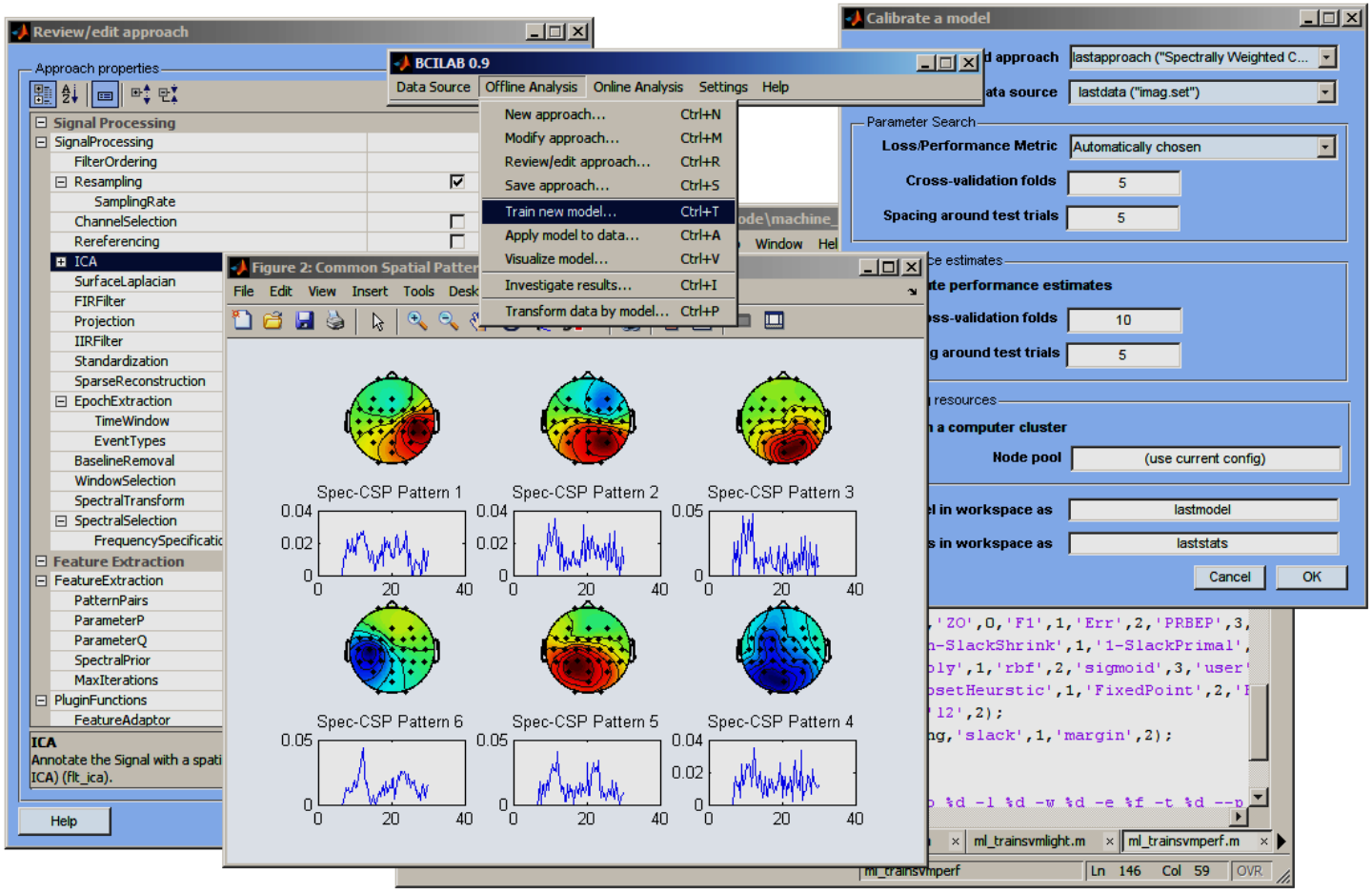
These slides are at:

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

BCILAB Briefing



http://sccn.ucsd.edu/**wiki/BCILAB**

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, …)

