

LIBSVM

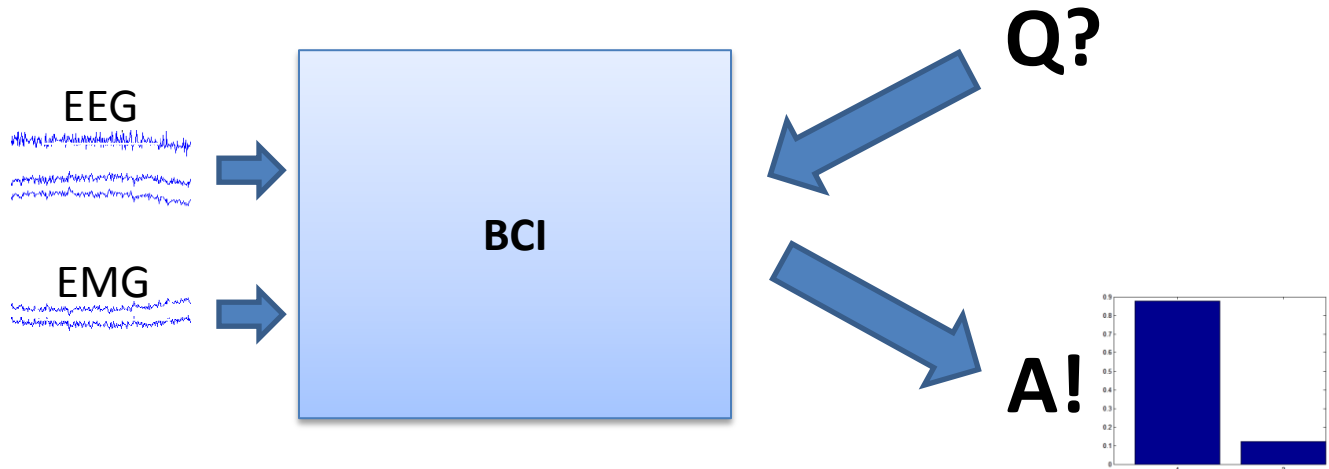
GLMNET

...

Driver
I/O

BCI Behavior

- BCIs in BCILAB are acting as an oracle that consumes one or more biosignals and can respond to (pre-defined) queries about cognitive state





Signal Processing?

- Some signal-level computations can be done more efficiently than window-by-window (esp. when successive windows overlap a lot)

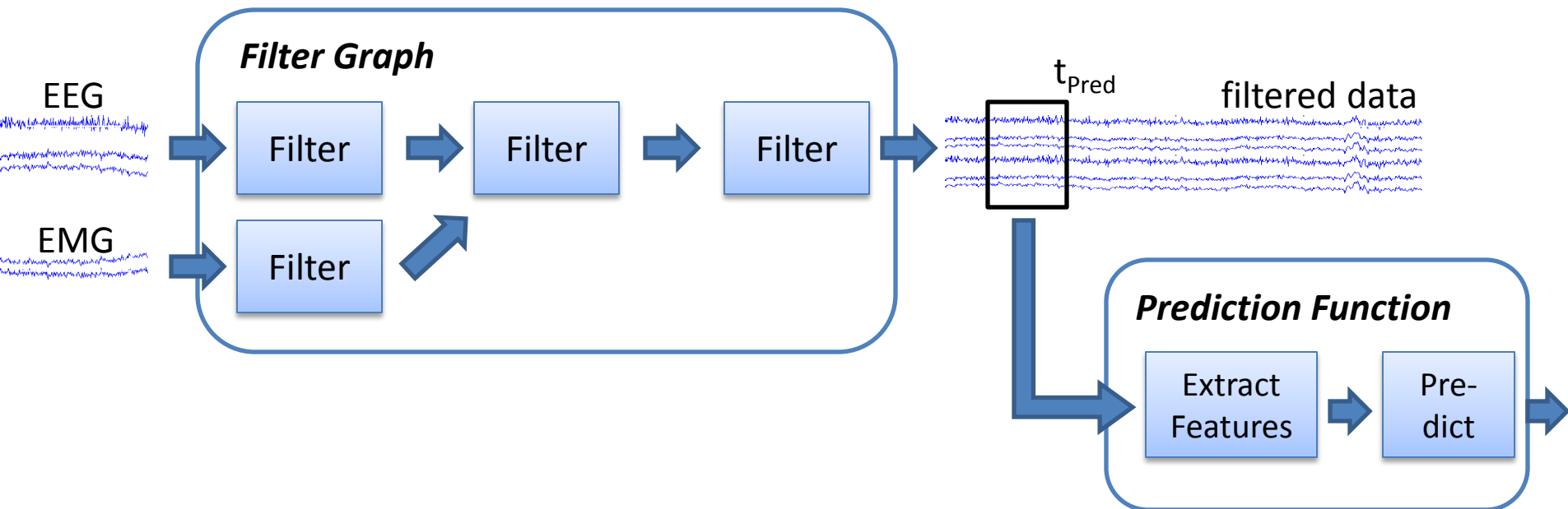
Signal Processing?

- Some signal-level computations can be done more efficiently than window-by-window (esp. when successive windows overlap a lot)
- Room for good DSP use (e.g., frequency filter, spatial filter, ...) before actual prediction
- Also, can assemble approaches from existing components



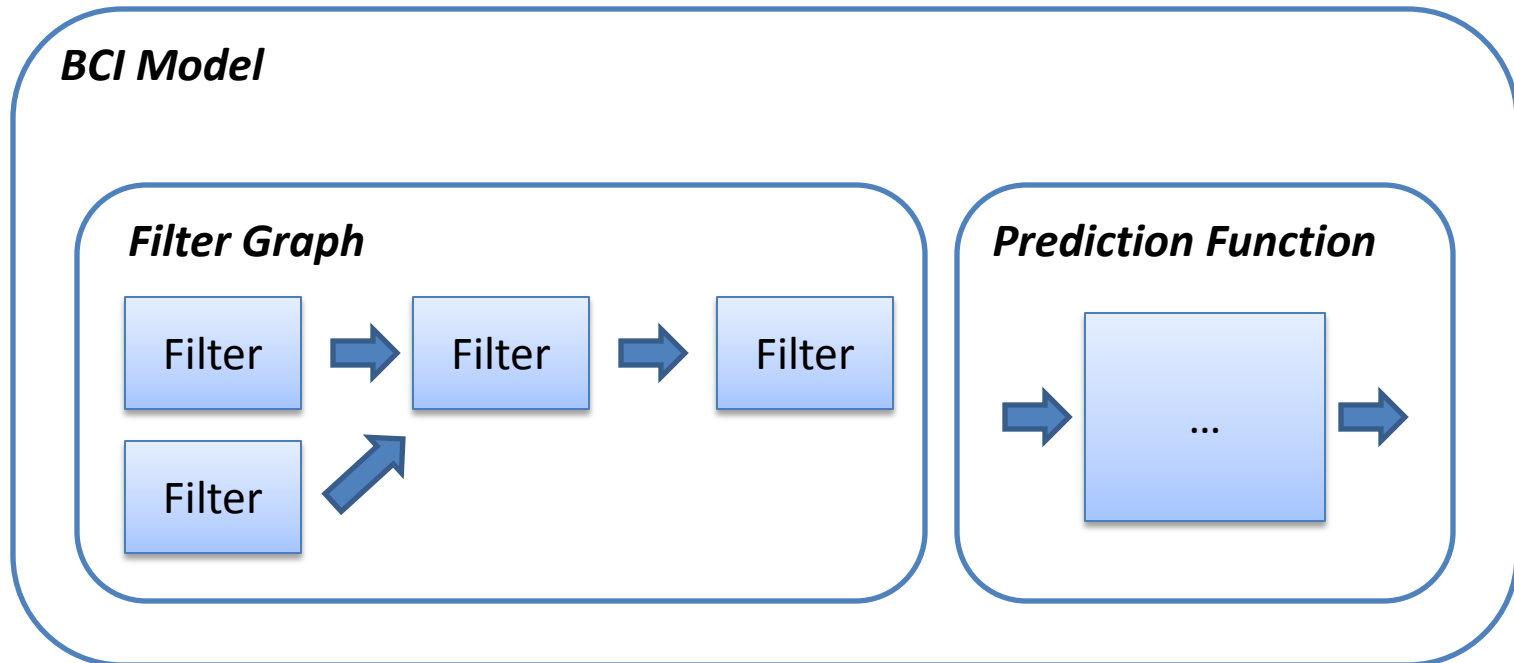
Online Data Flow

- A filter graph receives all input samples and produces pre-filtered data (signal flows through it)
- The prediction function may be queried on demand on the filter graph's outputs



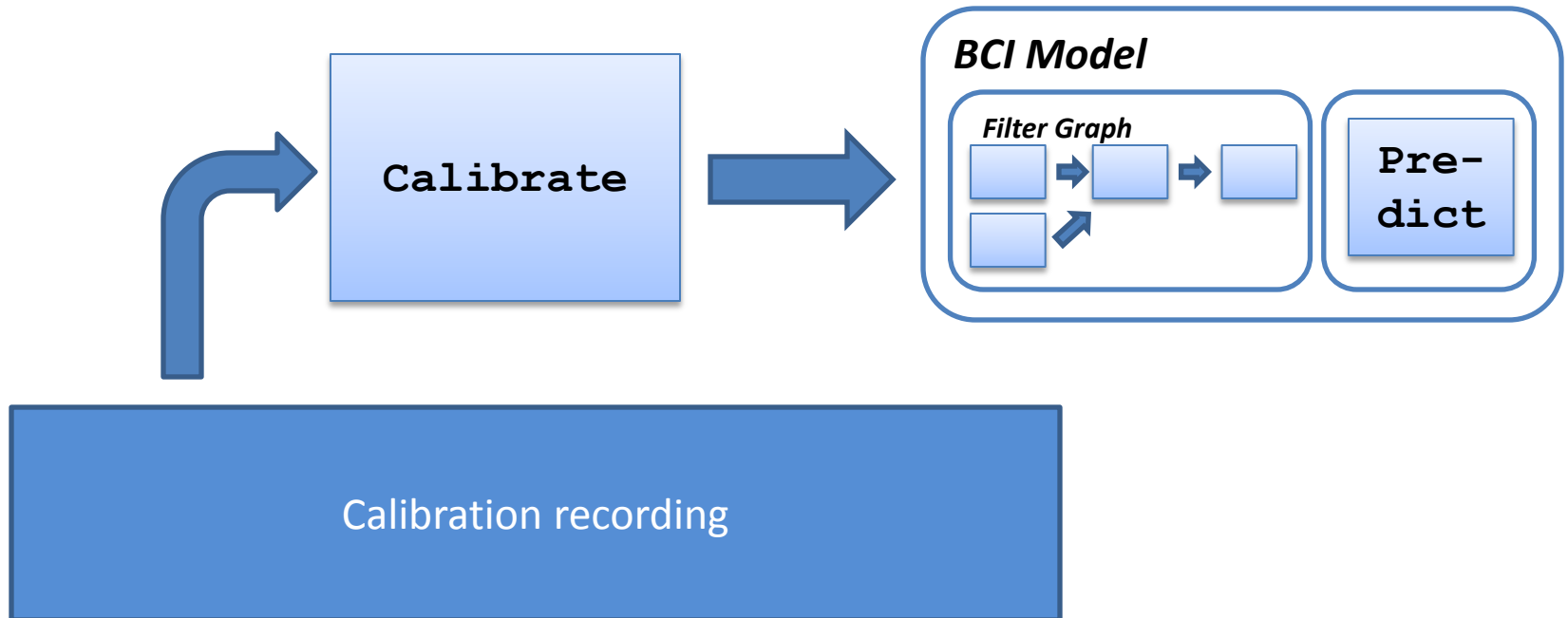
BCI Models

- BCIs are described by “BCI models” that specify both the *filter graph* and the *prediction function* (incl. parameters)



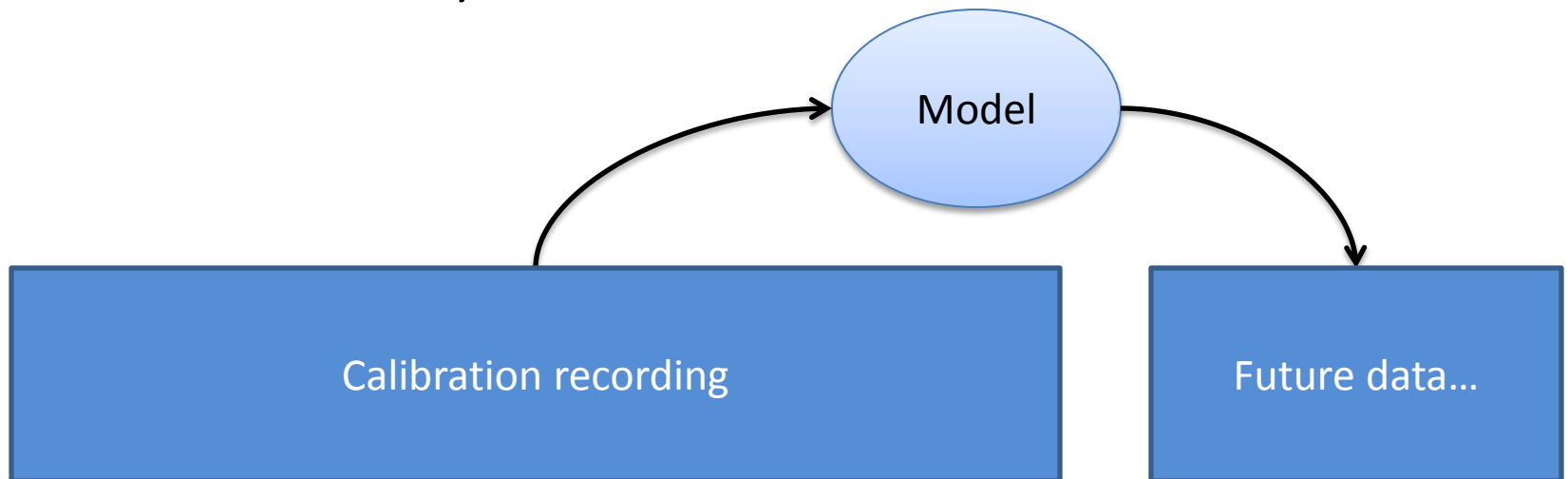
BCI Paradigms

- BCI paradigms are the coarsest plugin type in BCILAB and tie all parts of a BCI approach together
- They are seeds for new BCI designs and cornerstones of BCILAB usage



Offline Evaluation

- Given calibration data
- Estimate model parameters (spatial filters, statistics)
- Apply the model to new data (online / single-trial)
- Optionally: compare outputs with known state, compute loss statistics for the model / approach (e.g., misclassification rate)



Offline Evaluation

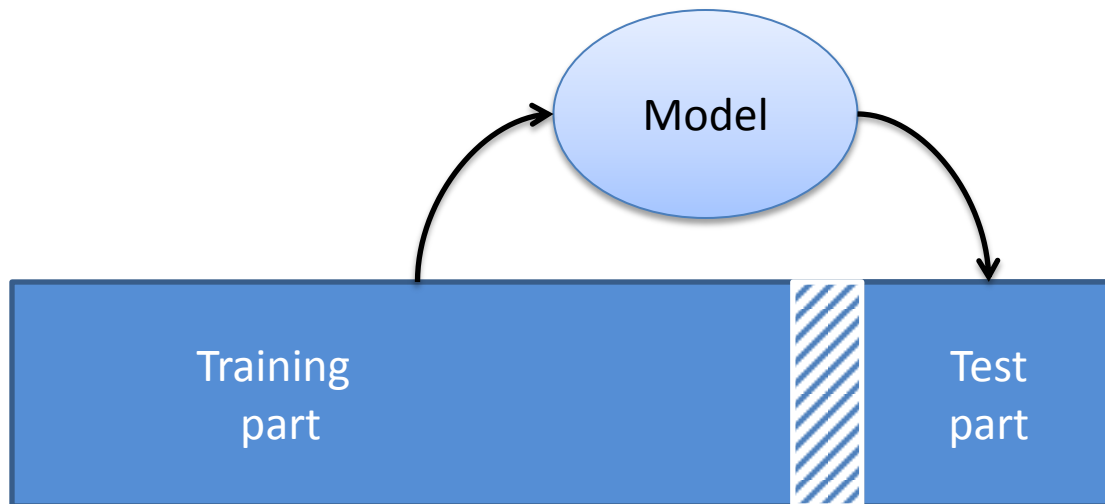
- Evaluation of computational approaches on a **single** data set?

?

Calibration recording

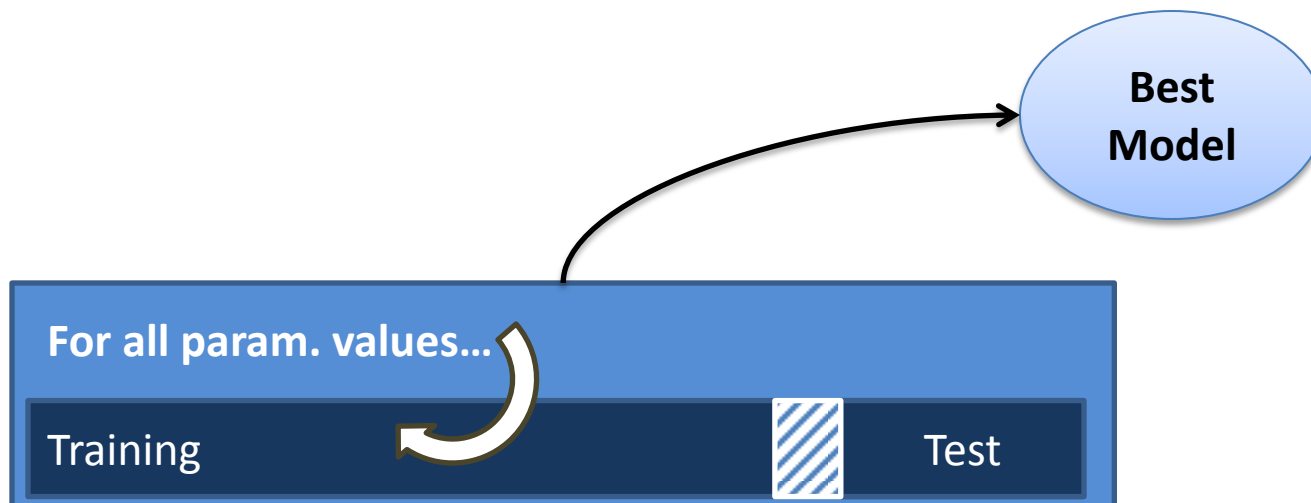
Offline Evaluation

- Evaluation of computational approaches on a single data set?
 - Can not test on the training data! (always on separate data)
 - Instead can split data set repeatedly into training/test blocks systematically, a.k.a. *cross-validation*



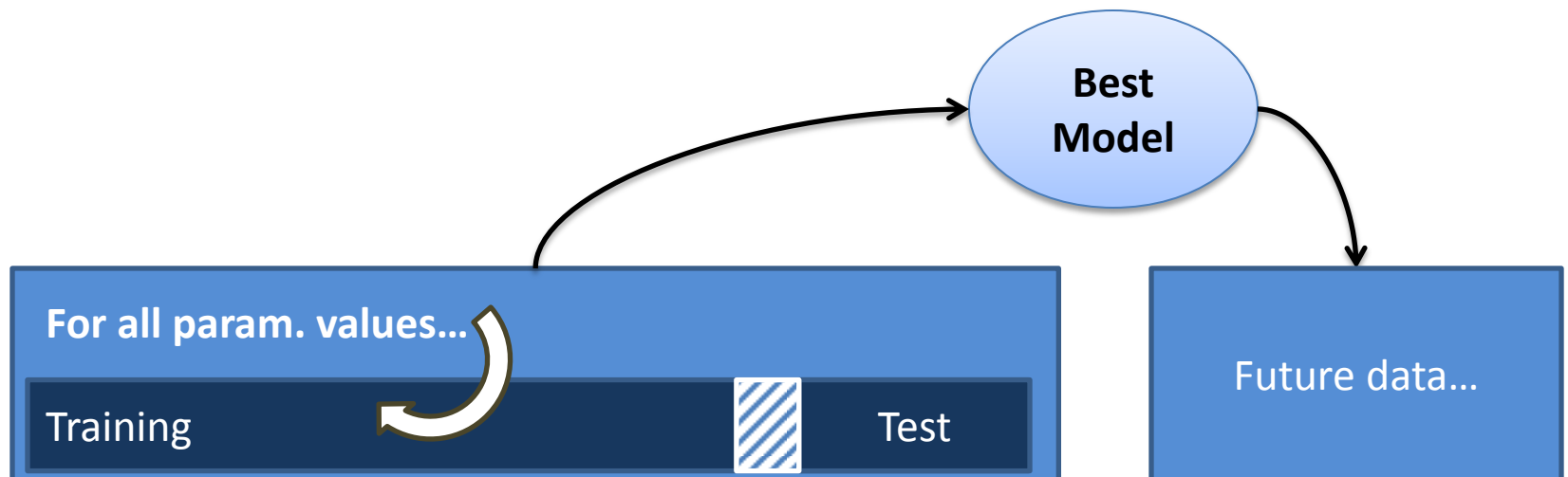
Resolving Free Parameters

- Can be done using cross-validation in a grid search (try all values of free parameters)
- **Caveat:** Resulting “optimal” numbers are *non-reportable* (cherry-picked!)