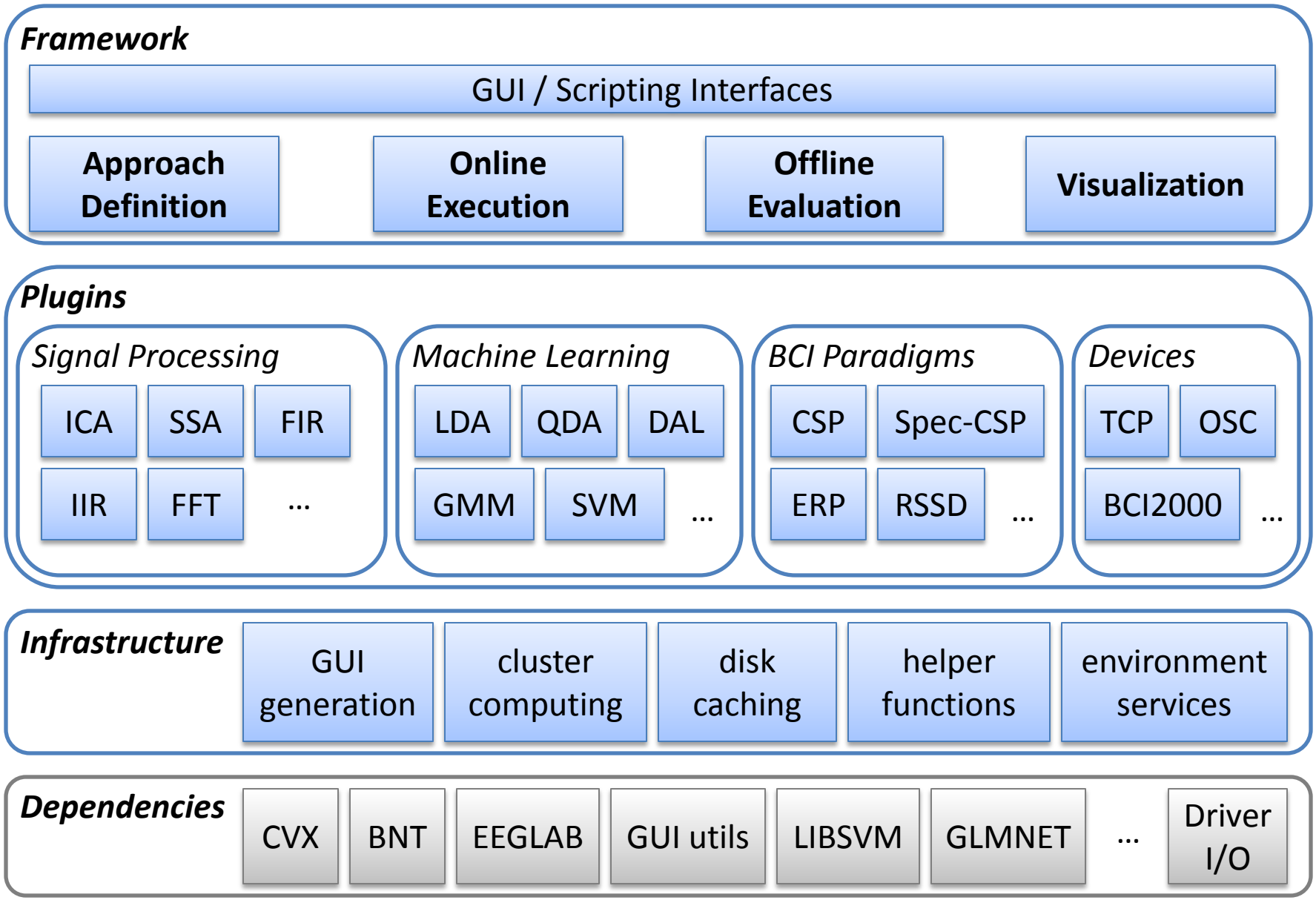
Office of Naval Research

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

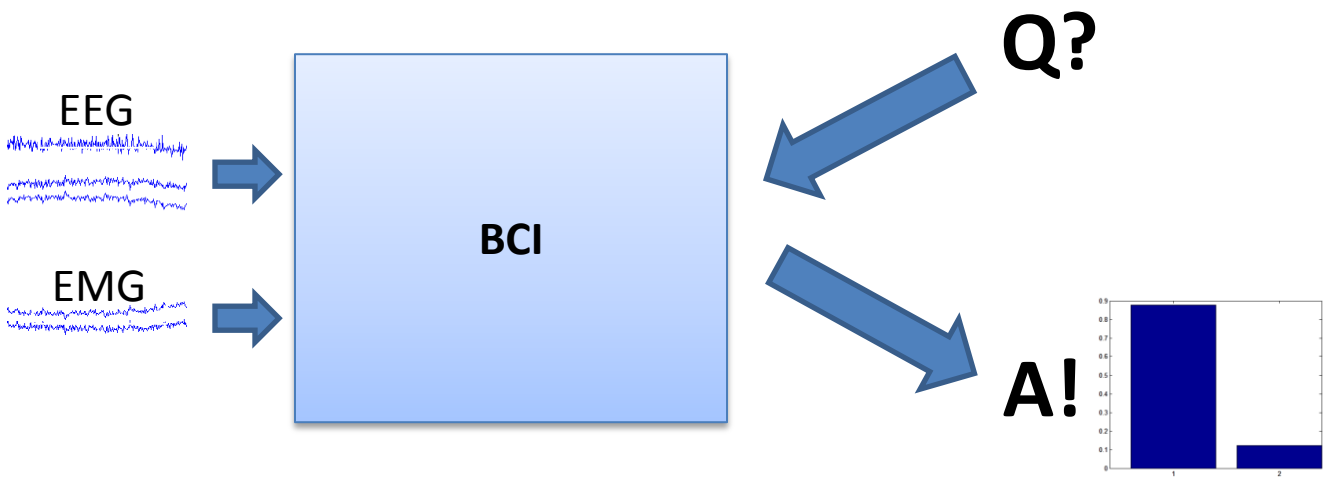
(*: not yet in the current release)

BCILAB Components



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

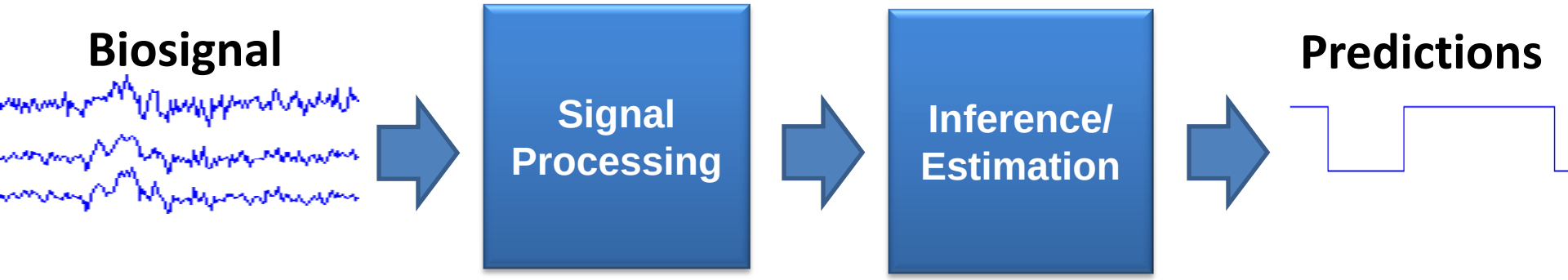


Signal Processing?

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

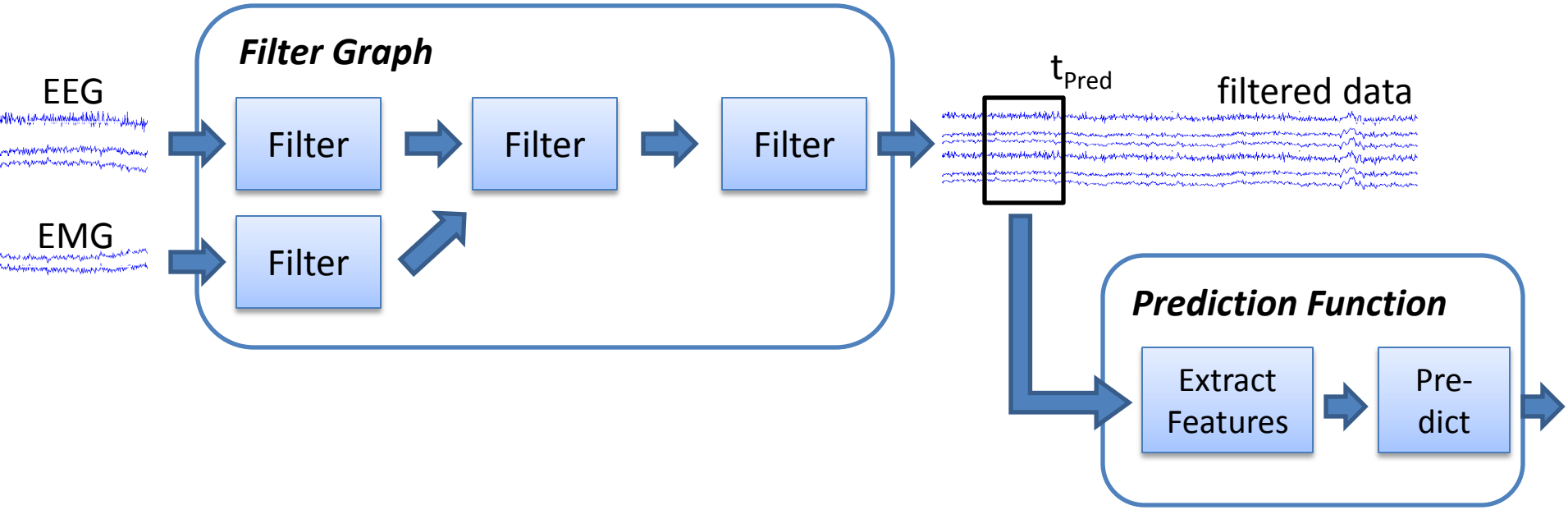
Signal Processing?