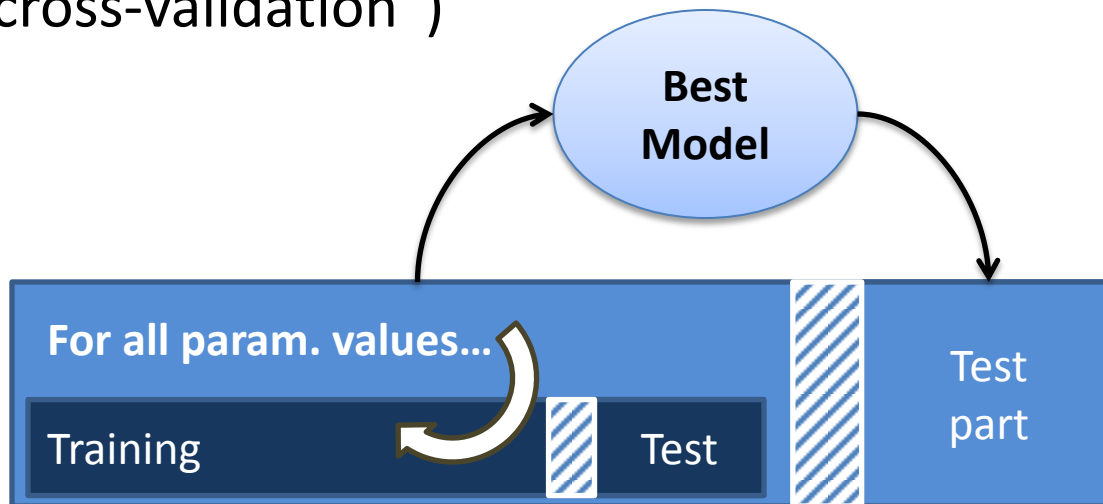
Resolving Free Parameters

- Can be done using cross-validation in a grid search (try all values of free parameters)
- **Caveat:** Resulting “optimal” numbers are *non-reportable* (cherry-picked!)
- But may test resulting best model on separate data



Resolving Free Parameters


- Can be done using cross-validation in a grid search (try all values of free parameters)
- **Caveat:** Resulting “optimal” numbers are *non-reportable* (cherry-picked!)
- But may test resulting best model on separate data
- **Or** run grid search *within* an outer cross-validation (“nested cross-validation”)



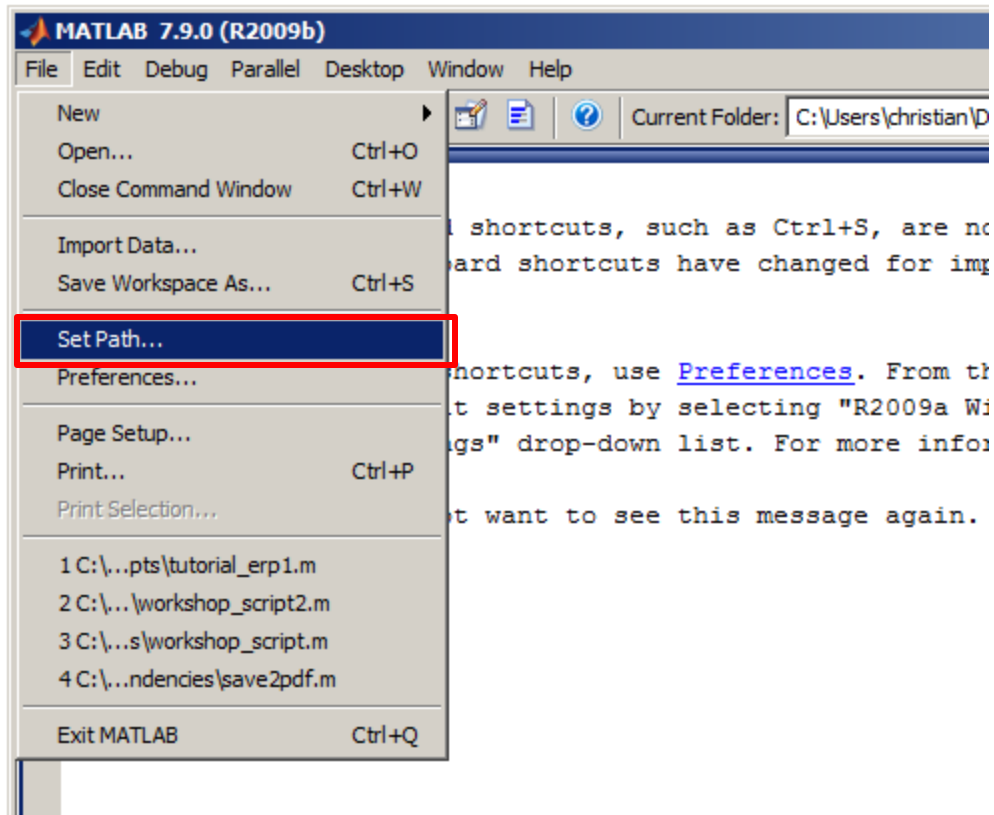
System requirements

- 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, Real-time experimentation environment (DataRiver, BCI2000, OpenViBE or your own)
- 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)

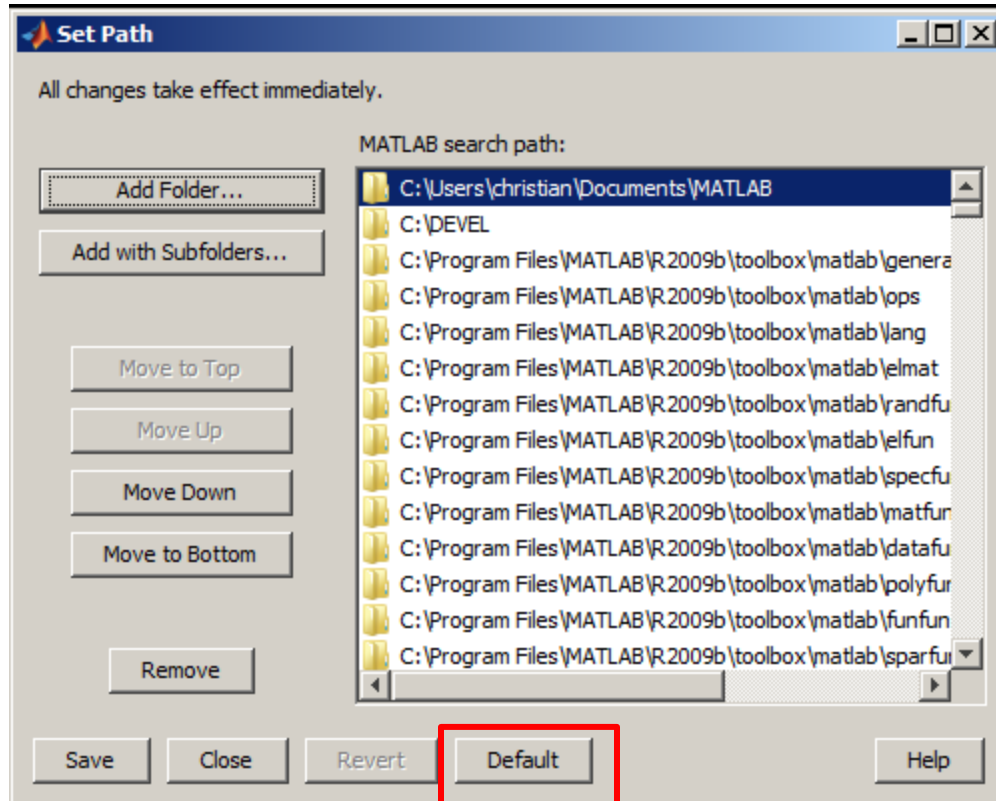
When Processing your own Data

- 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 for the time points with known target condition
 - BCILAB needs raw (unprocessed) data
 - Make sure you have a file format supported by EEGLAB
- Rawr!
- 