- 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



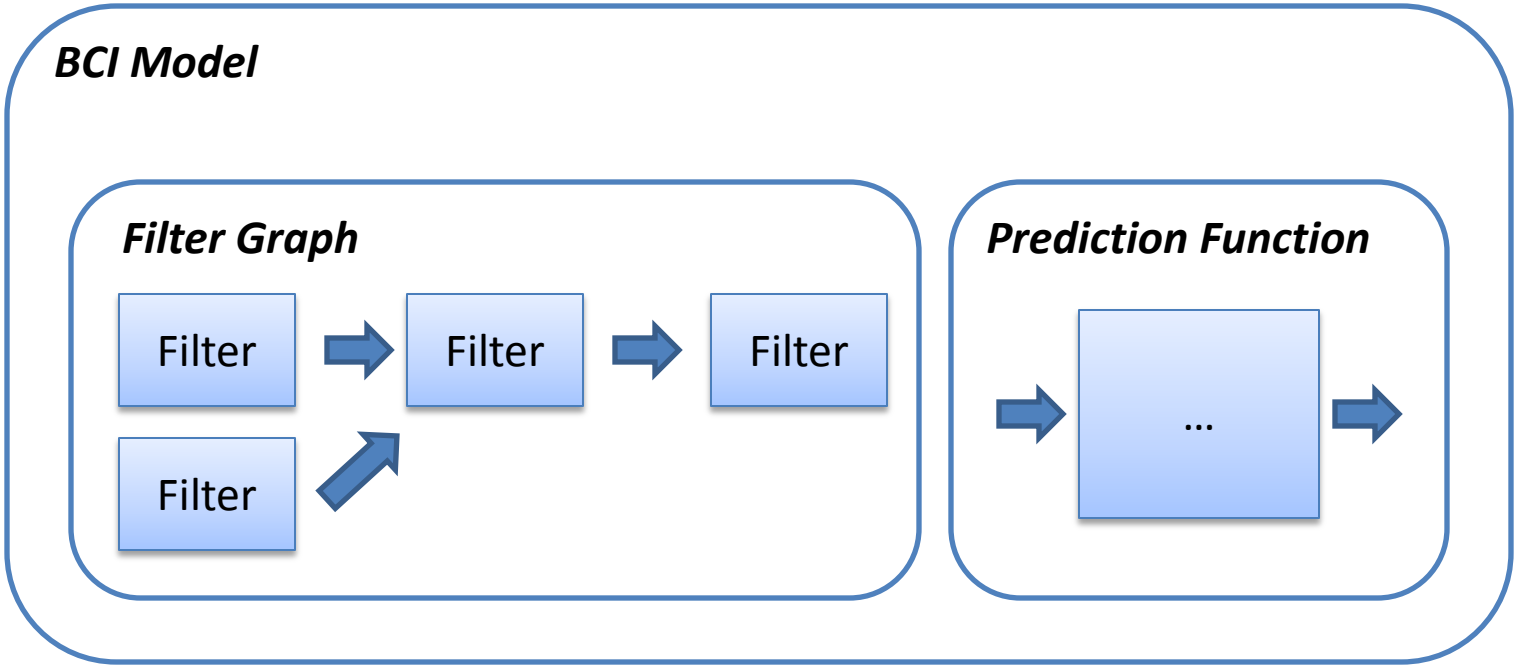
Online Data Flow

- 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



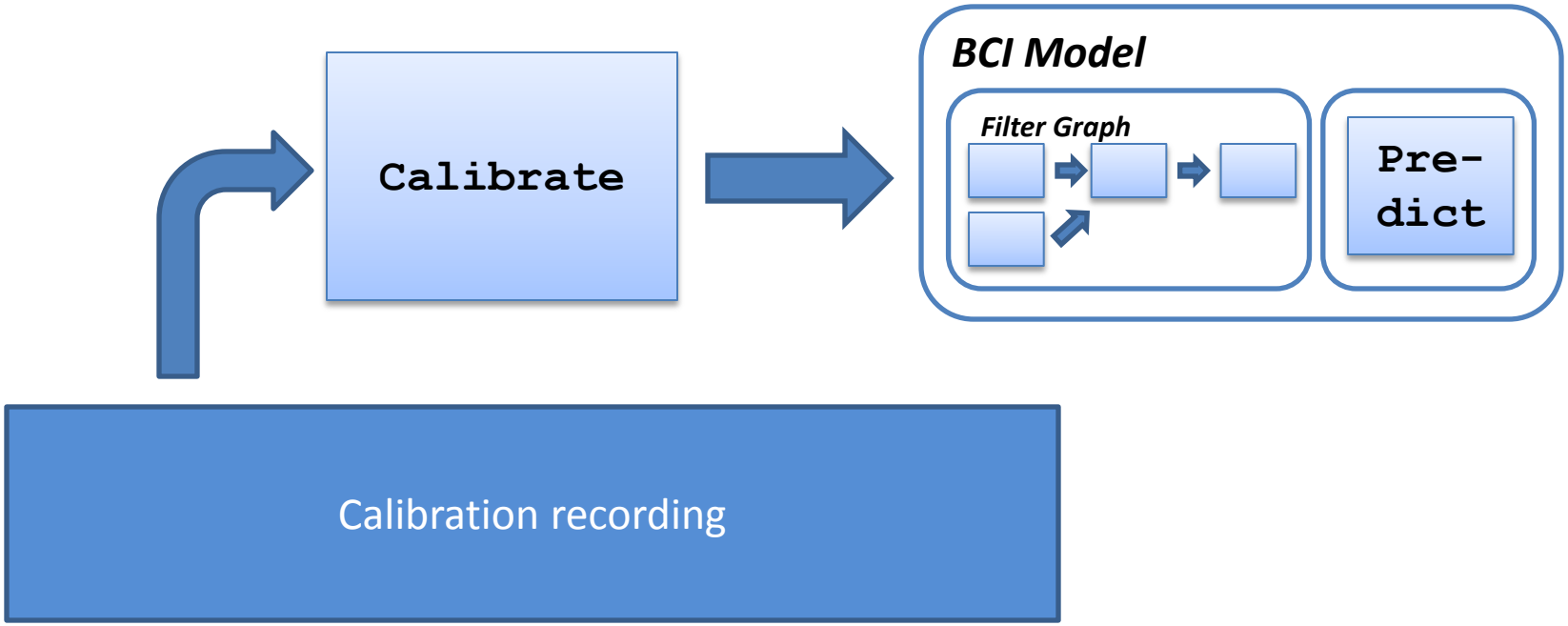
BCI Models

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



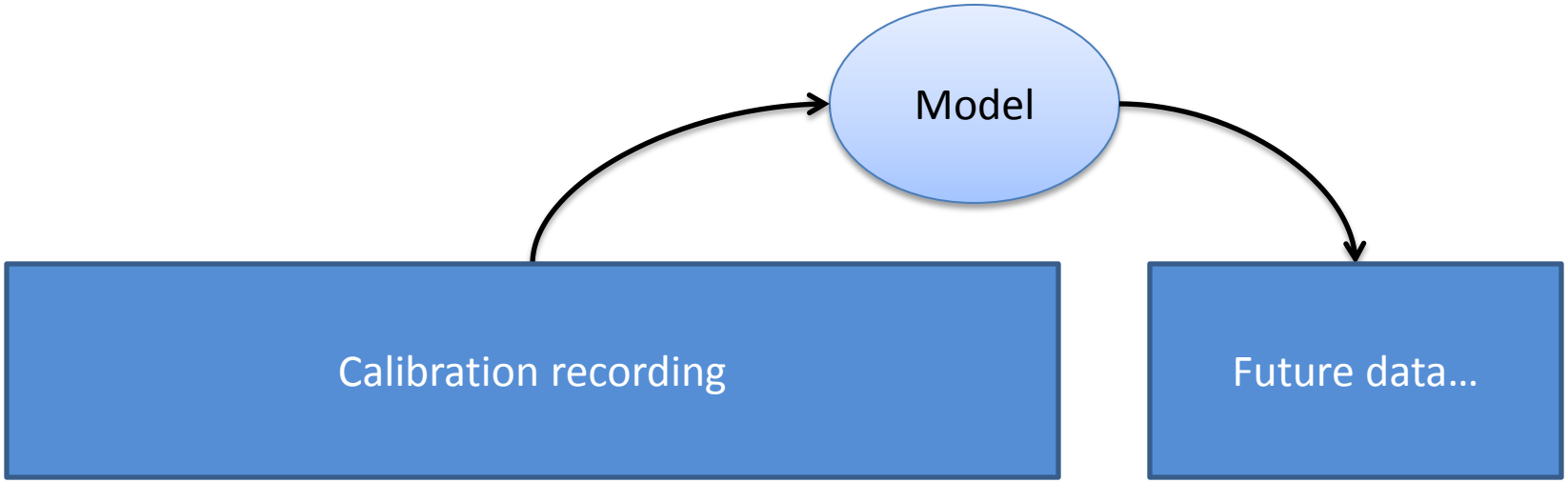
BCI Paradigms

- 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., mis-classification rate)



Offline Evaluation

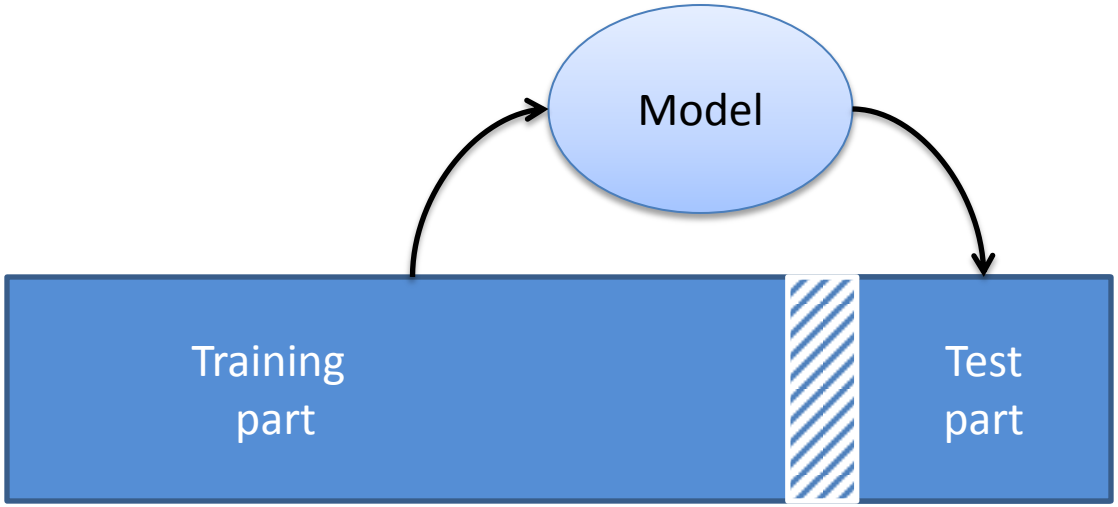
- Evaluation of computational approaches on a **single** data set?

?

Calibration recording

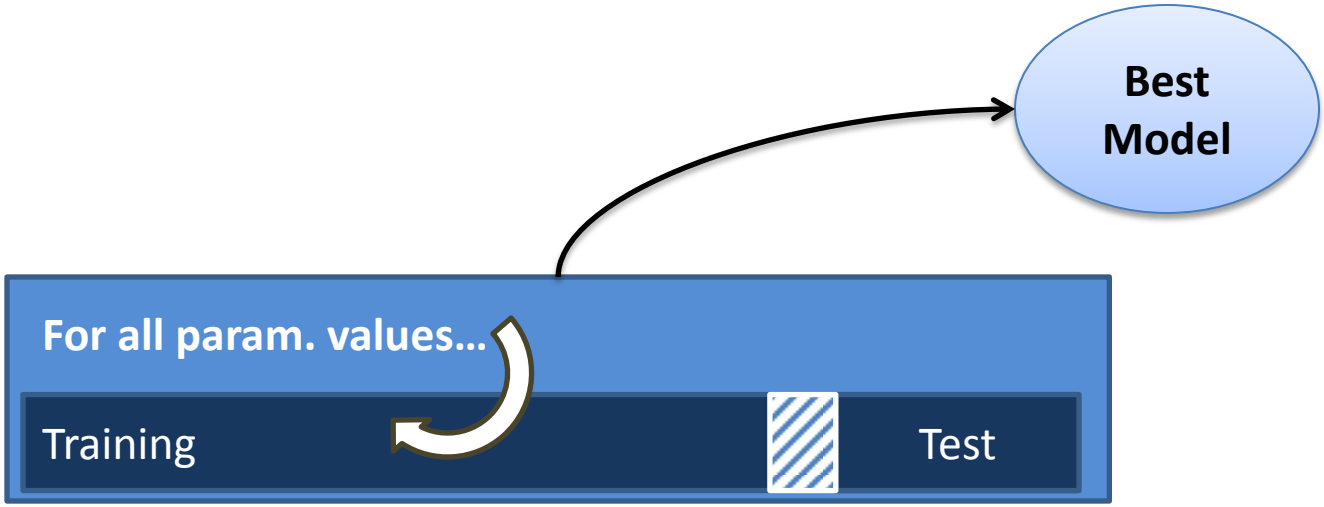
Offline Evaluation

- 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*



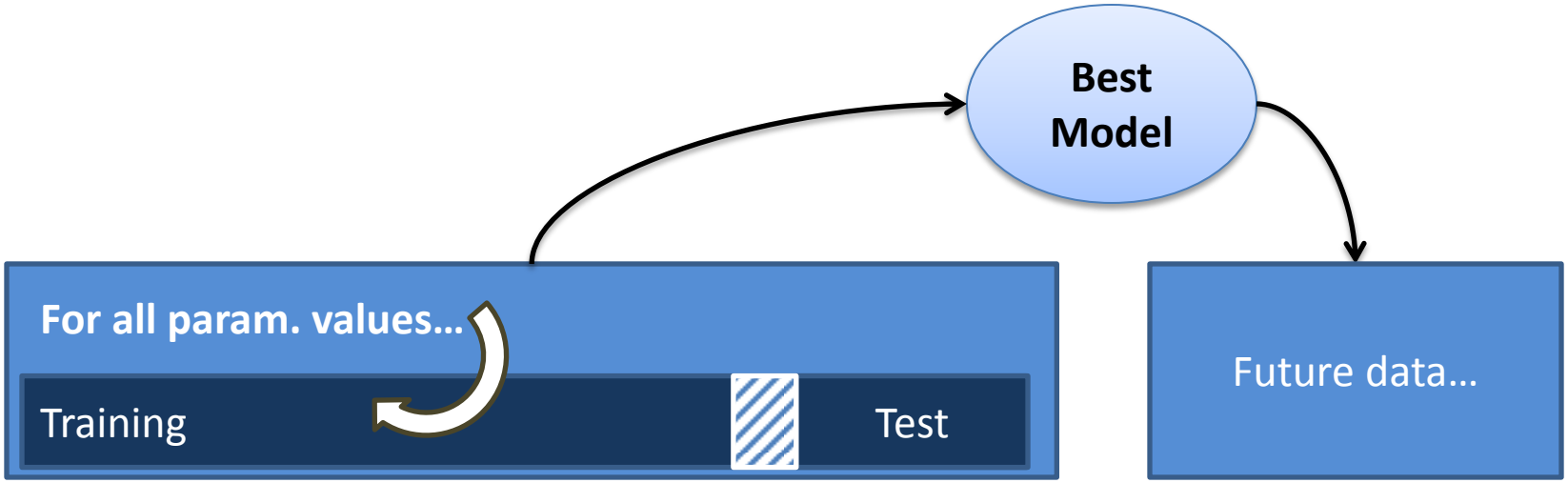
Resolving Free Parameters

- 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!)