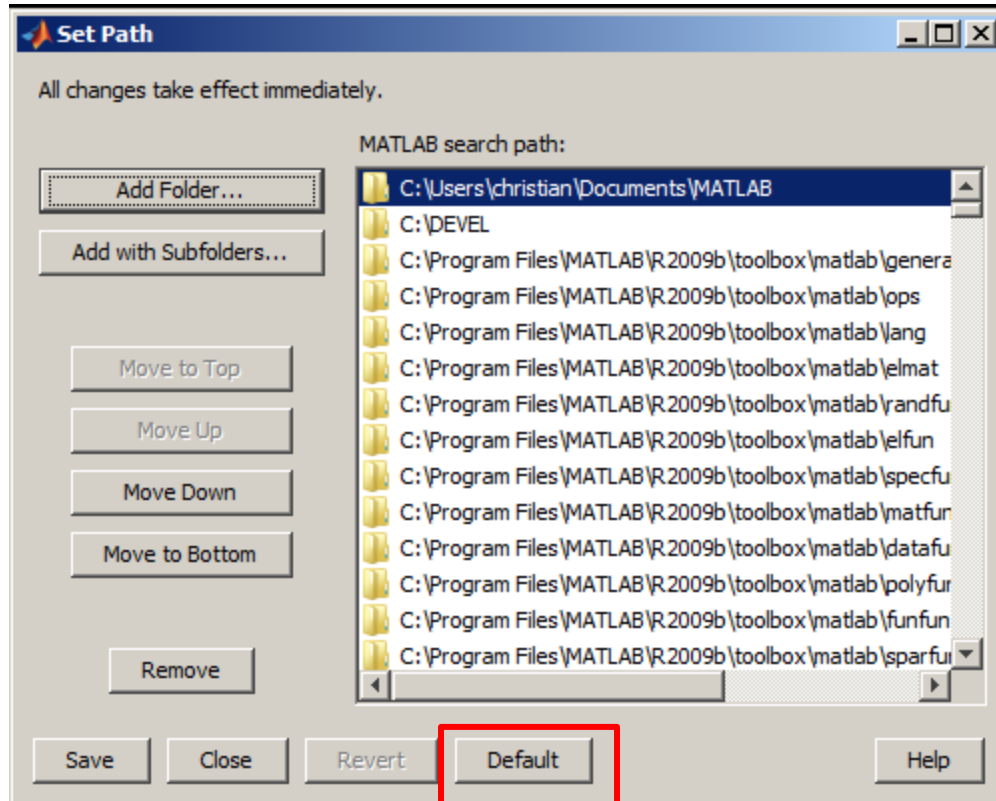
Clearing the Path



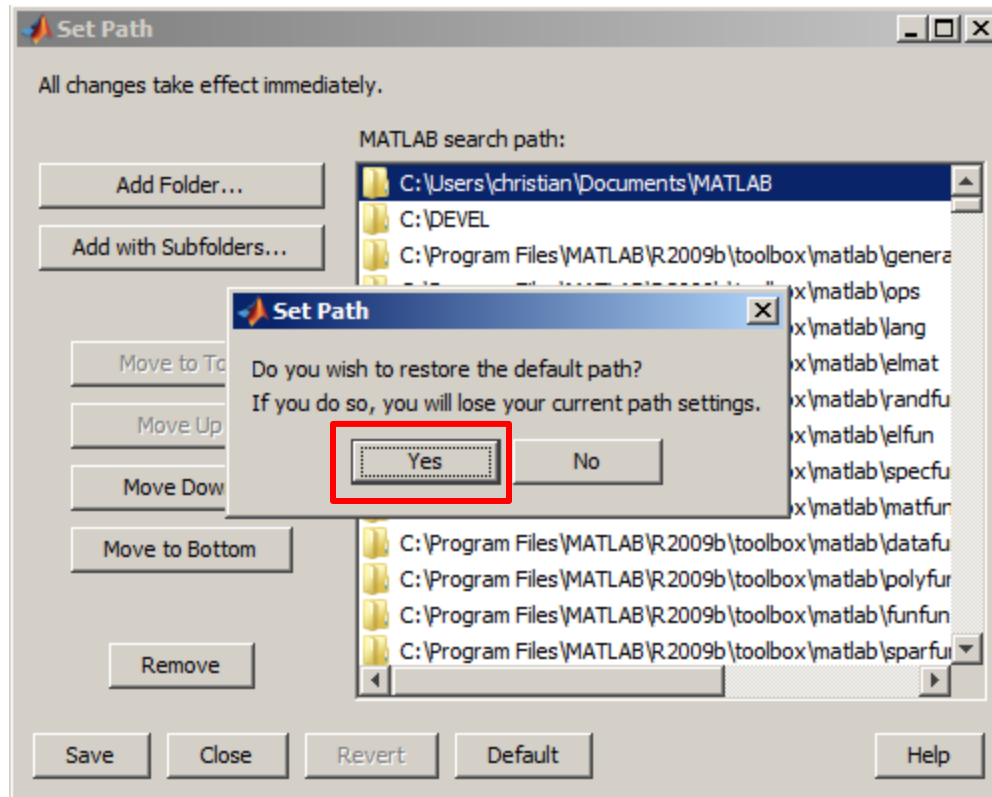
Clearing the Path



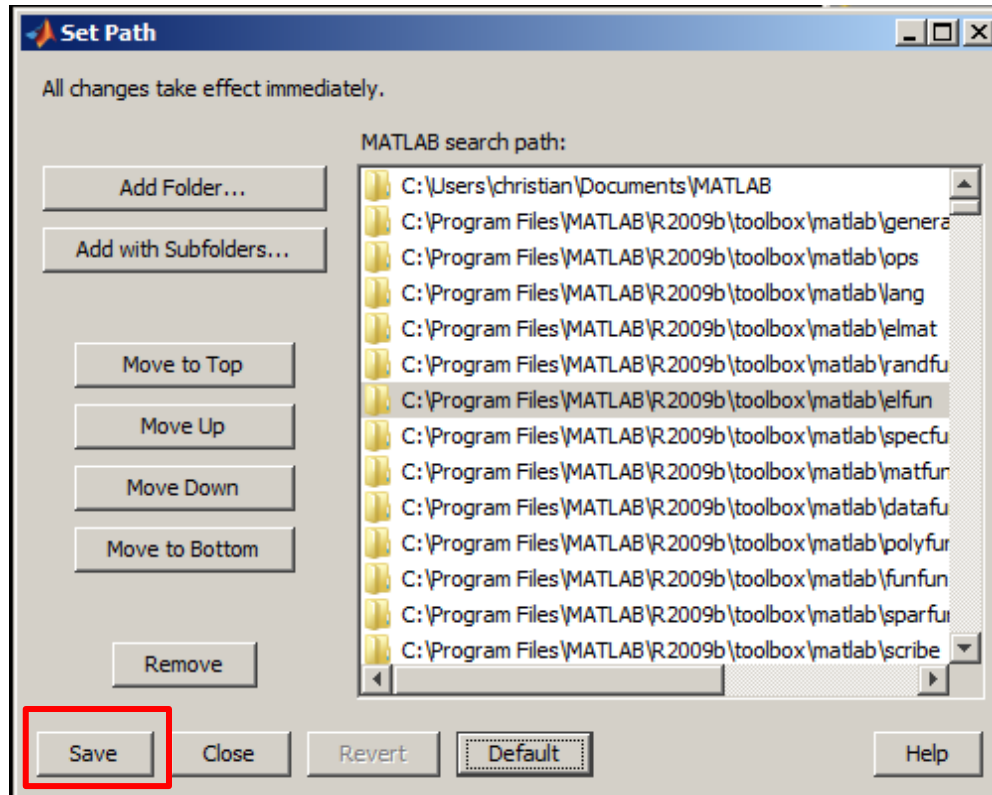
Clearing the Path



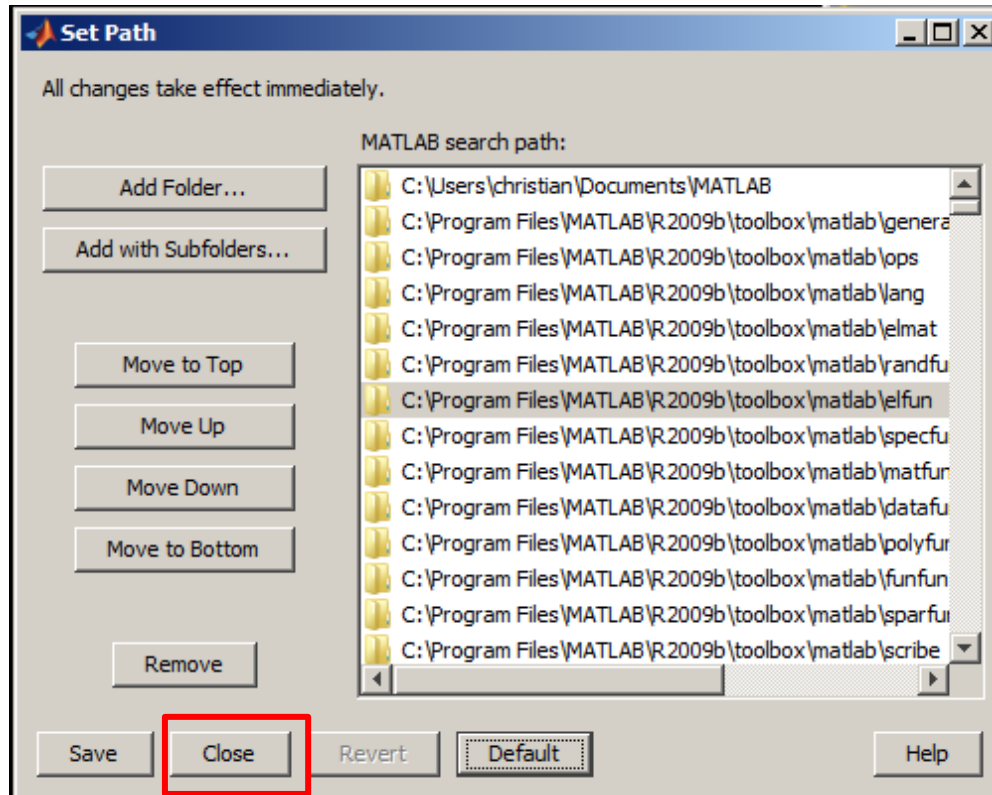
Clearing the Path



Clearing the Path

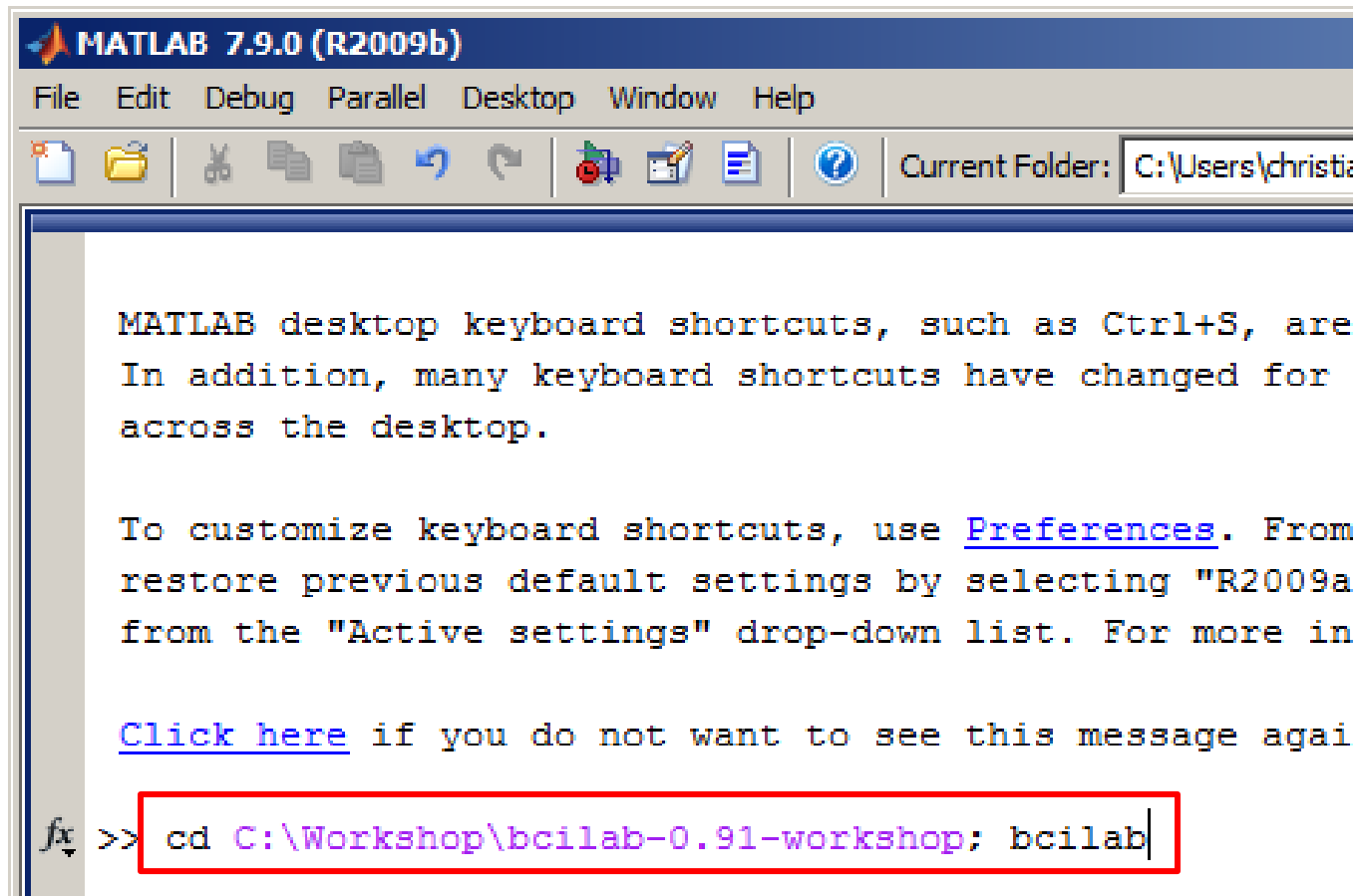


Clearing the Path



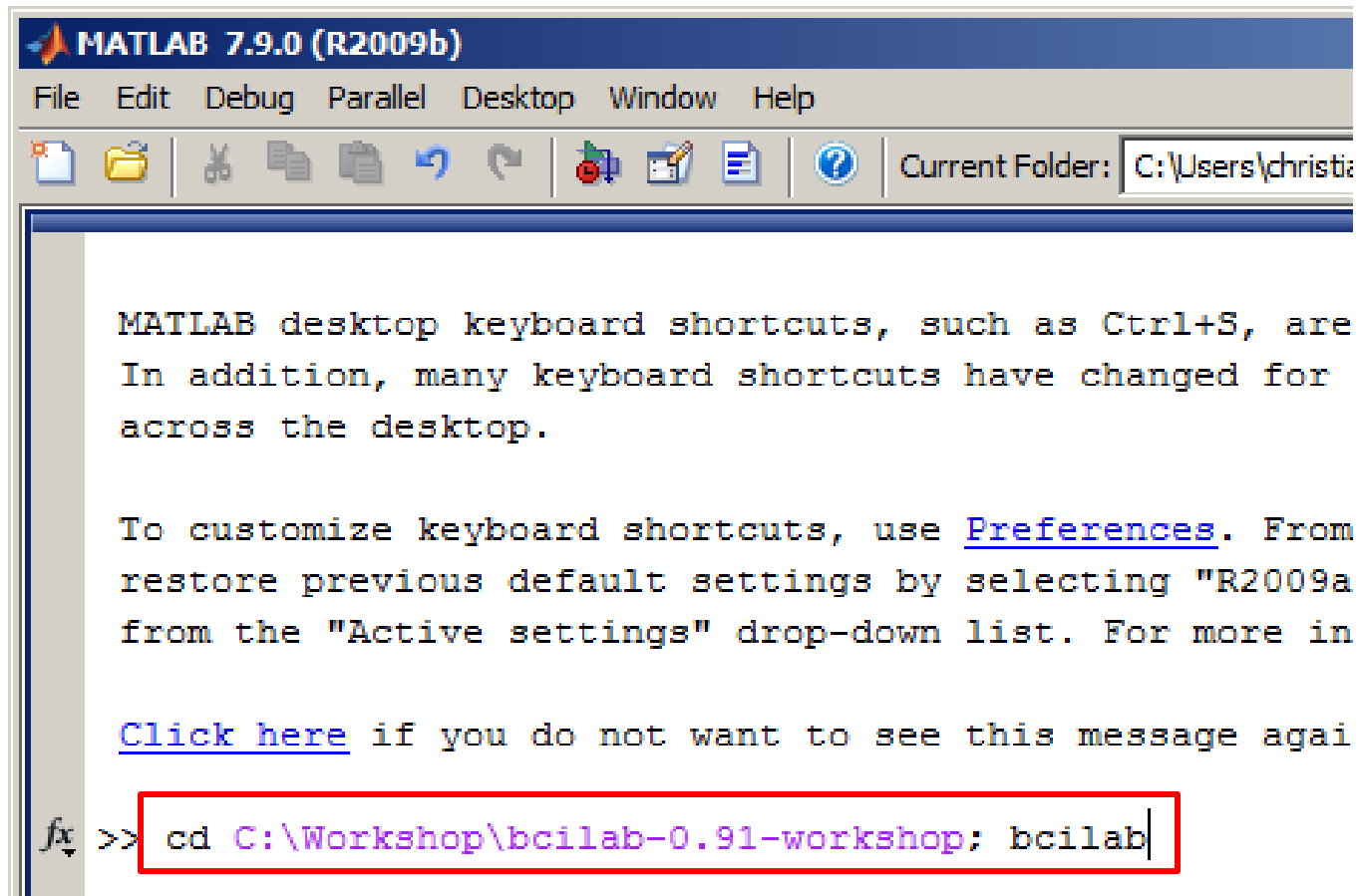
Starting the Toolbox

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



Starting the Toolbox

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



Starting the Toolbox

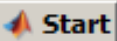
- You should see the welcome message

```
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

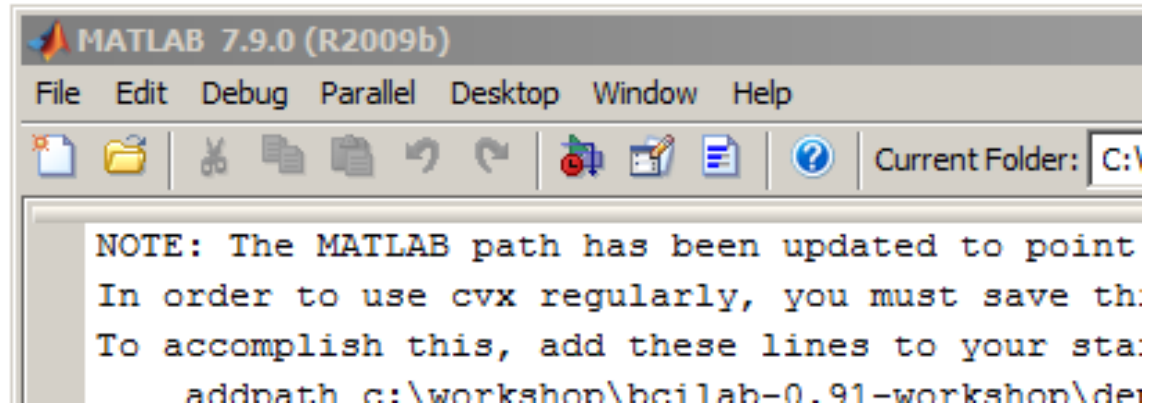
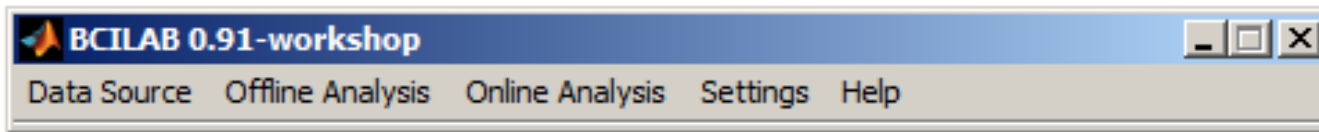
Welcome to the BCILAB toolbox!

fx >>
```



Starting the Toolbox

- ... and the main menu

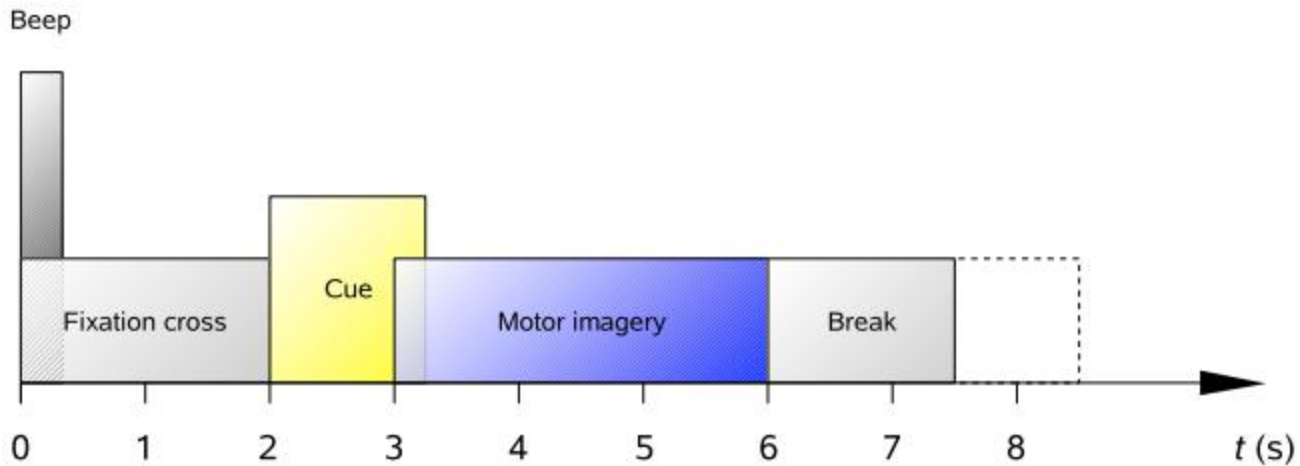


The Data

- Recorded at the famous BCI Lab in Graz, Austria
- Part of the BCI competition IV (“dataset 2a”)
- More info:
http://www.bbci.de/competition/iv/desc_2a.pdf
- Contains imagined movements of 4 classes:
 - Left hand
 - Right hand
 - Feet
 - Tongue
- 2 Training sets, 2 evaluation sets

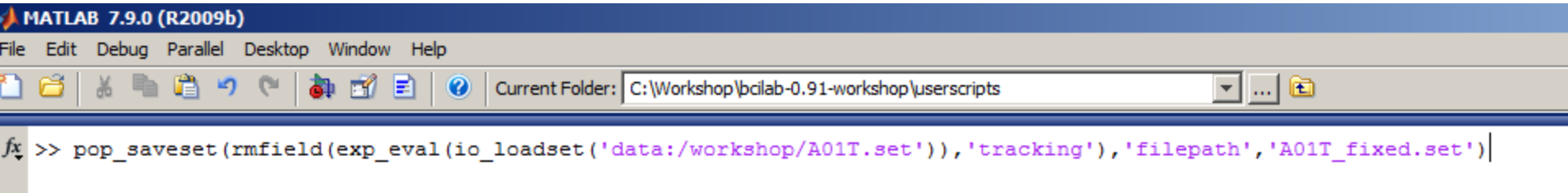
The Data

- Timing overview



Fixing the Data 😊

- Execute the following line:



MATLAB 7.9.0 (R2009b)

File Edit Debug Parallel Desktop Window Help

Current Folder: C:\Workshop\bclab-0.91-workshop\userscripts

```
>> pop_saveset (rmfield (exp_eval (io_loadset ('data:/workshop/A01T.set')), 'tracking'), 'filepath', 'A01T_fixed.set')
```

- You can copy & paste this from the .pdf:

```
pop_saveset (rmfield (exp_eval (io_loadset ('data:/  
workshop/A01T.set')), 'tracking'), 'filepath', 'A01T_  
fixed.set')
```


Fixing the Data 😊

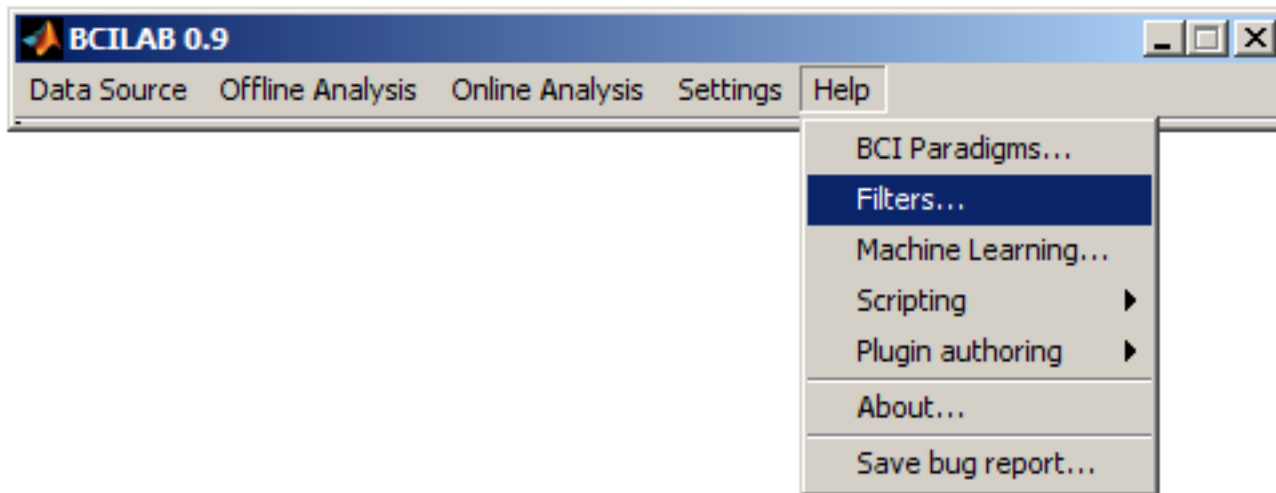
- ... and the same for the three other data sets:

```
pop_saveset(rmfield(exp_eval(io_loadset('data:/workshop/A01E.set')),'tracking'),'filepath','A01E_fixed.set')
```

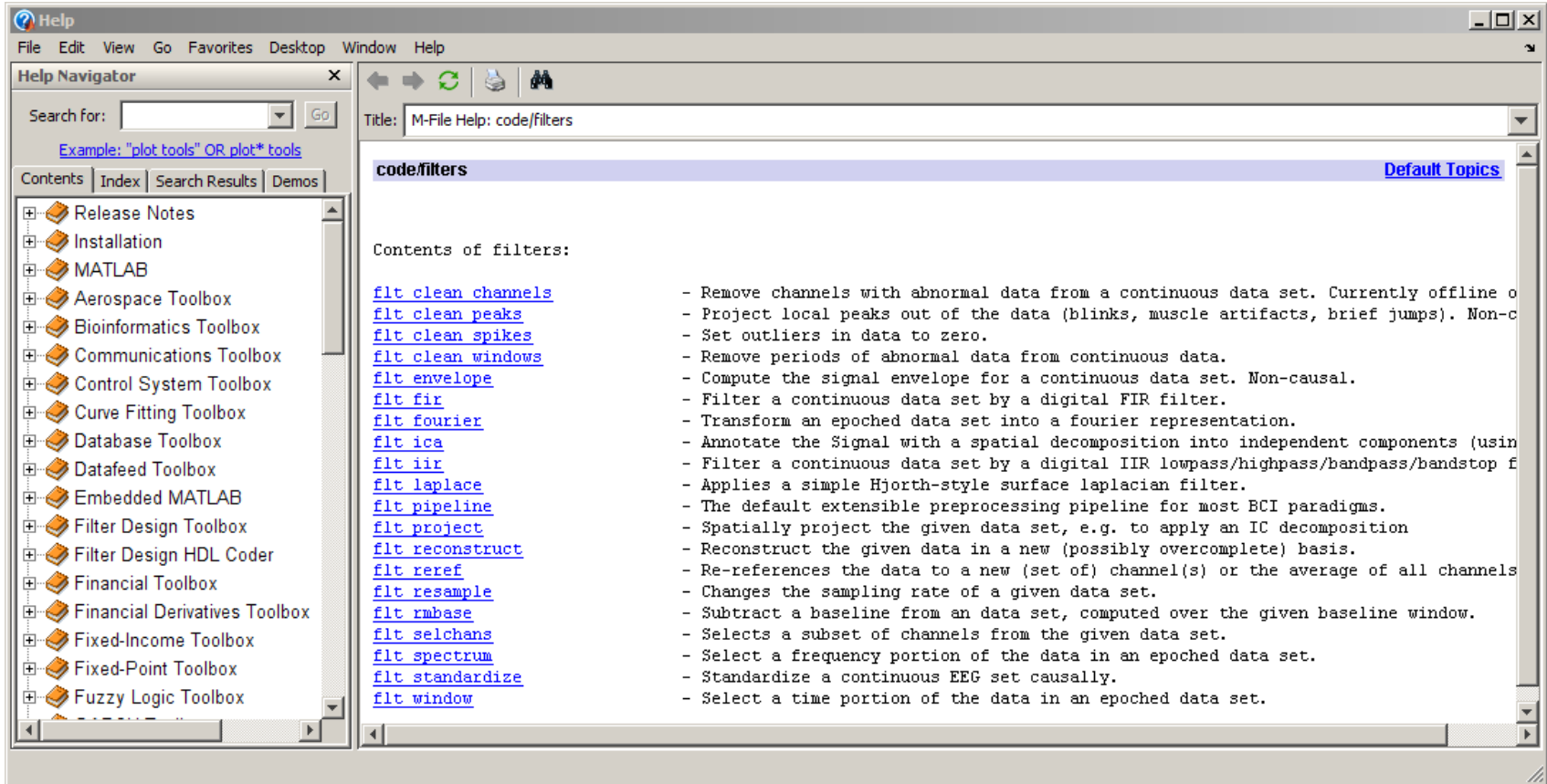
```
pop_saveset(rmfield(exp_eval(io_loadset('data:/workshop/A03T.set')),'tracking'),'filepath','A03T_fixed.set')
```

```
pop_saveset(rmfield(exp_eval(io_loadset('data:/workshop/A03E.set')),'tracking'),'filepath','A03E_fixed.set')
```

Getting help (if needed)



Getting help (if needed)



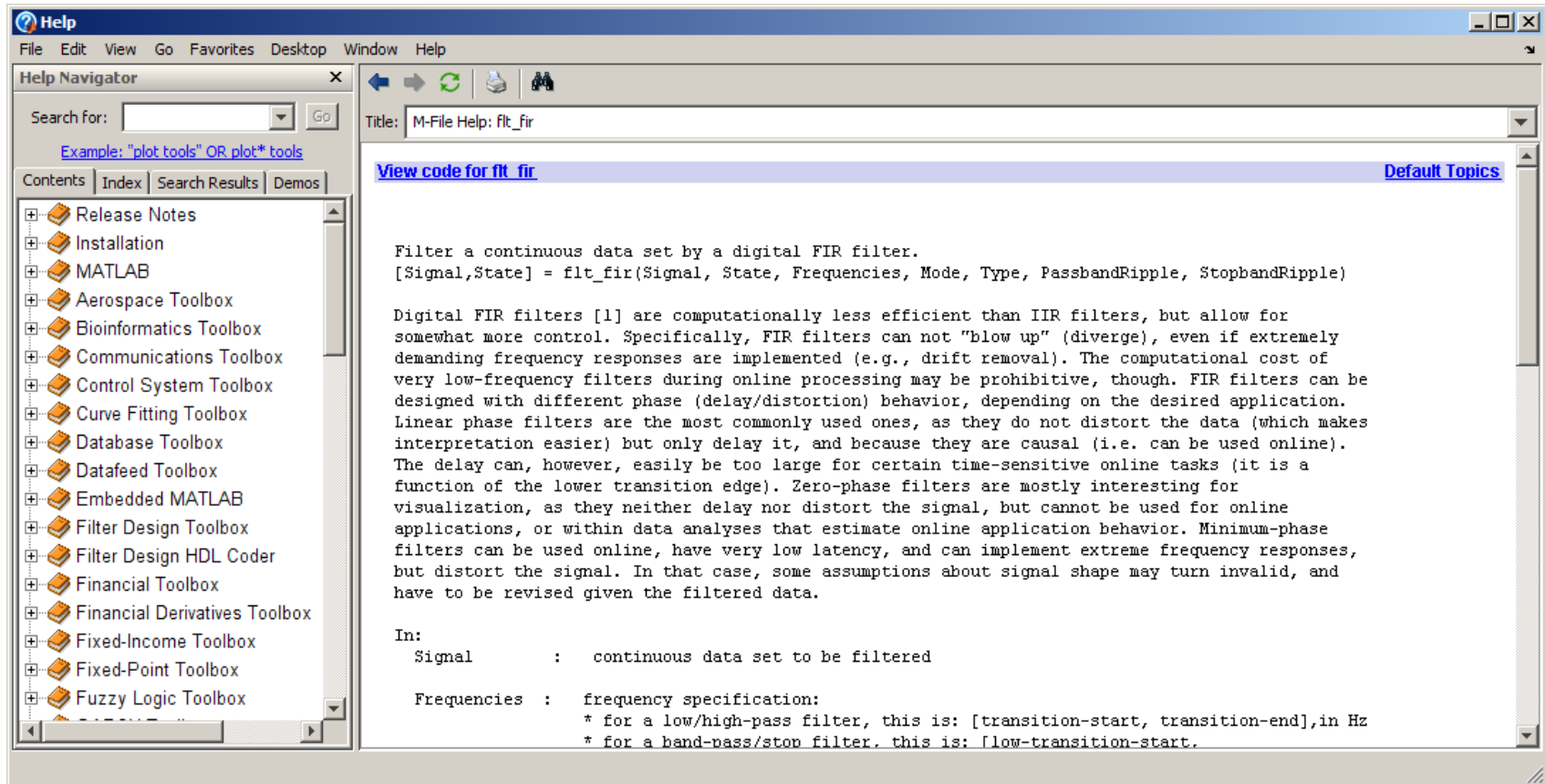
The screenshot shows the MATLAB Help Navigator window. The 'Help Navigator' pane on the left lists various toolboxes, including 'Filter Design Toolbox'. The main window displays the 'code/filters' section, which lists various filter functions and their descriptions.

code/filters [Default Topics](#)

Contents of filters:

- [flt_clean_channels](#) - Remove channels with abnormal data from a continuous data set. Currently offline o
- [flt_clean_peaks](#) - Project local peaks out of the data (blinks, muscle artifacts, brief jumps). Non-c
- [flt_clean_spikes](#) - Set outliers in data to zero.
- [flt_clean_windows](#) - Remove periods of abnormal data from continuous data.
- [flt_envelope](#) - Compute the signal envelope for a continuous data set. Non-causal.
- [flt_fir](#) - Filter a continuous data set by a digital FIR filter.
- [flt_fourier](#) - Transform an epoched data set into a fourier representation.
- [flt_ica](#) - Annotate the Signal with a spatial decomposition into independent components (usin
- [flt_iir](#) - Filter a continuous data set by a digital IIR lowpass/highpass/bandpass/bandstop f
- [flt_laplace](#) - Applies a simple Hjorth-style surface laplacian filter.
- [flt_pipeline](#) - The default extensible preprocessing pipeline for most BCI paradigms.
- [flt_project](#) - Spatially project the given data set, e.g. to apply an IC decomposition
- [flt_reconstruct](#) - Reconstruct the given data in a new (possibly overcomplete) basis.
- [flt_reref](#) - Re-references the data to a new (set of) channel(s) or the average of all channels
- [flt_resample](#) - Changes the sampling rate of a given data set.
- [flt_rmbase](#) - Subtract a baseline from an data set, computed over the given baseline window.
- [flt_selchans](#) - Selects a subset of channels from the given data set.
- [flt_spectrum](#) - Select a frequency portion of the data in an epoched data set.
- [flt_standardize](#) - Standardize a continuous EEG set causally.
- [flt_window](#) - Select a time portion of the data in an epoched data set.