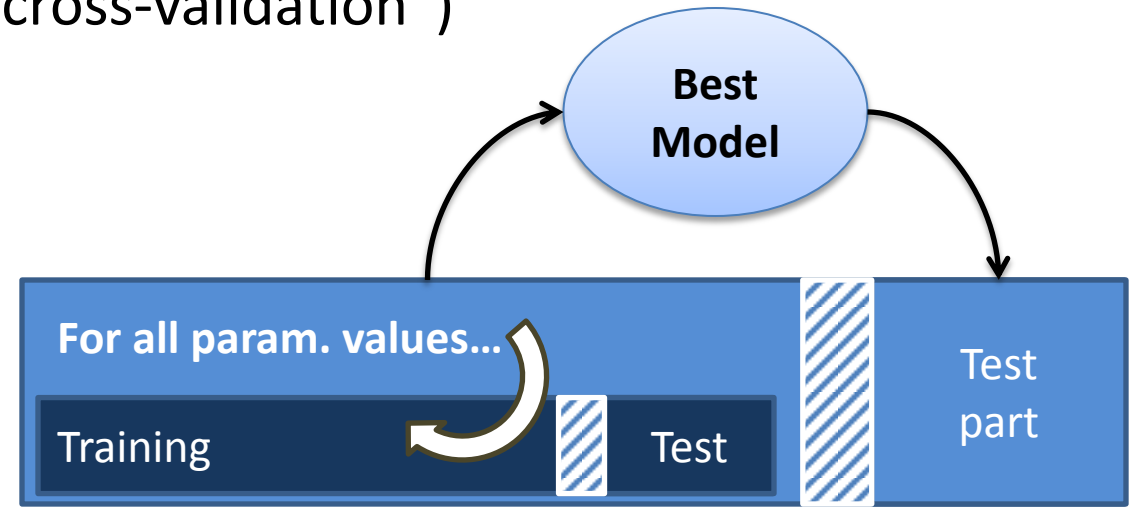
Resolving Free Parameters

- 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

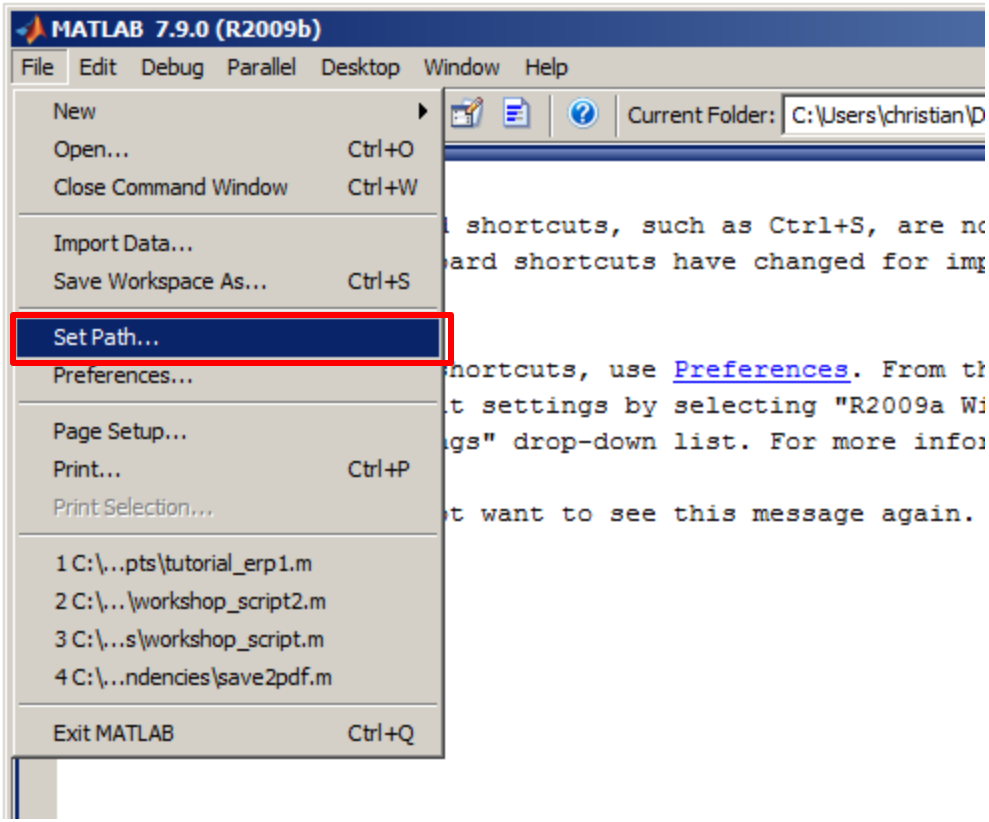
- MATLAB 2008a
- 1GB+ RAM (better: 2GB+)
- Windows, Linux, or Mac
- For smooth workshop: **No** toolboxes in MATLAB path other than Mathworks toolboxes (or EEGLAB)
- To use certain additional features (not covered today): Signal Processing Toolbox, Statistics Toolbox, Real-time experimentation environment (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

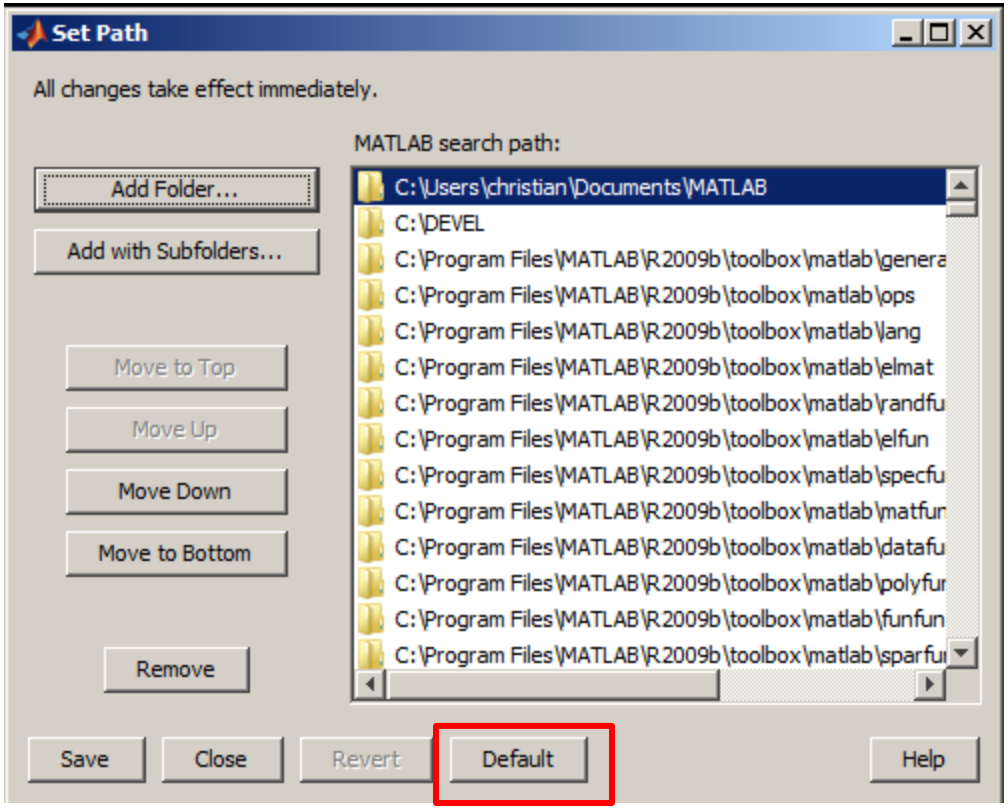
Rawr!

Clearing the Path



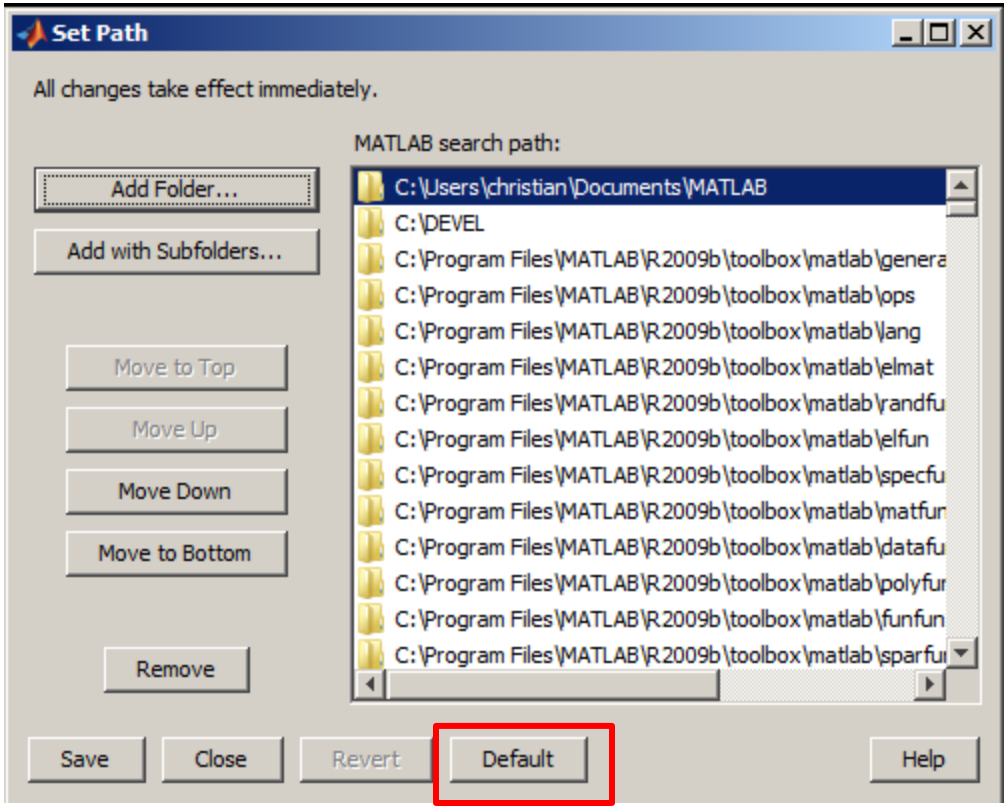
Swartz Center for Computational Neuroscience