Getting help (if needed)



The screenshot shows the MATLAB Help Navigator window. The 'Help Navigator' pane on the left lists various toolboxes, including 'Filter Design Toolbox'. The main window displays the help page for the 'flt_fir' function, titled 'M-File Help: flt_fir'. The page includes a 'View code for flt_fir' link and a 'Default Topics' link. The main content describes the function's purpose and provides a detailed explanation of FIR filters.

Help Navigator

Search for:

Example: "plot tools" OR plot* tools

Contents | Index | Search Results | Demos

- Release Notes
- Installation
- MATLAB
- Aerospace Toolbox
- Bioinformatics Toolbox
- Communications Toolbox
- Control System Toolbox
- Curve Fitting Toolbox
- Database Toolbox
- Datafeed Toolbox
- Embedded MATLAB
- Filter Design Toolbox
- Filter Design HDL Coder
- Financial Toolbox
- Financial Derivatives Toolbox
- Fixed-Income Toolbox
- Fixed-Point Toolbox
- Fuzzy Logic Toolbox

M-File Help: flt_fir

[View code for flt_fir](#) [Default Topics](#)

Filter a continuous data set by a digital FIR filter.

```
[Signal,State] = flt_fir(Signal, State, Frequencies, Mode, Type, PassbandRipple, StopbandRipple)
```

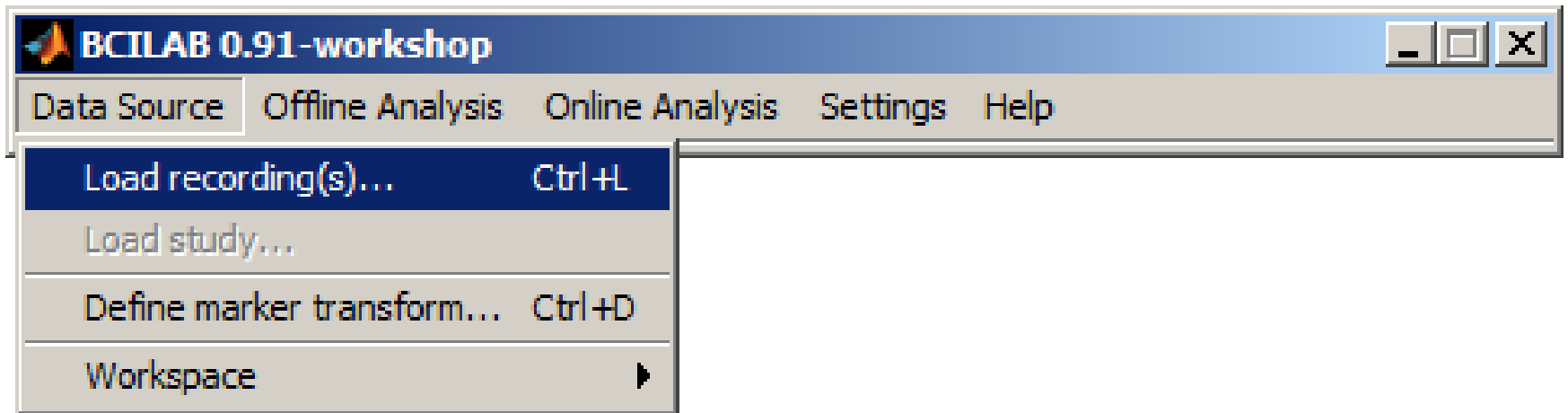
Digital FIR filters [1] are computationally less efficient than IIR filters, but allow for somewhat more control. Specifically, FIR filters can not "blow up" (diverge), even if extremely demanding frequency responses are implemented (e.g., drift removal). The computational cost of very low-frequency filters during online processing may be prohibitive, though. FIR filters can be designed with different phase (delay/distortion) behavior, depending on the desired application. Linear phase filters are the most commonly used ones, as they do not distort the data (which makes interpretation easier) but only delay it, and because they are causal (i.e. can be used online). The delay can, however, easily be too large for certain time-sensitive online tasks (it is a function of the lower transition edge). Zero-phase filters are mostly interesting for visualization, as they neither delay nor distort the signal, but cannot be used for online applications, or within data analyses that estimate online application behavior. Minimum-phase filters can be used online, have very low latency, and can implement extreme frequency responses, but distort the signal. In that case, some assumptions about signal shape may turn invalid, and have to be revised given the filtered data.

In:

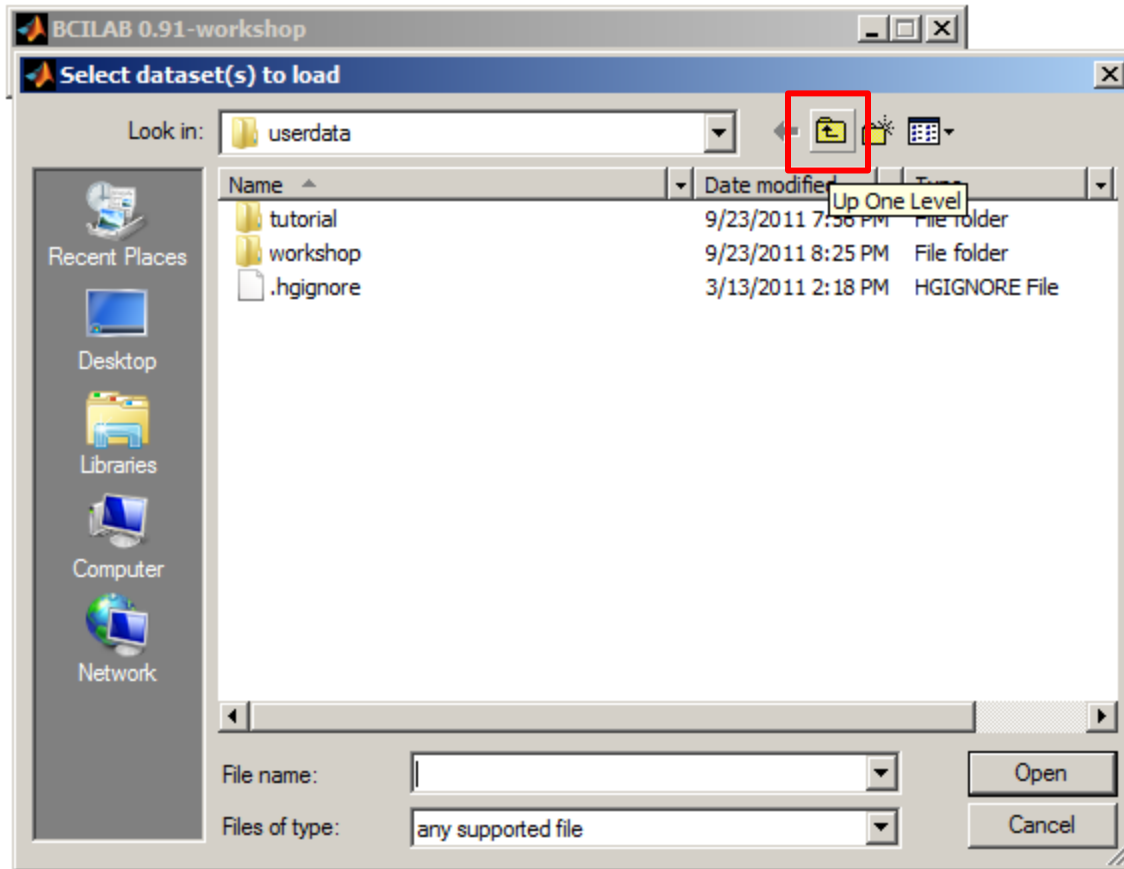
```
Signal      : continuous data set to be filtered

Frequencies : frequency specification:
              * for a low/high-pass filter, this is: [transition-start, transition-end],in Hz
              * for a band-pass/stop filter, this is: [low-transition-start,
```

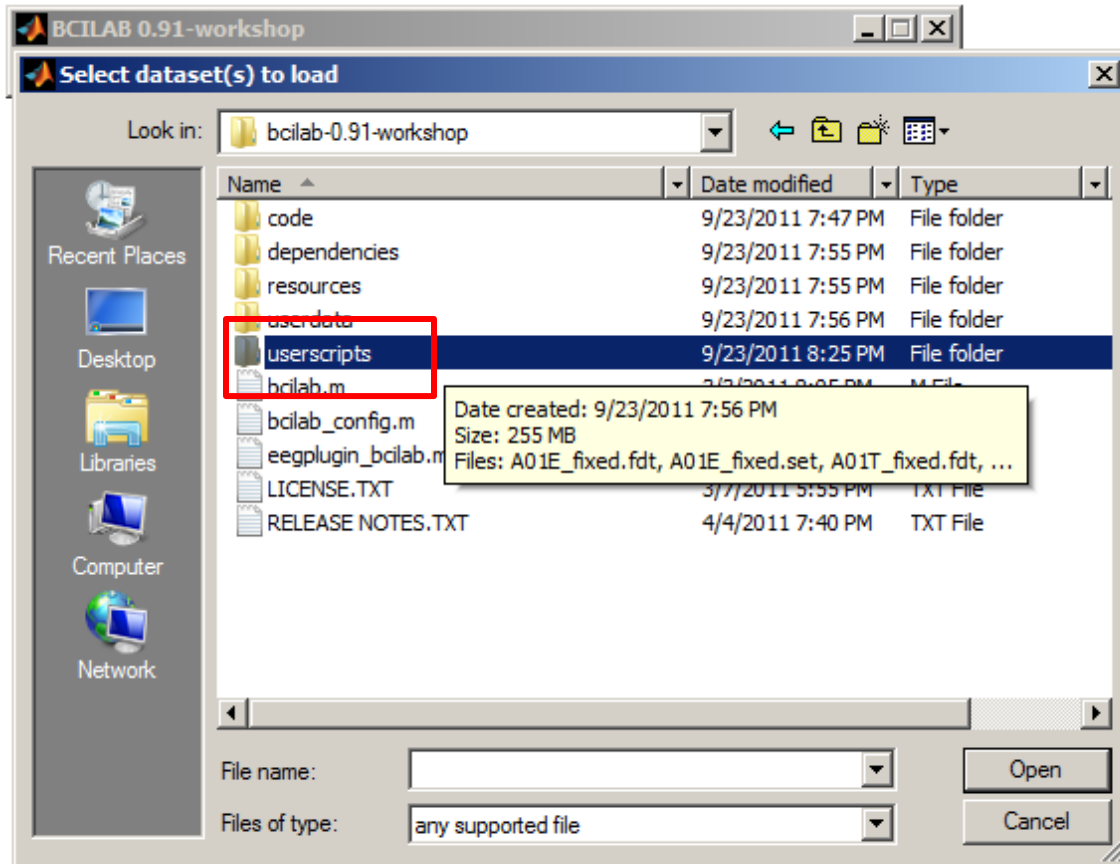
Loading the Data



Loading the Data

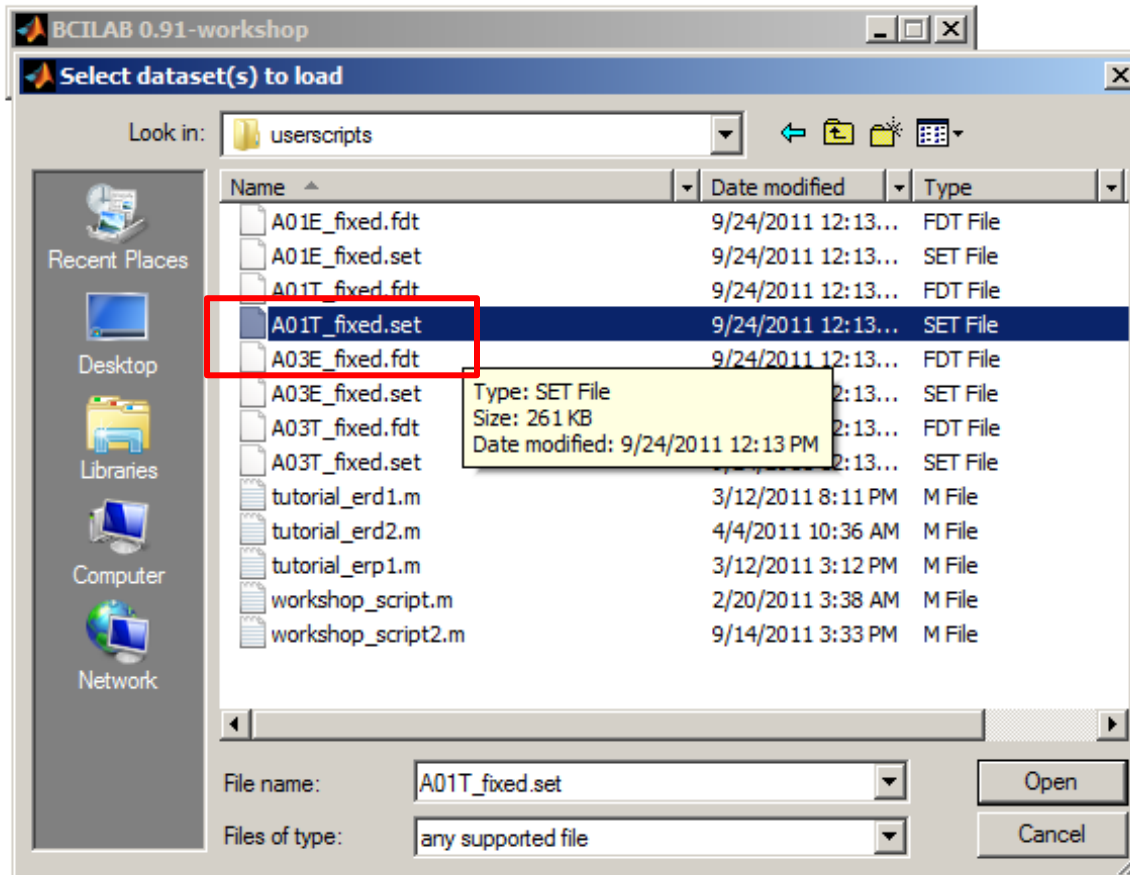


Loading the Data

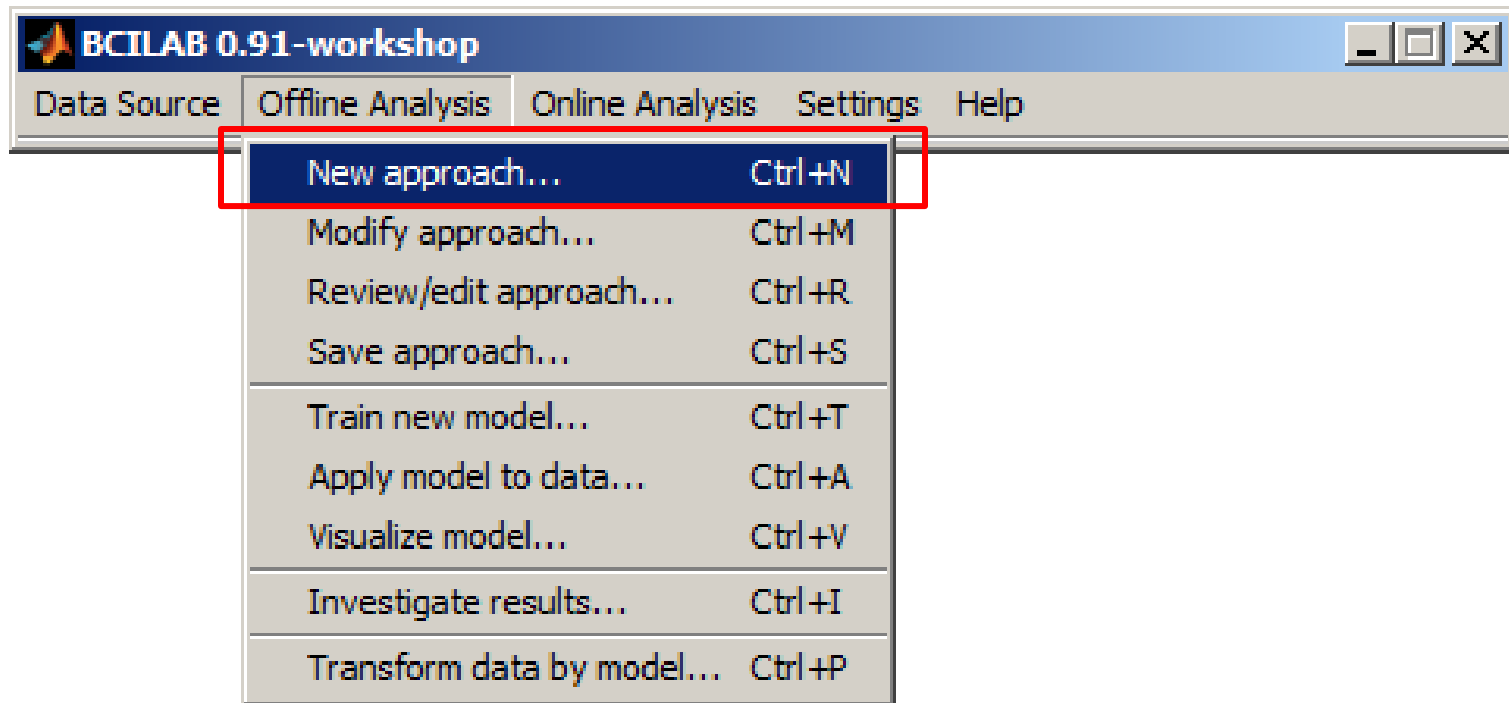


Loading the Data

- Please load A01T_fixed.set (training set #1)