Clearing the Path

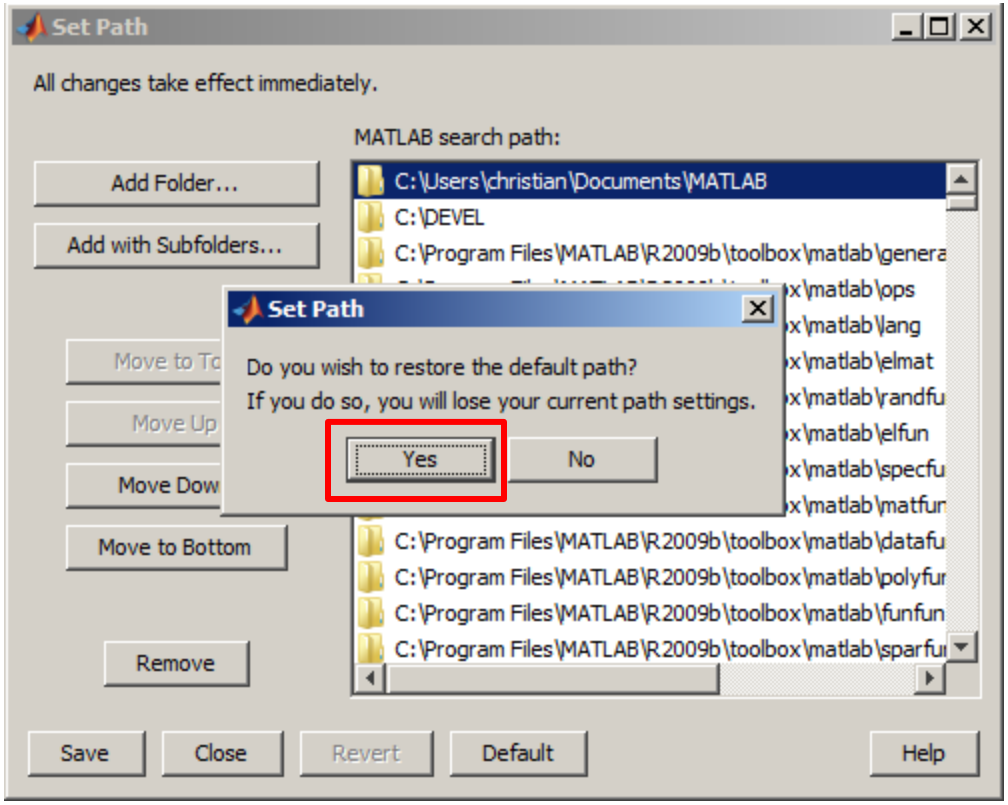


Swartz Center for Computational Neuroscience

Clearing the Path

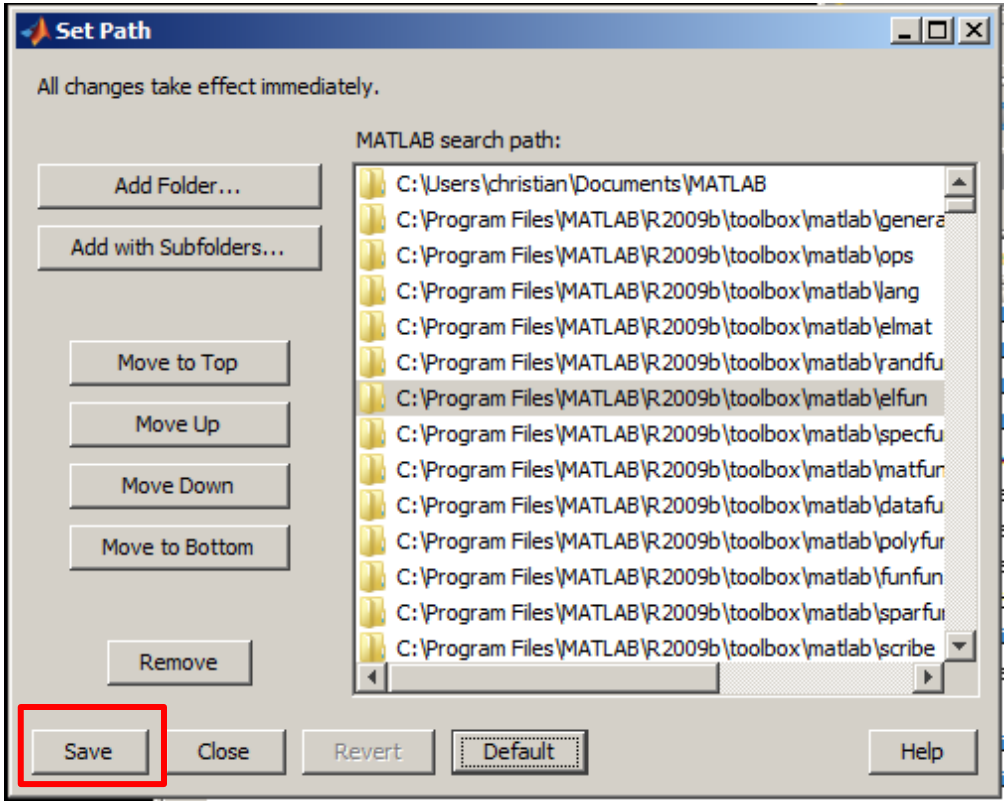


Clearing the Path



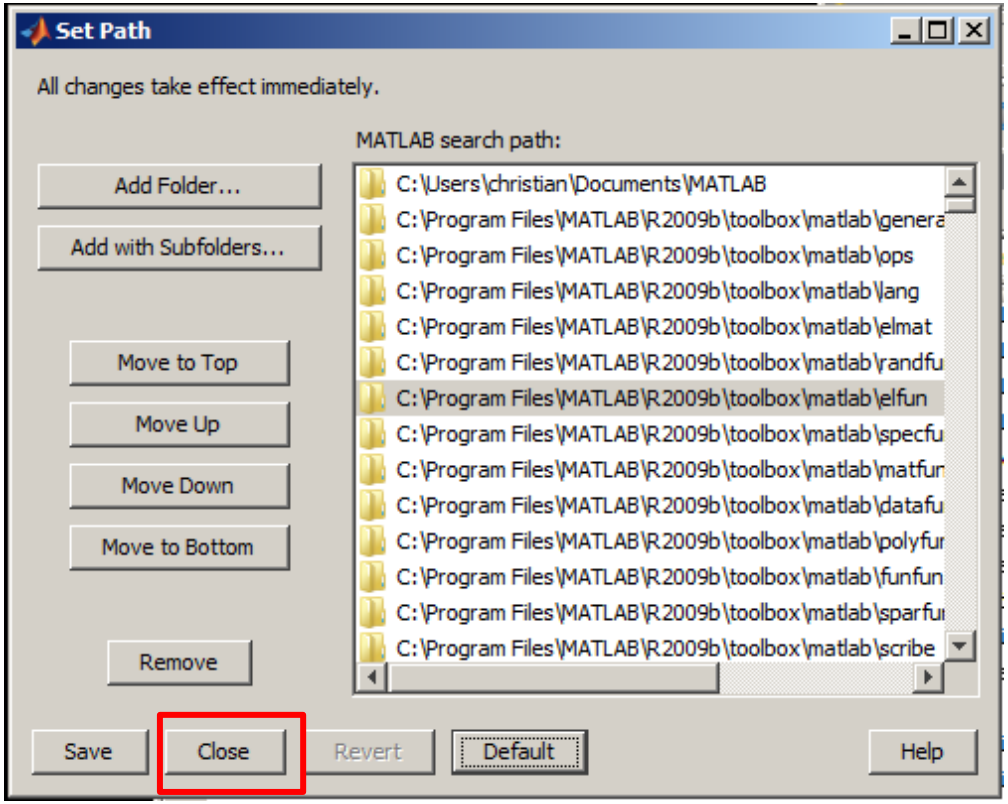
Swartz Center for Computational Neuroscience

Clearing the Path



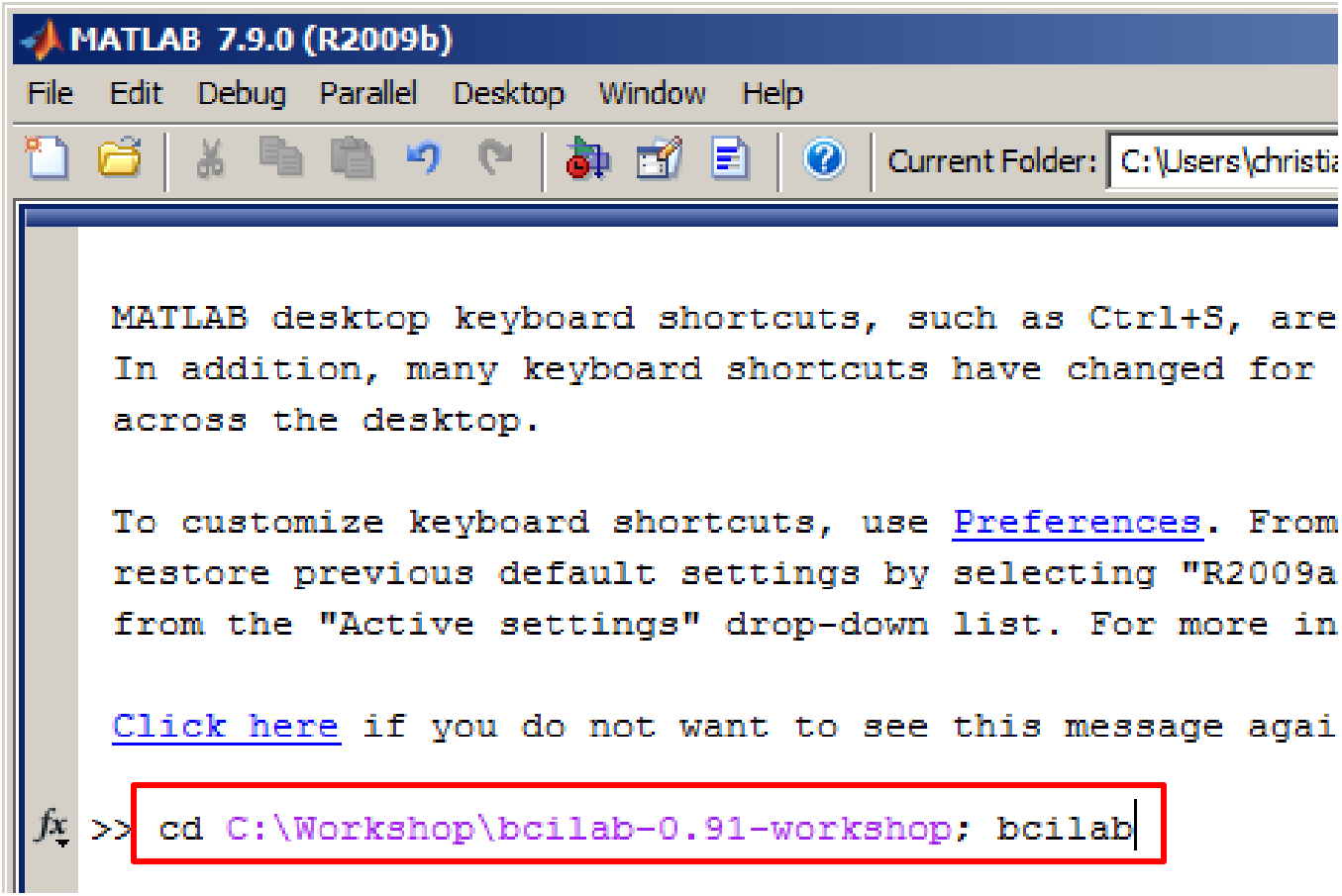
Swartz Center for Computational Neuroscience