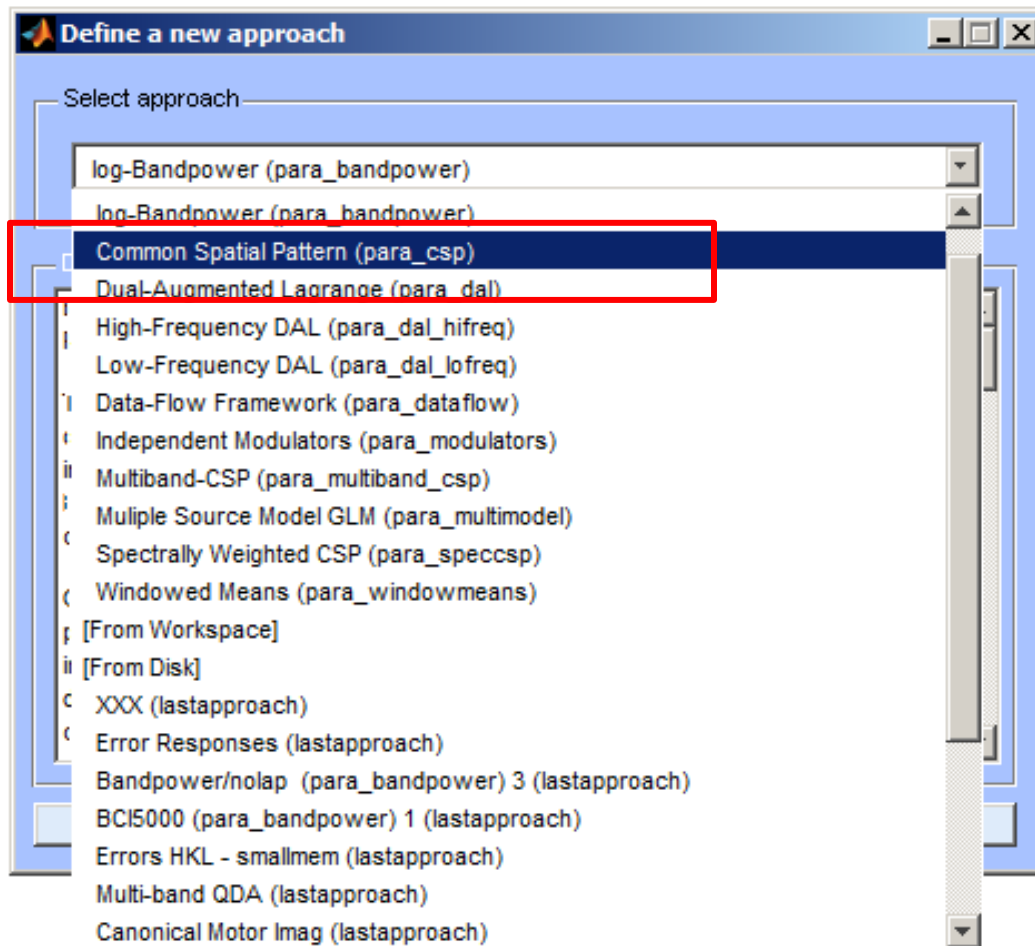


Defining a new Approach

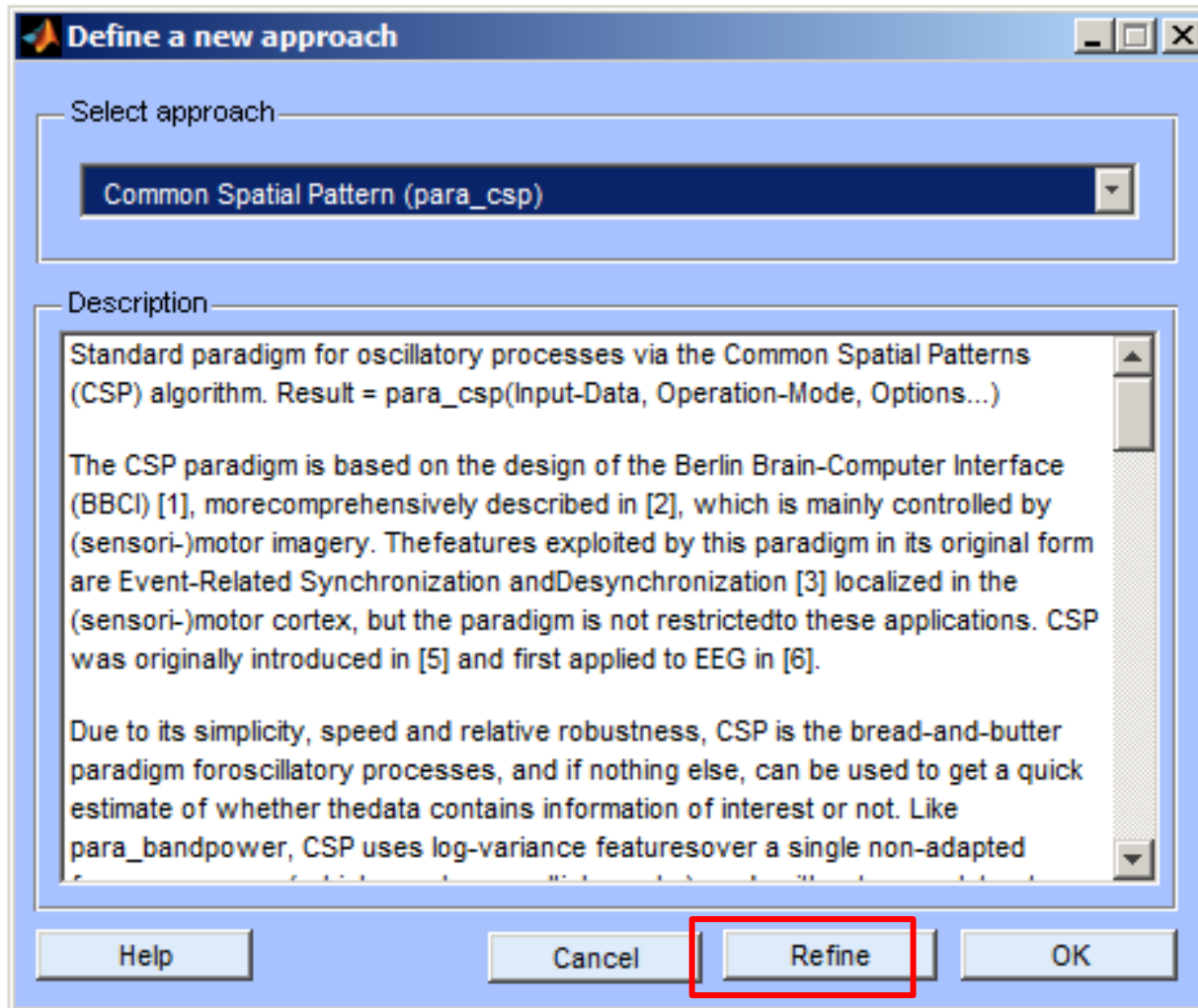


Defining a new Approach

- Let's start with Common Spatial Patterns:

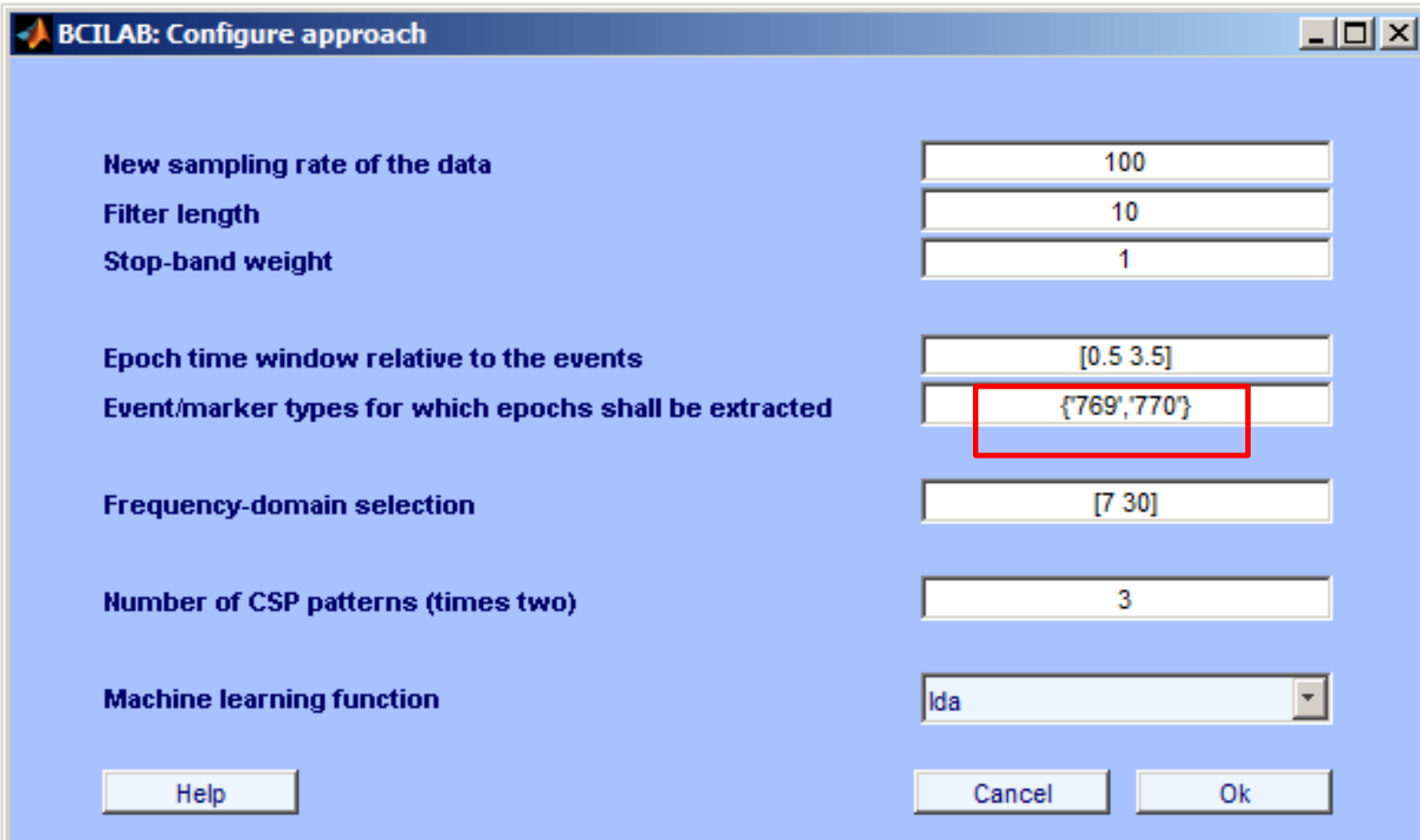


Defining a new Approach



Adapting an Approach

- Marker names are {'769','770'} in this set:



The image shows a screenshot of a software dialog box titled "BCILAB: Configure approach". The dialog box has a blue background and a title bar with standard window controls (minimize, maximize, close). The main area contains several configuration options, each with a corresponding input field:

- New sampling rate of the data:** Input field contains "100".
- Filter length:** Input field contains "10".
- Stop-band weight:** Input field contains "1".
- Epoch time window relative to the events:** Input field contains "[0.5 3.5]".
- Event/marker types for which epochs shall be extracted:** Input field contains {"769','770'"}, which is highlighted with a red rectangular box.
- Frequency-domain selection:** Input field contains "[7 30]".
- Number of CSP patterns (times two):** Input field contains "3".
- Machine learning function:** A dropdown menu showing "lda".

At the bottom of the dialog box, there are three buttons: "Help", "Cancel", and "Ok".

Further Editing an Approach

BCILAB 0.9
Data Source

Review/edit approach

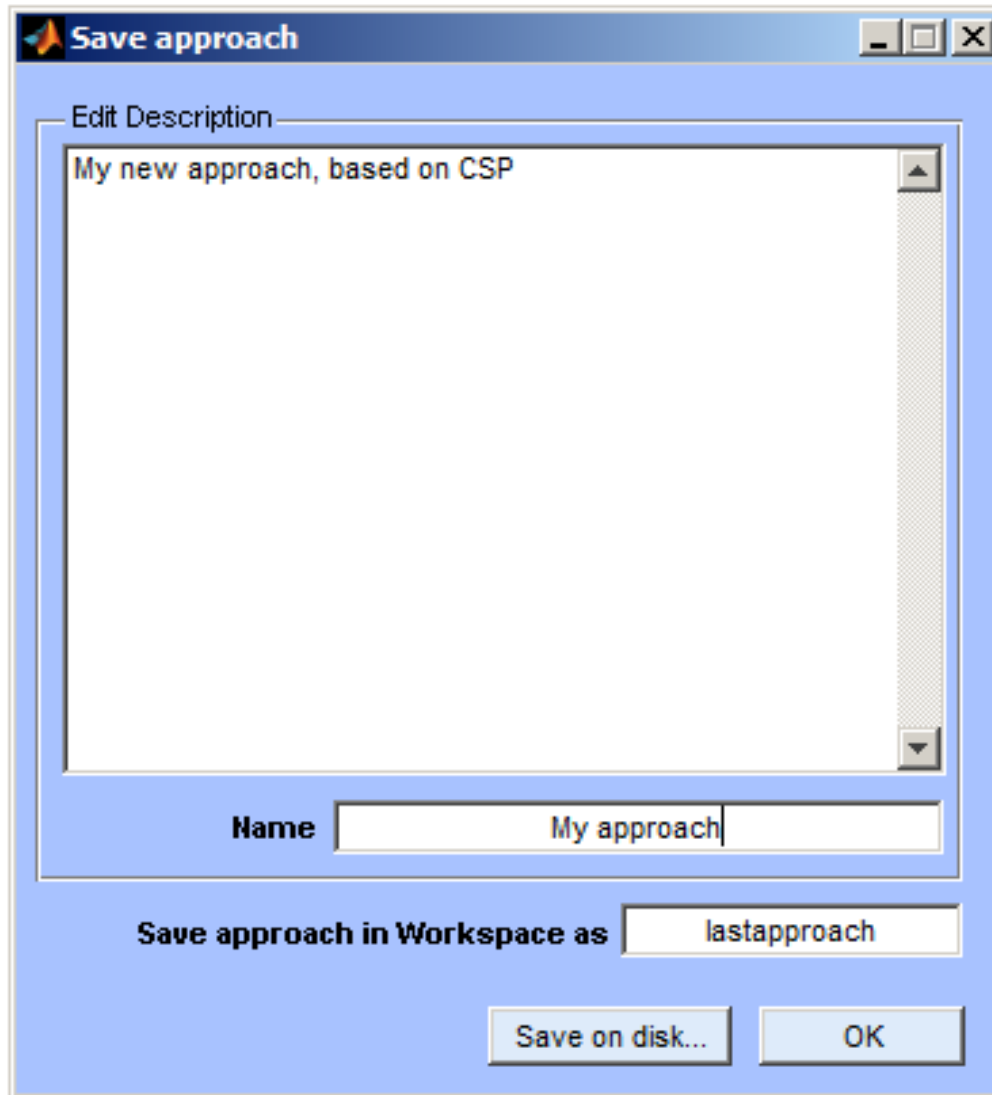
Approach properties

Signal Processing		
SignalProcessing		
FilterOrdering		
Resampling	<input checked="" type="checkbox"/>	
SamplingRate	100	
FilterLength		10
StopbandWeight		1
ChannelSelection	<input type="checkbox"/>	
Rereferencing	<input type="checkbox"/>	
SurfaceLaplacian	<input type="checkbox"/>	
ICA	<input type="checkbox"/>	
DipoleFitting	<input type="checkbox"/>	
FIRFilter	<input type="checkbox"/>	
Projection	<input type="checkbox"/>	
VolumeSelection	<input type="checkbox"/>	
IIRFilter	<input type="checkbox"/>	
Standardization	<input type="checkbox"/>	
SparseReconstruction	<input type="checkbox"/>	
EpochExtraction	<input checked="" type="checkbox"/>	
TimeWindow	[0.5 3.5]	
EventTypes	769 770	
BaselineRemoval	<input type="checkbox"/>	
WindowSelection	<input type="checkbox"/>	
EpochedFFT	<input type="checkbox"/>	
SpectralTransform	<input type="checkbox"/>	
SpectralSelection	<input checked="" type="checkbox"/>	
FrequencySpecification	[7 30]	
WaveletTransform	<input type="checkbox"/>	
Feature Extraction		
FeatureExtraction		

(Name)
(Description)

Help Cancel OK

Saving the Approach



The image shows a software dialog box titled "Save approach". It features a blue header bar with the title and standard window controls (minimize, maximize, close). Below the header is a text area labeled "Edit Description" containing the text "My new approach, based on CSP". At the bottom of the dialog, there are two text input fields: one labeled "Name" with the value "My approach", and another labeled "Save approach in Workspace as" with the value "lastapproach". At the very bottom, there are two buttons: "Save on disk..." and "OK".

Save approach

Edit Description

My new approach, based on CSP

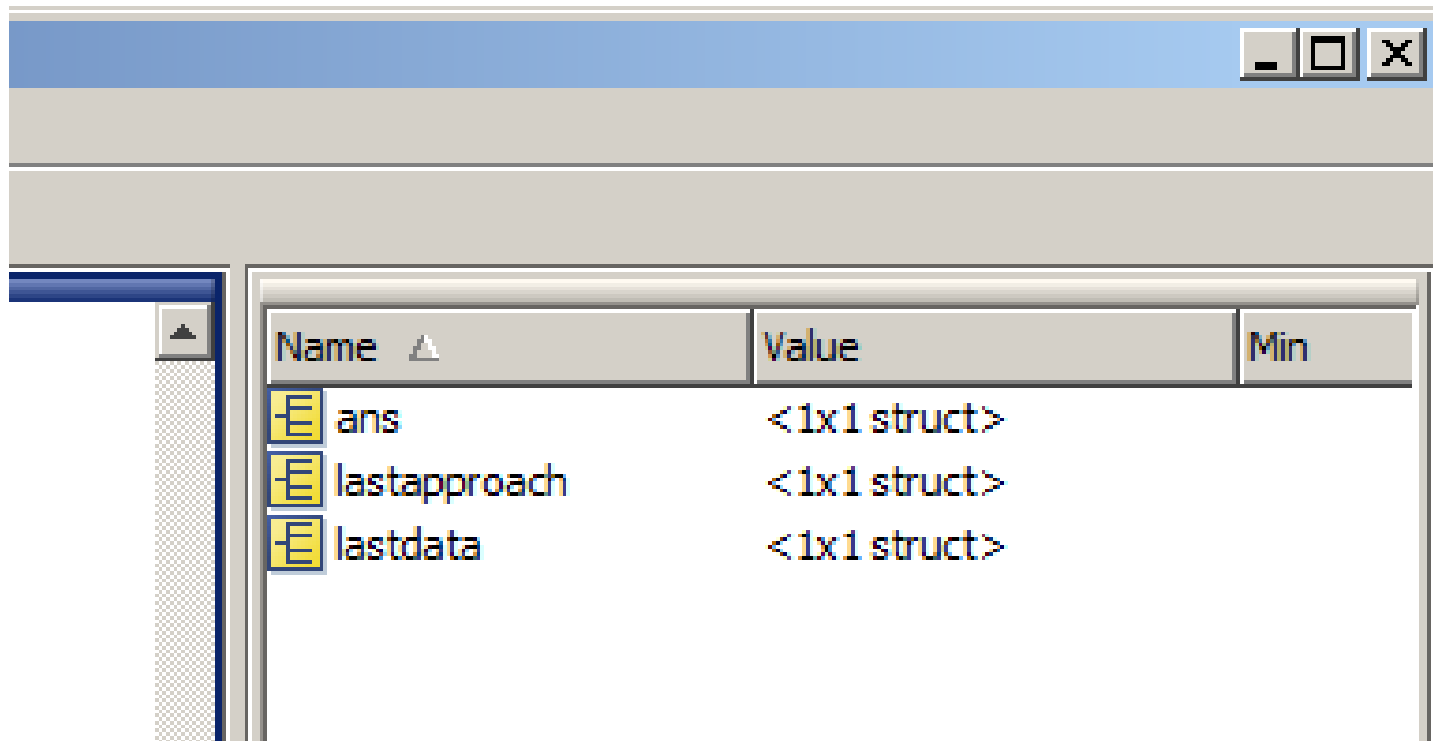
Name My approach

Save approach in Workspace as lastapproach

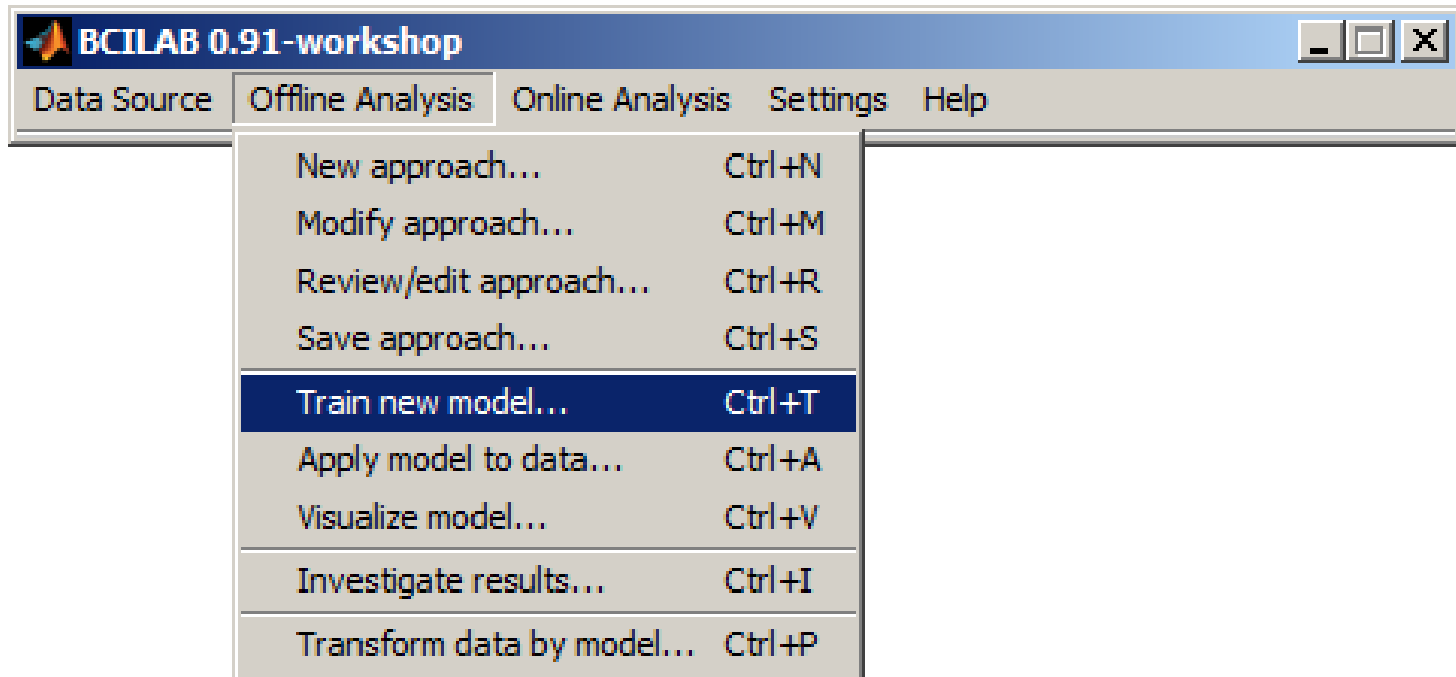
Save on disk... OK

See also: MATLAB's Workspace

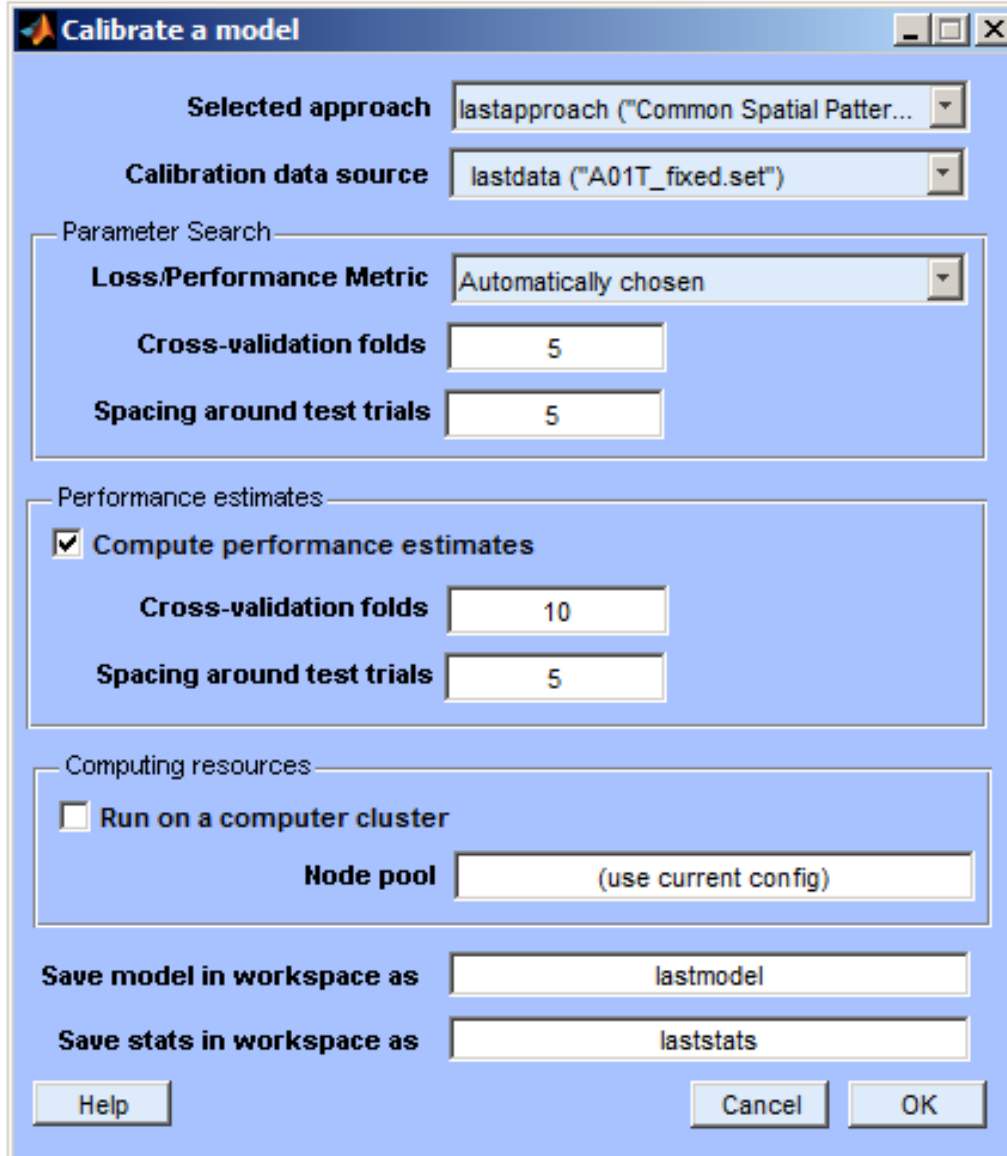
- Can be enabled via “Desktop” menu



Now Hitting the Data...



Now Hitting the Data...



Calibrate a model

Selected approach lastapproach ("Common Spatial Patter...")

Calibration data source lastdata ("A01T_fixed.set")

Parameter Search

Loss/Performance Metric Automatically chosen

Cross-validation folds 5

Spacing around test trials 5

Performance estimates

Compute performance estimates

Cross-validation folds 10

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

Watching the Computation...

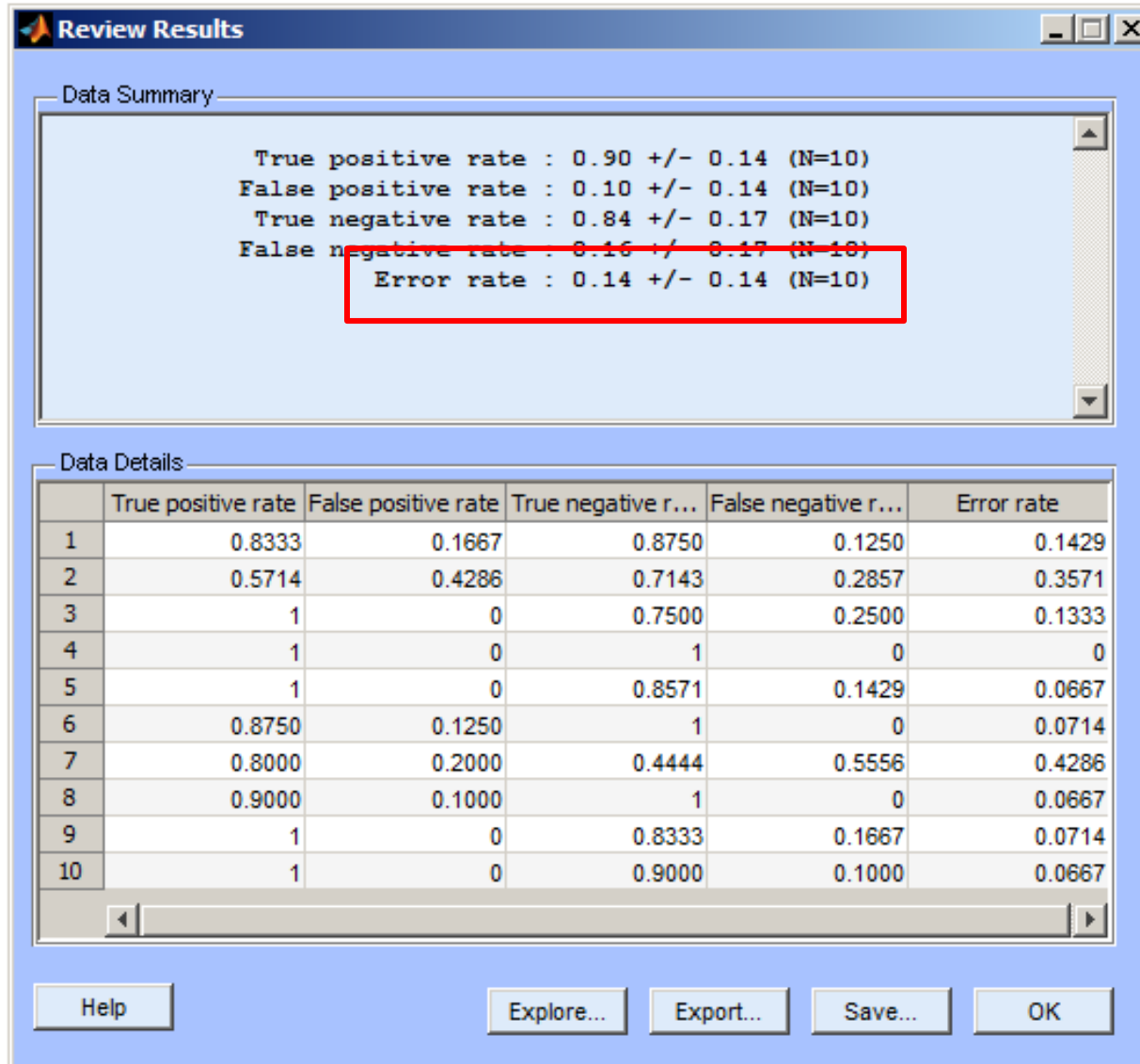
```
loading C:\Workshop\bcilab-0.91-workshop\userscripts\A01T_fi:
pop_loadset(): loading file C:\Workshop\bcilab-0.91-workshop\
Reading float file 'C:\Workshop\bcilab-0.91-workshop\userscr:
The loaded EEGLAB set is lacking an online expression; assum:
If it contains filtered data, however, BCI models derived fro
pop_epoch():144 epochs selected
Epoching...
pop_epoch():144 epochs generated
pop_epoch(): checking epochs for data discontinuity
```

fx

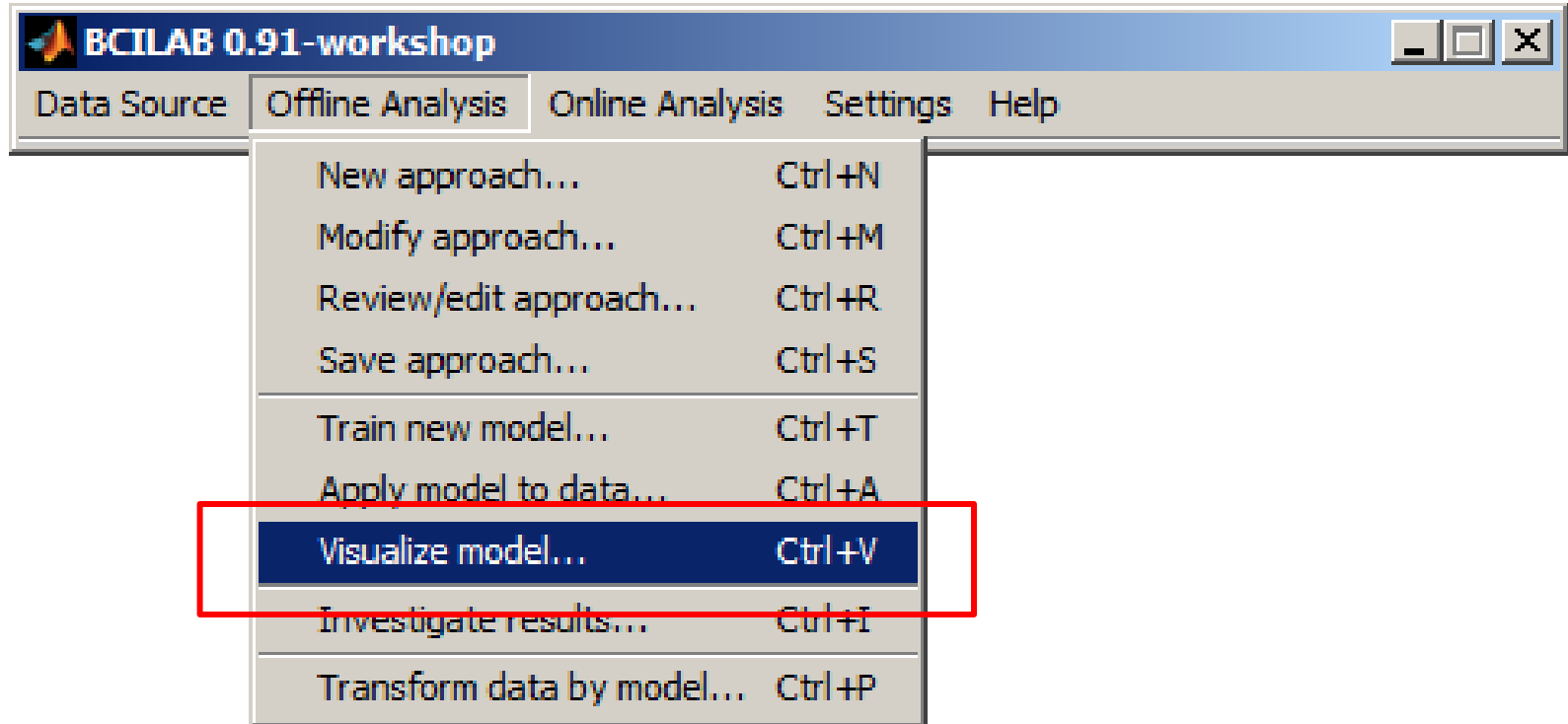
>>

 Start

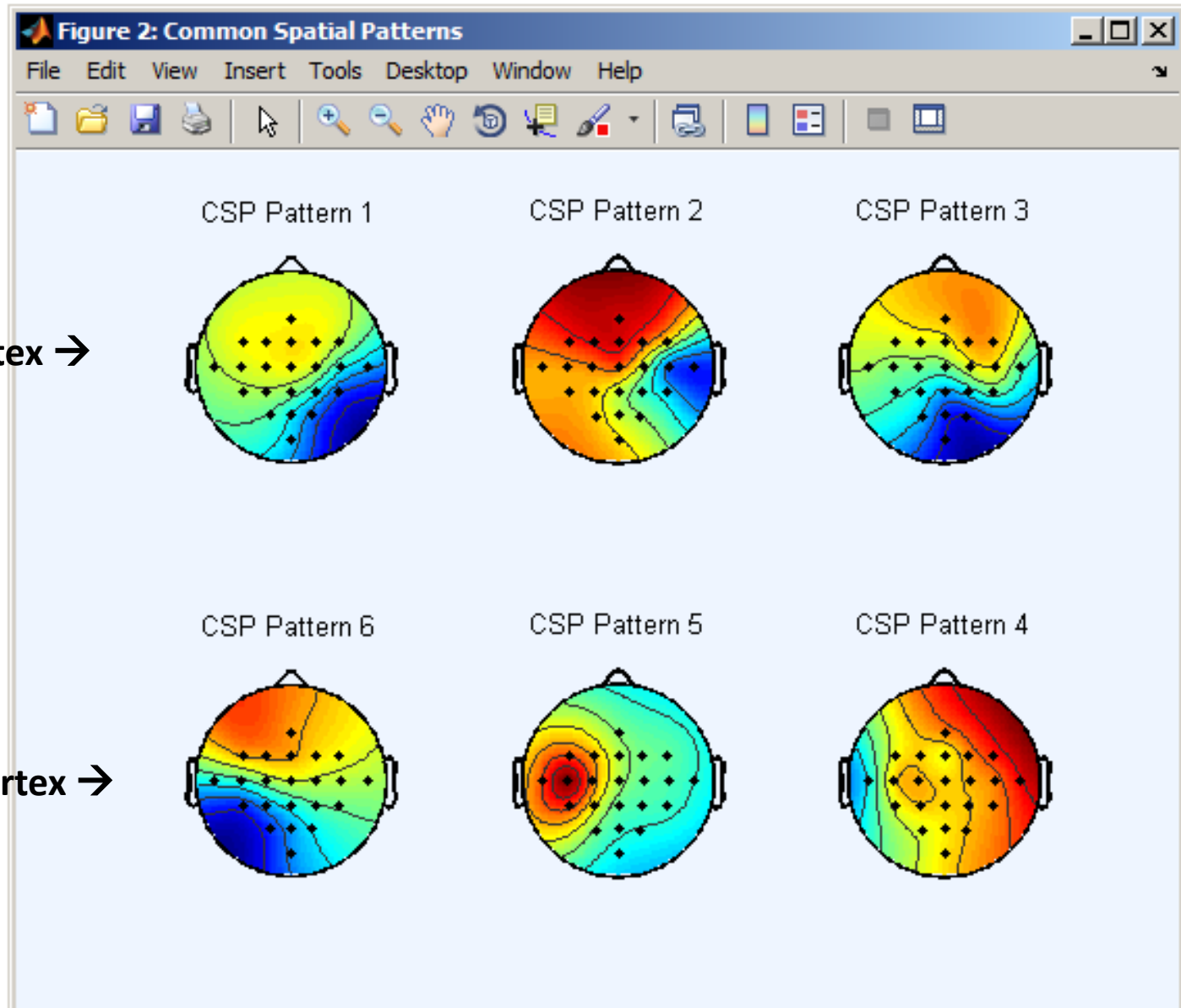
Cross-validation Results:



Visualizing a Trained Model



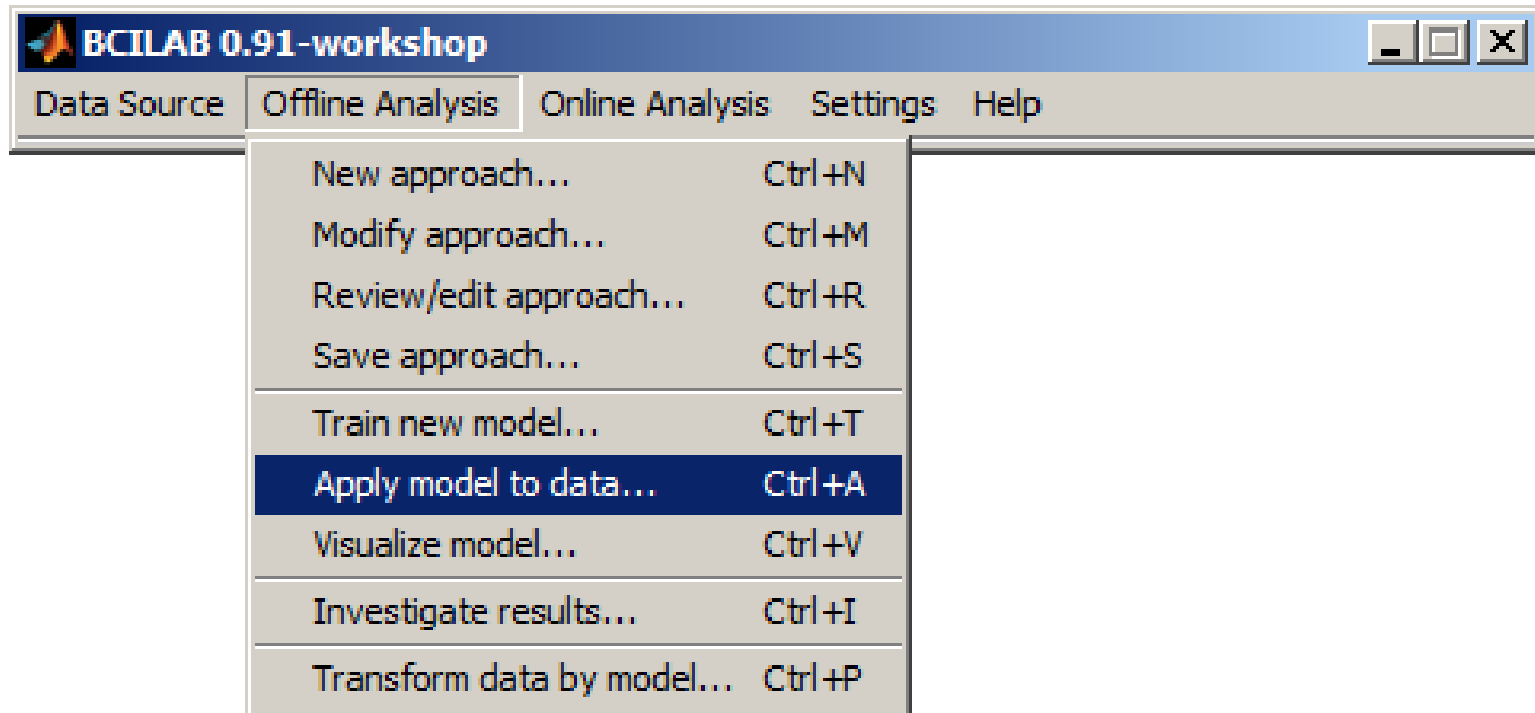
Visualizing a Trained Model



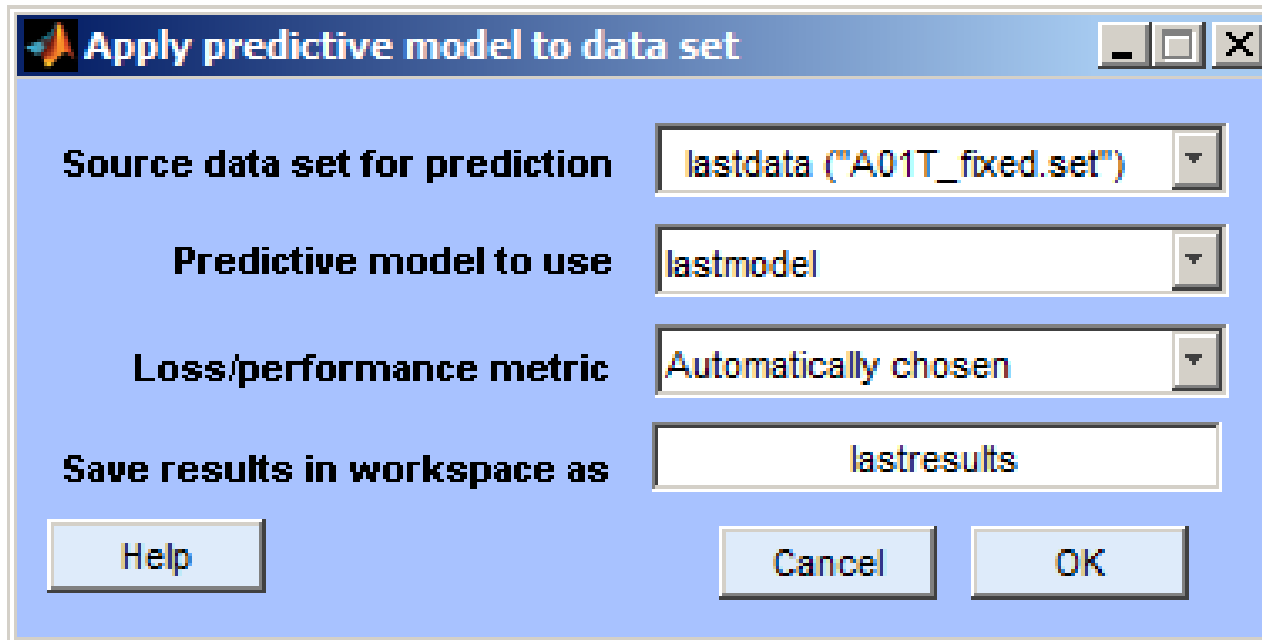
Left hand motor cortex →

Right hand motor cortex →

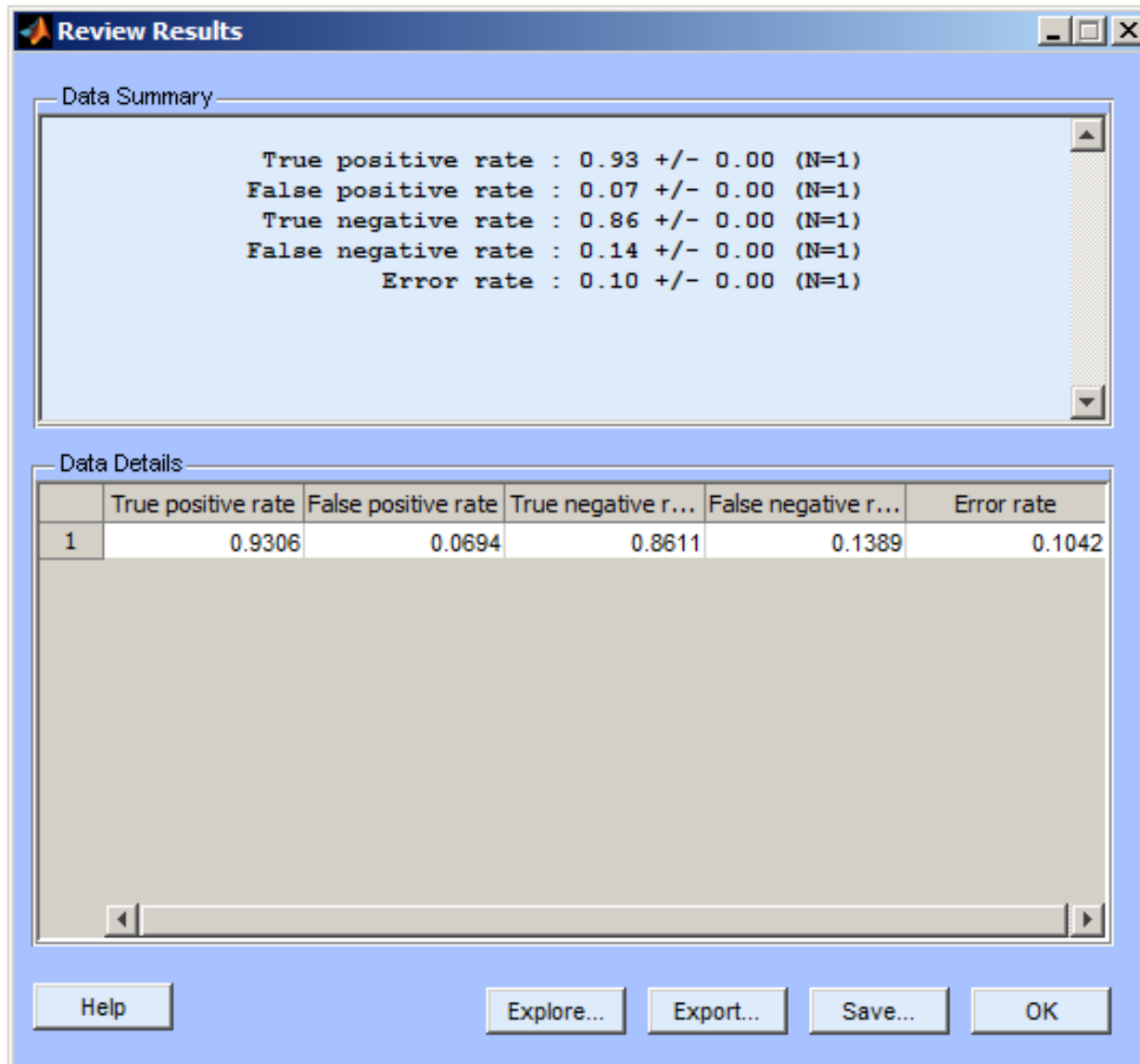
Applying a Model to New Data



Applying a Model to New Data

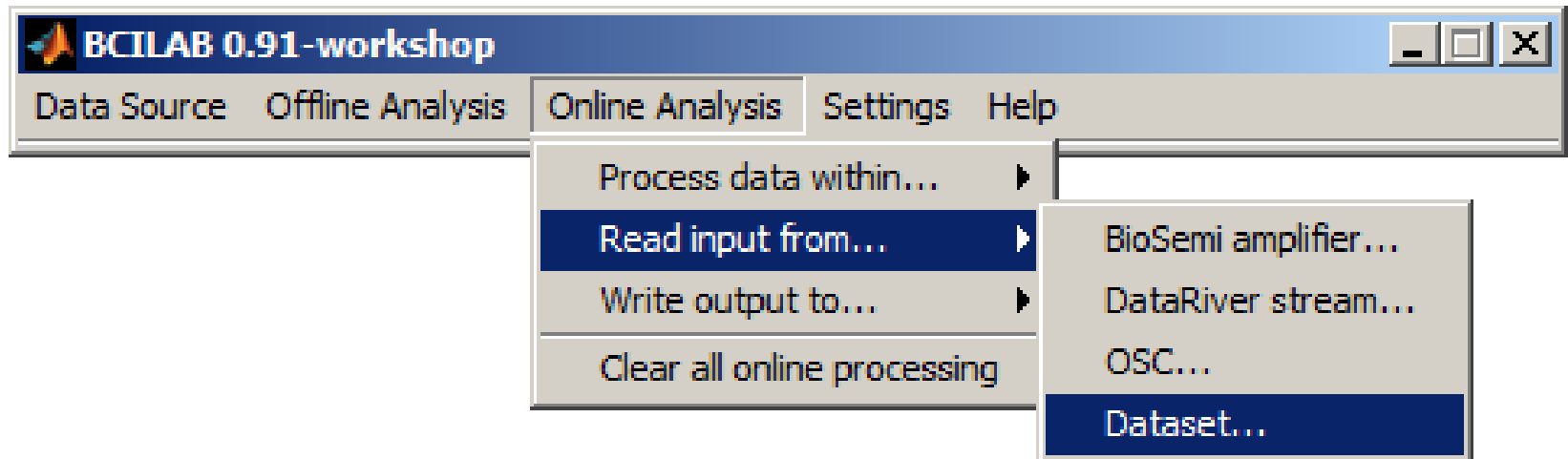


Results: Training-Set Error



Online Processing

- For lack of EEG hardware, play back a data set in real time:



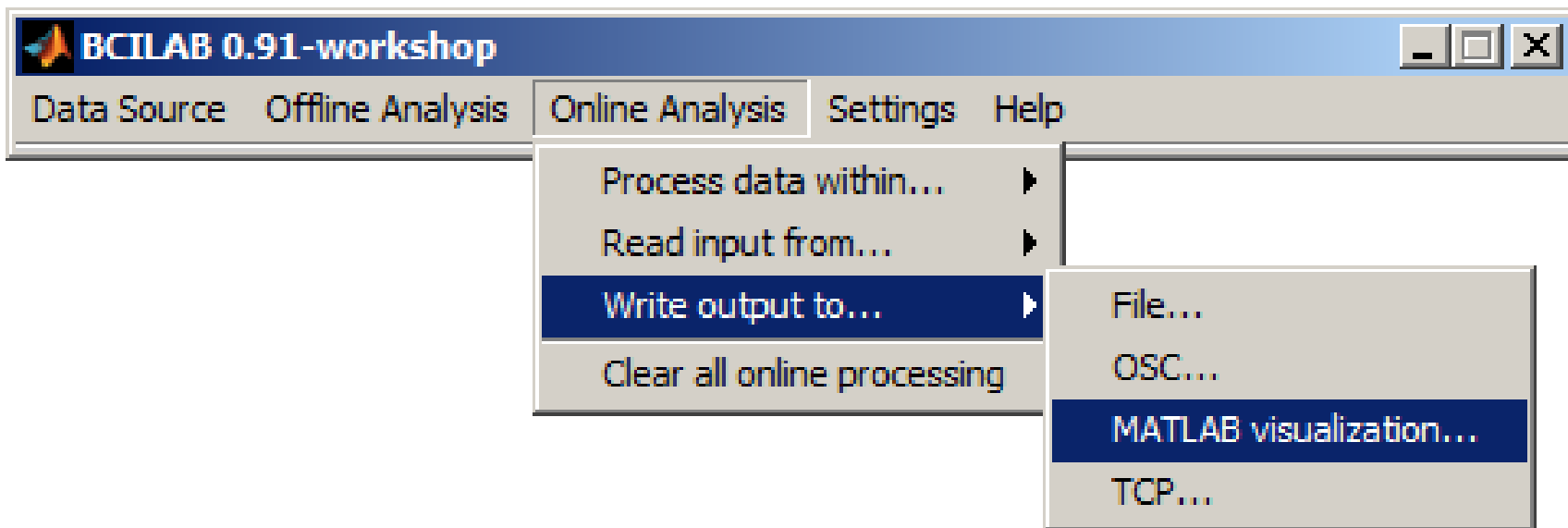
Online Processing

- For lack of EEG hardware, play back a data set in real time:

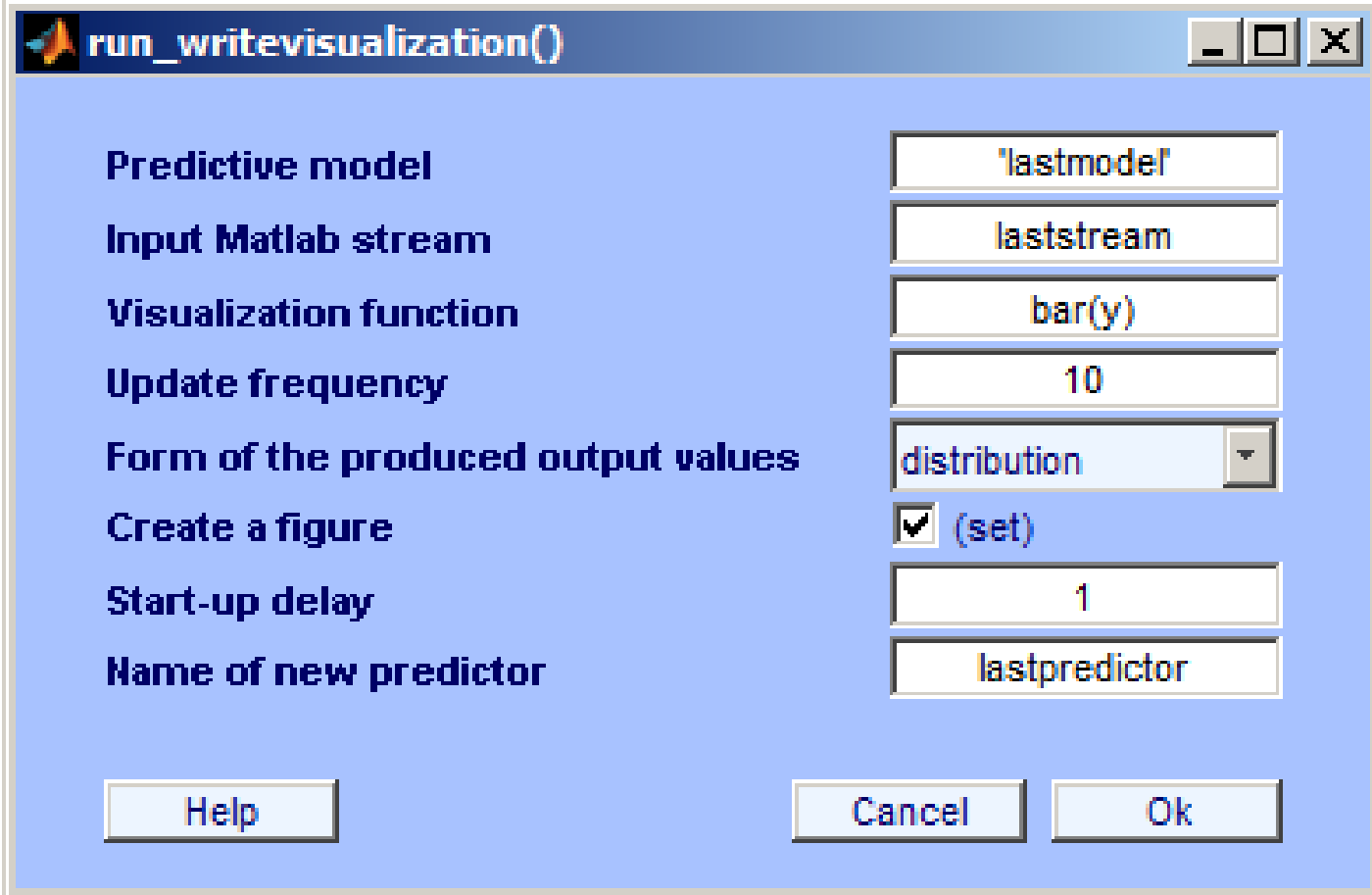


Online Processing

- Choosing a destination for outputs:



Setting up a Visualization

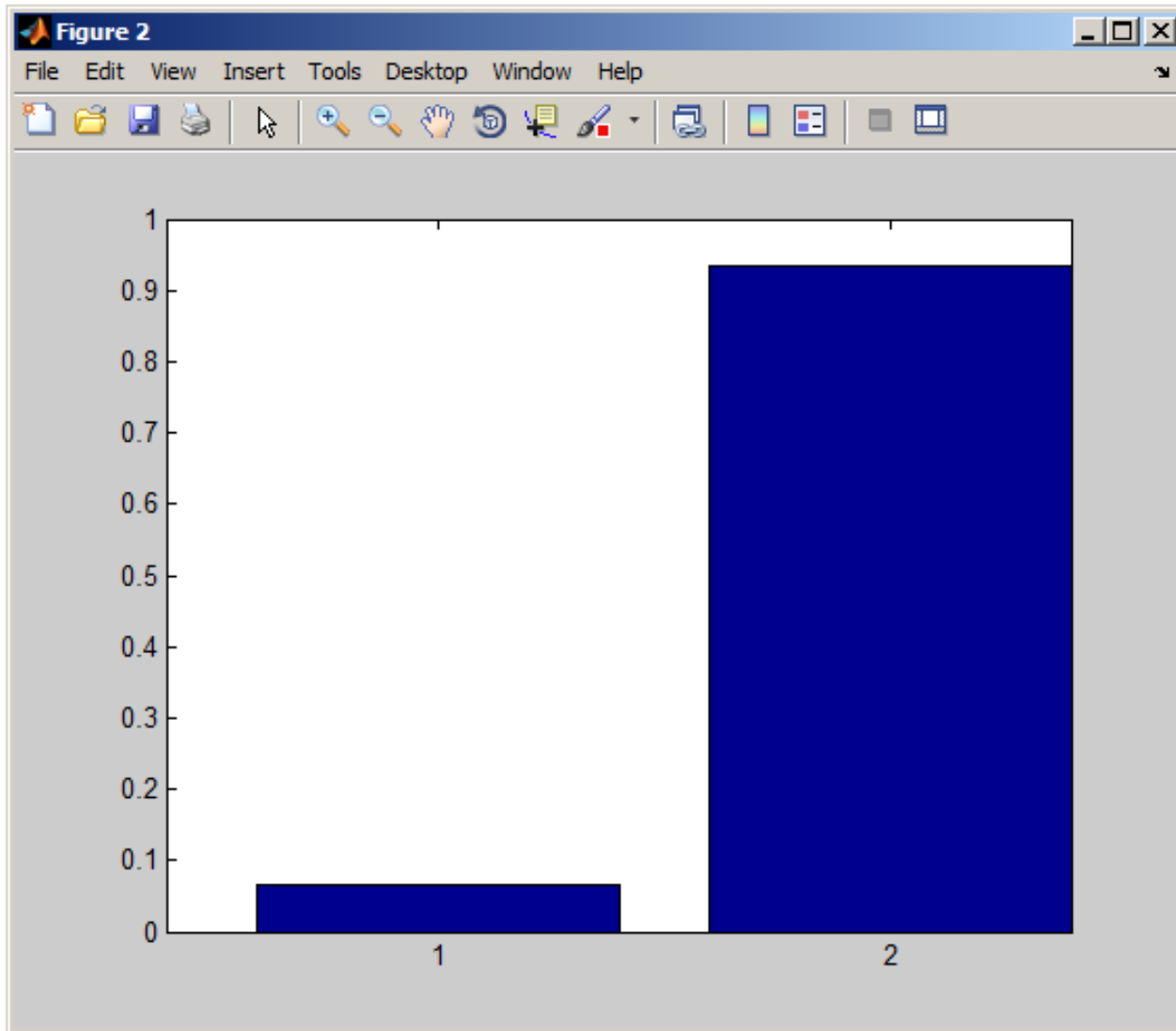


The image shows a MATLAB dialog box titled "run_writevisualization()". The dialog has a light blue background and a title bar with standard window controls. It contains several configuration options for a visualization process. The options are listed on the left, and their corresponding values are shown in input fields on the right. At the bottom, there are three buttons: "Help", "Cancel", and "Ok".

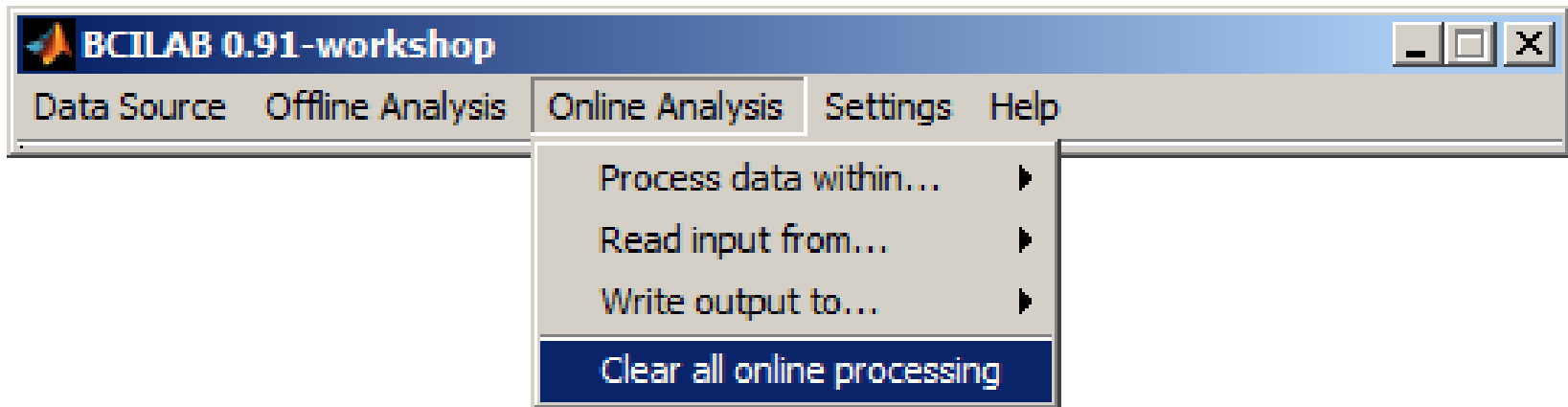
Option	Value
Predictive model	'lastmodel'
Input Matlab stream	laststream
Visualization function	bar(y)
Update frequency	10
Form of the produced output values	distribution
Create a figure	<input checked="" type="checkbox"/> (set)
Start-up delay	1
Name of new predictor	lastpredictor

Buttons: Help, Cancel, Ok

Real-time Class Probabilities



Clearing the Online Processes

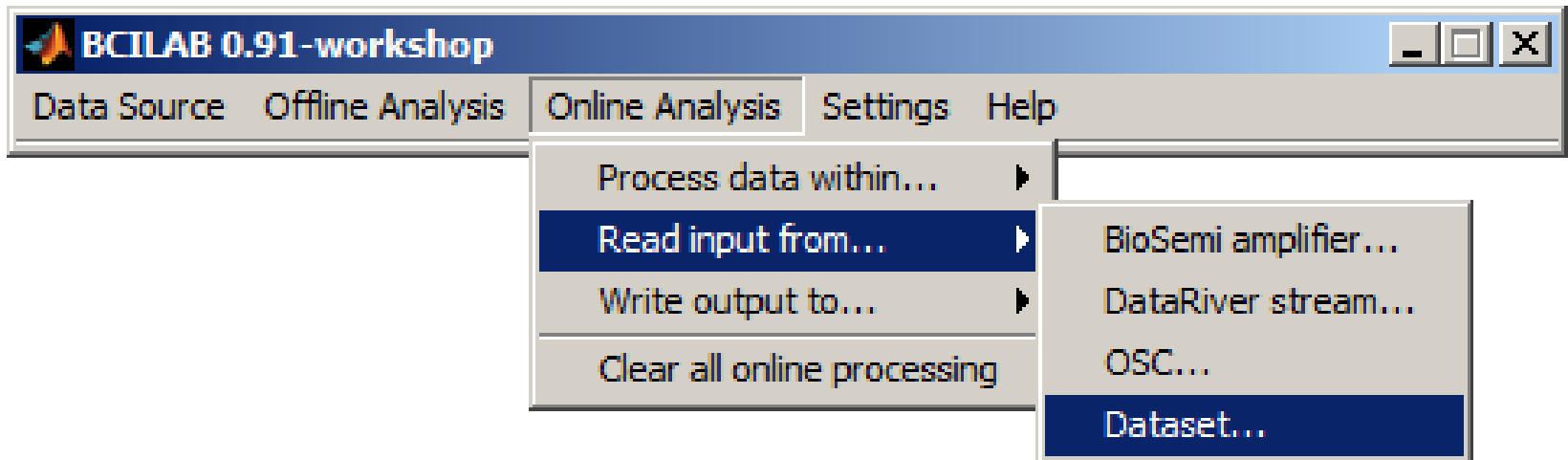




Fancier Visualization

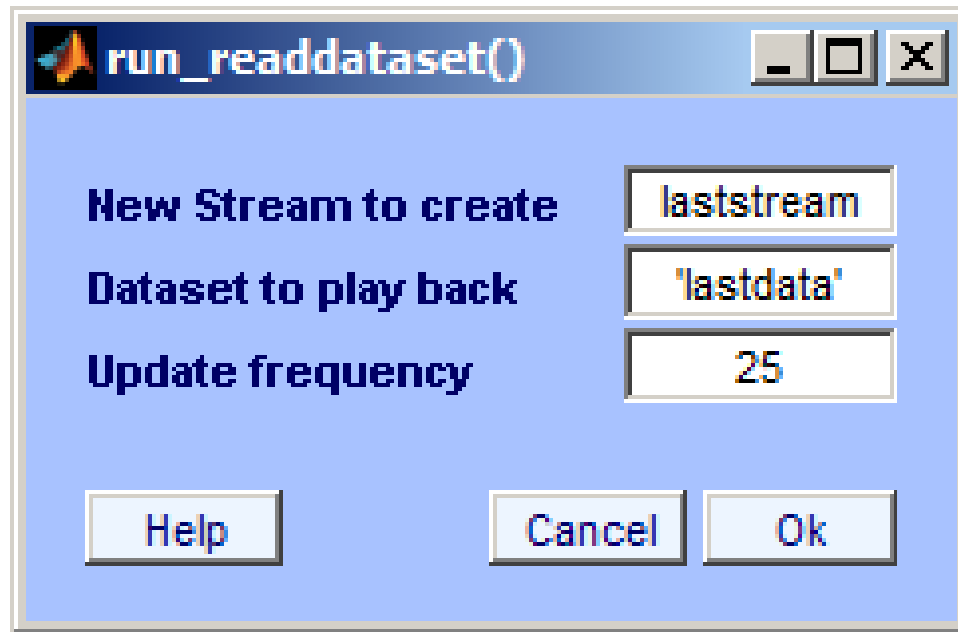
Online Processing

- For lack of EEG hardware, play back a data set in real time:



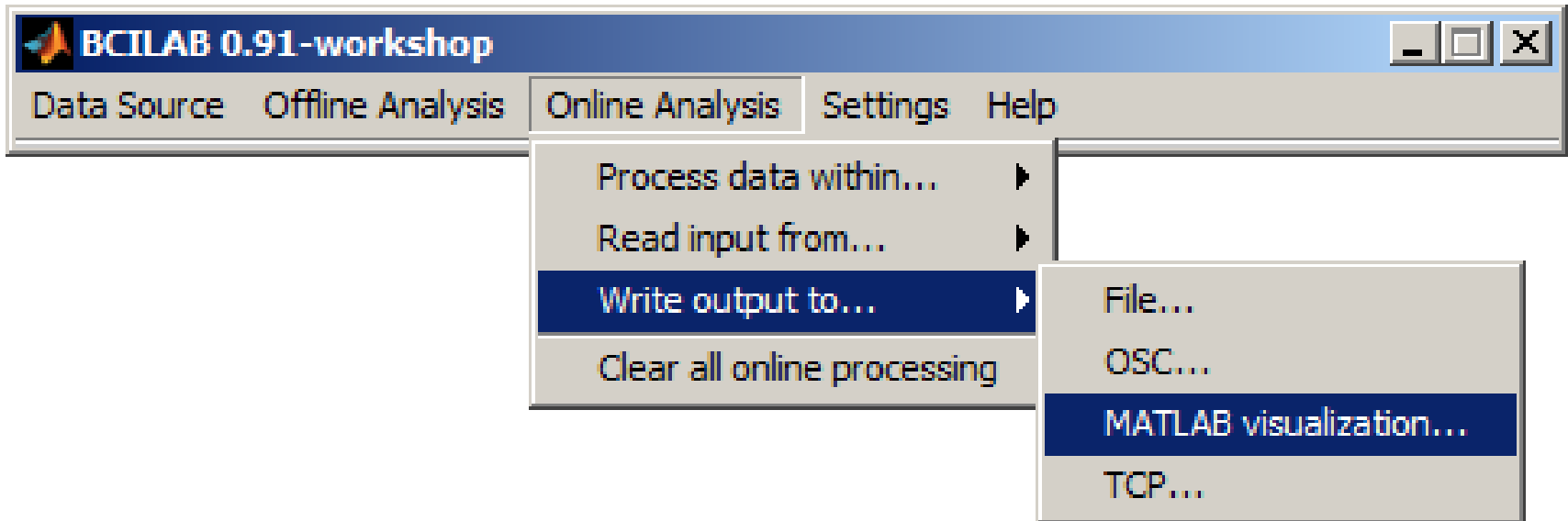
Online Processing

- For lack of EEG hardware, play back a data set in real time:



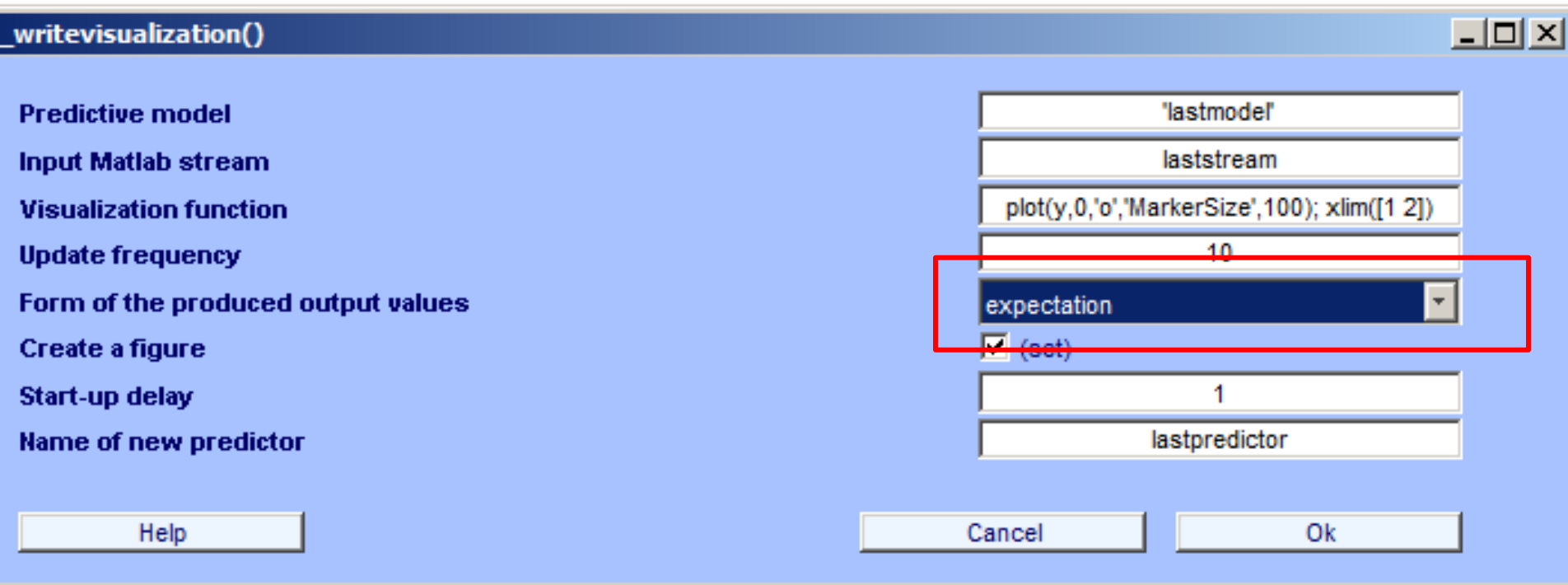
Online Processing

- Choosing a destination for outputs:



Setting up a Visualization

- The command is:
`plot(y,0,'o','MarkerSize',100); xlim([1 2])`



writevisualization()

Predictive model

Input Matlab stream

Visualization function

Update frequency

Form of the produced output values

Create a figure

Start-up delay

Name of new predictor

Help

Cancel

Ok

'lastmodel'

laststream

plot(y,0,'o','MarkerSize',100); xlim([1 2])

10

expectation

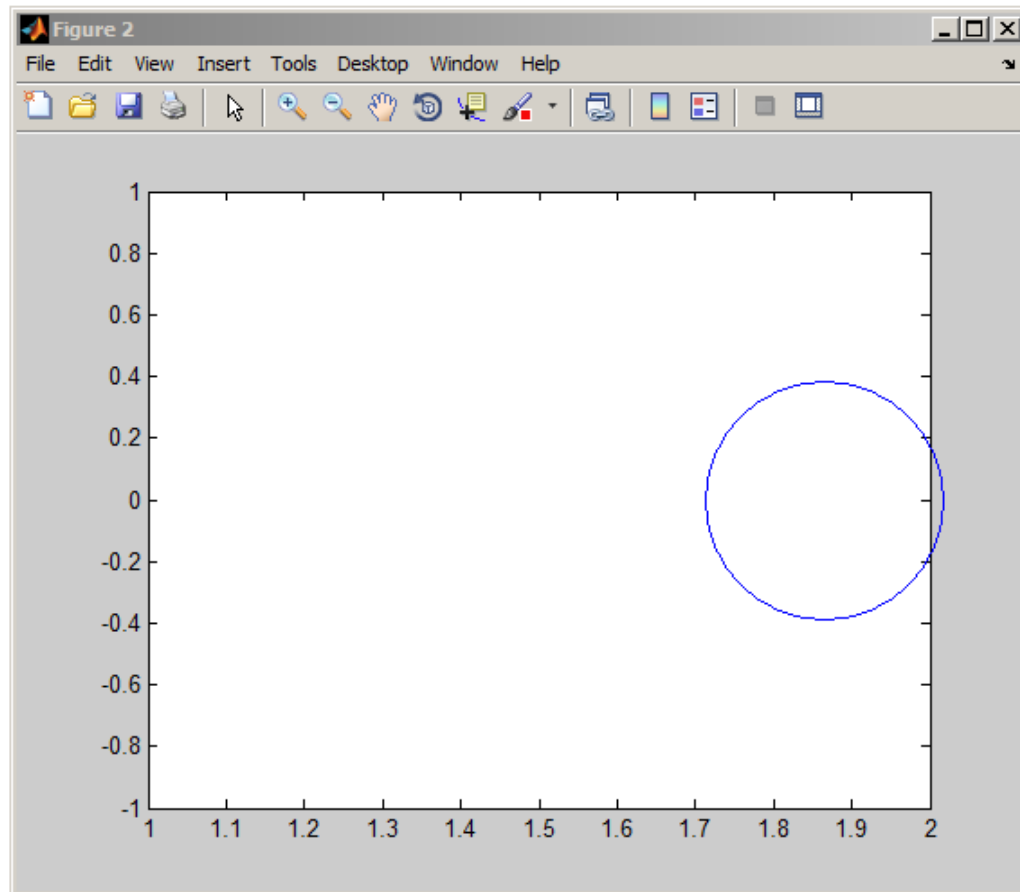
(set)

1

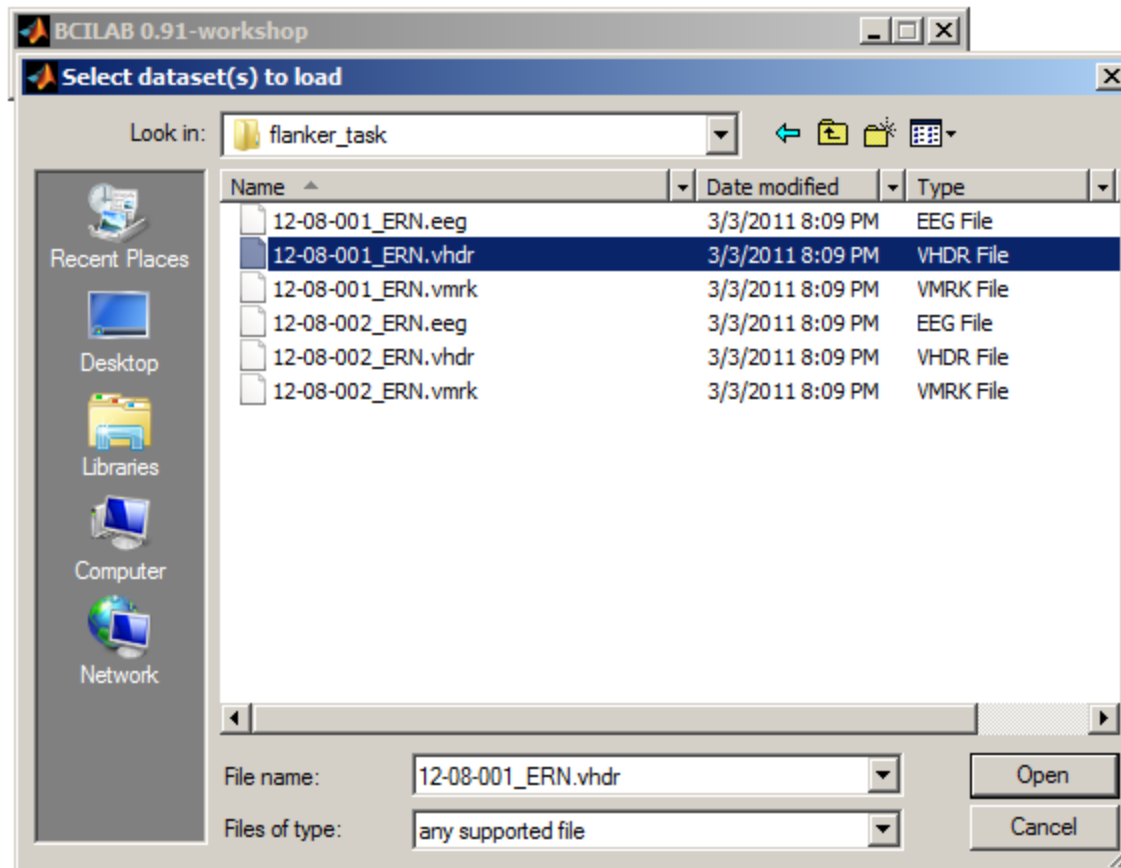
lastpredictor

Real-time Class Probabilities

- The sphere indicates the imagined movement side

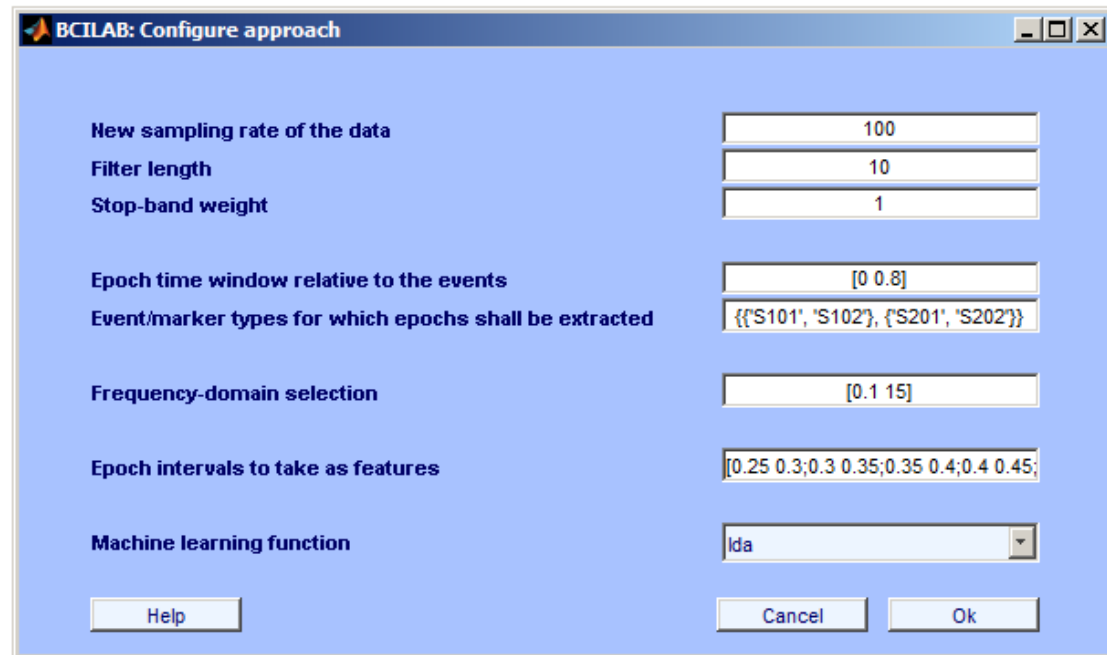


Looking at ERP Data



Setting up

- Markers: `{{'S101','S102'},{'S201','S202'}}`
- Time Windows: `[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]`
- Freq flt: `[0.1 15]`
- Epoch: `[0 0.8]`



BCILAB: Configure approach

New sampling rate of the data	<input type="text" value="100"/>
Filter length	<input type="text" value="10"/>
Stop-band weight	<input type="text" value="1"/>
Epoch time window relative to the events	<input type="text" value="[0 0.8]"/>
Event/marker types for which epochs shall be extracted	<input type="text" value="{{'S101','S102'},{'S201','S202'}}"/>
Frequency-domain selection	<input type="text" value="[0.1 15]"/>
Epoch intervals to take as features	<input type="text" value="[0.25 0.3;0.3 0.35;0.35 0.4;0.4 0.45;"/>
Machine learning function	<input type="text" value="lda"/>