Clearing the Path



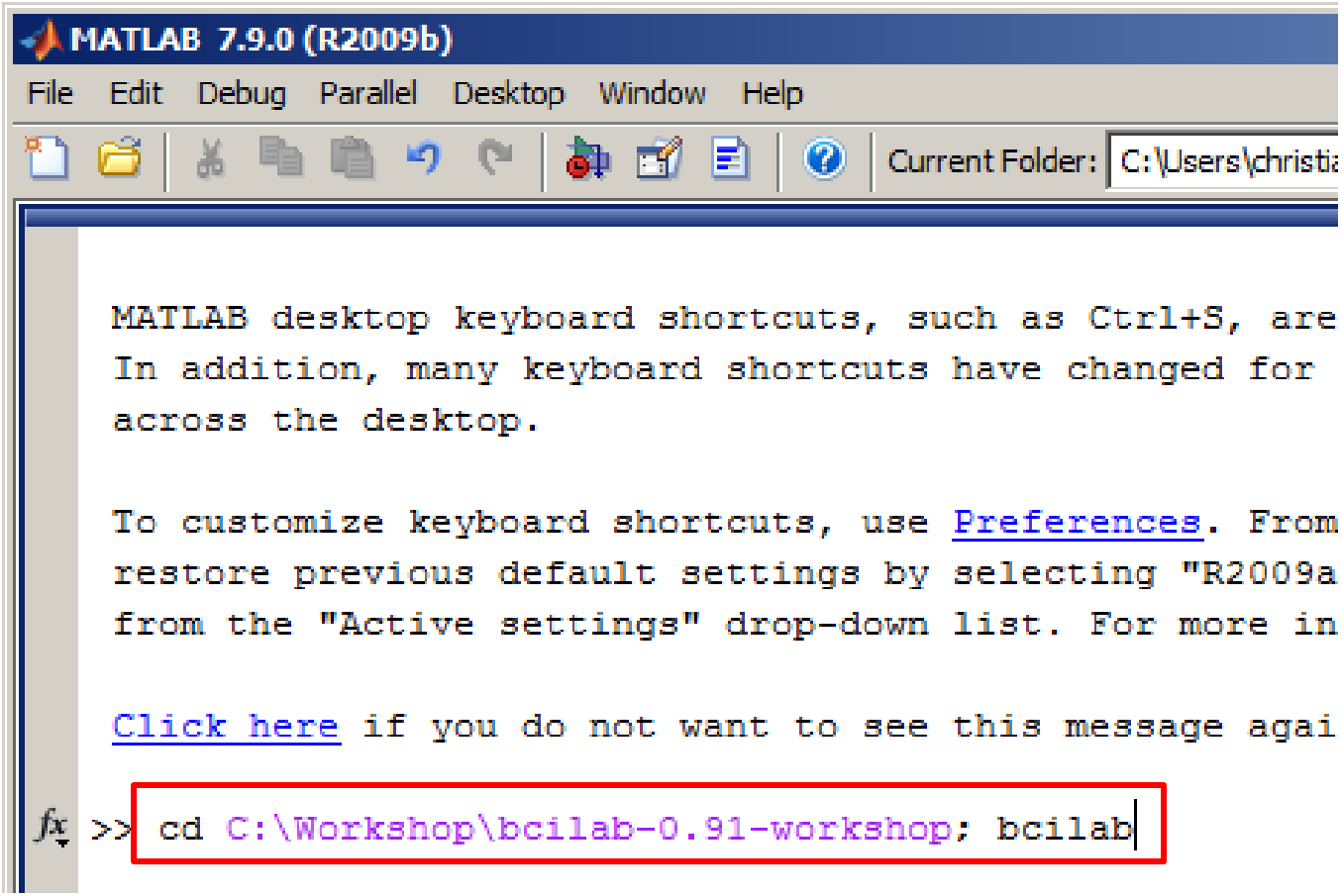
Starting the Toolbox

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



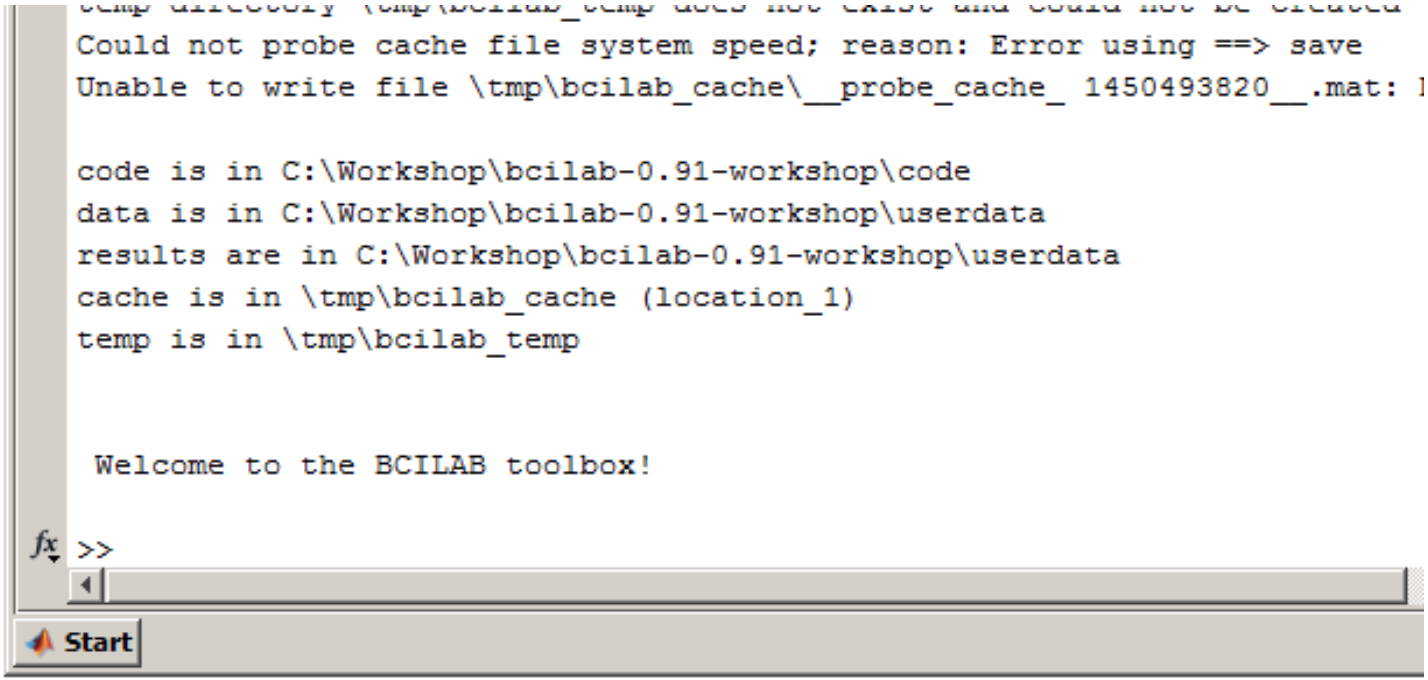
Starting the Toolbox

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



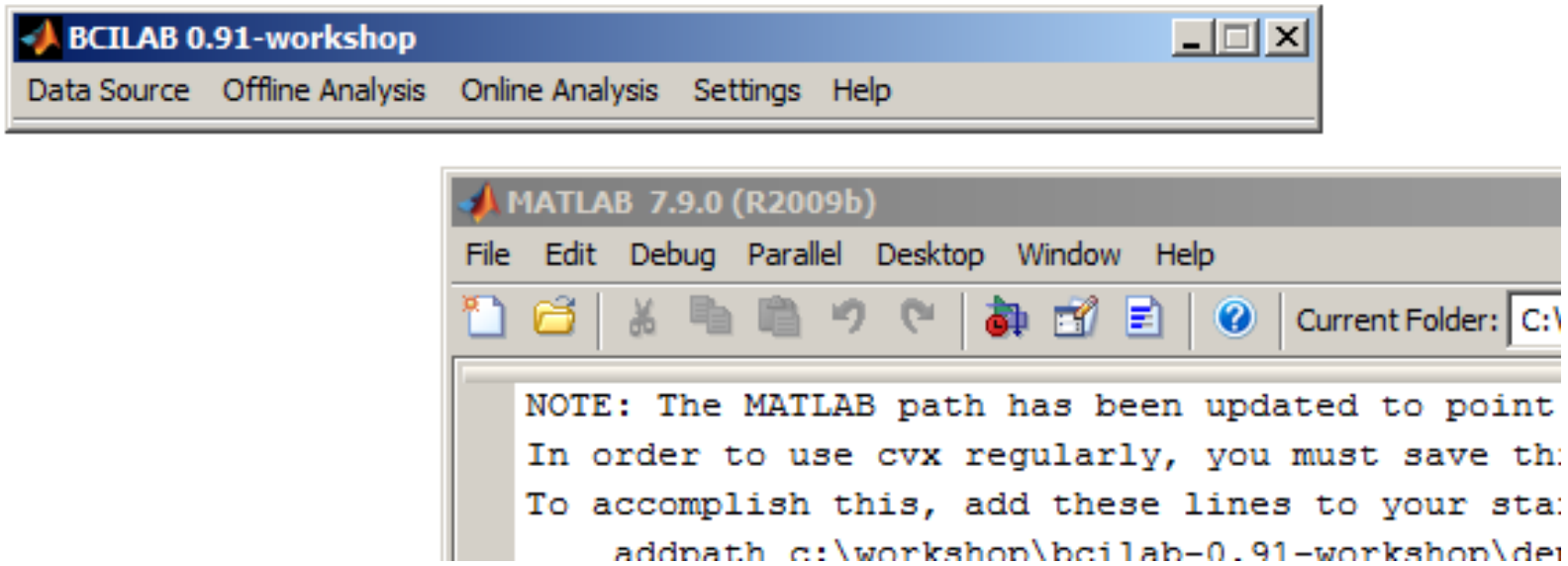
Starting the Toolbox

- You should see the welcome message



Starting the Toolbox

- … and the main menu

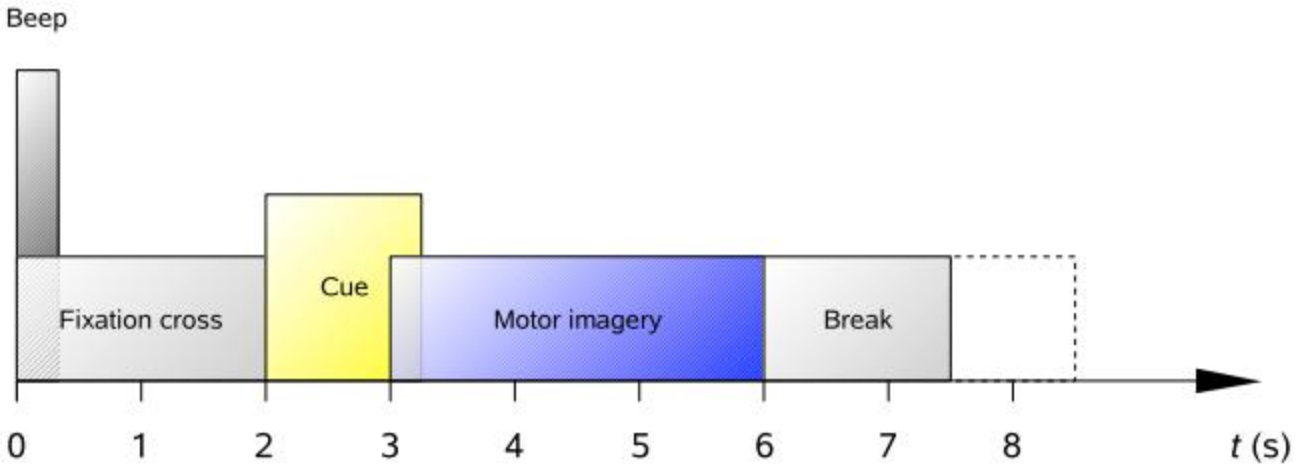


The Data

- 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

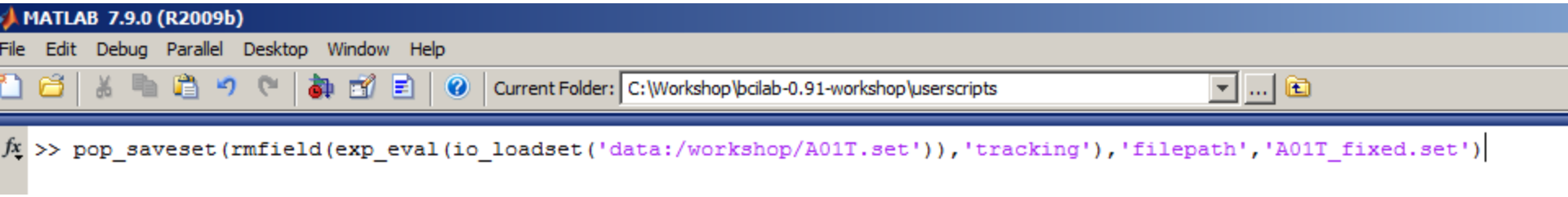
The Data

- Timing overview



Fixing the Data ☺

- Execute the following line:



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

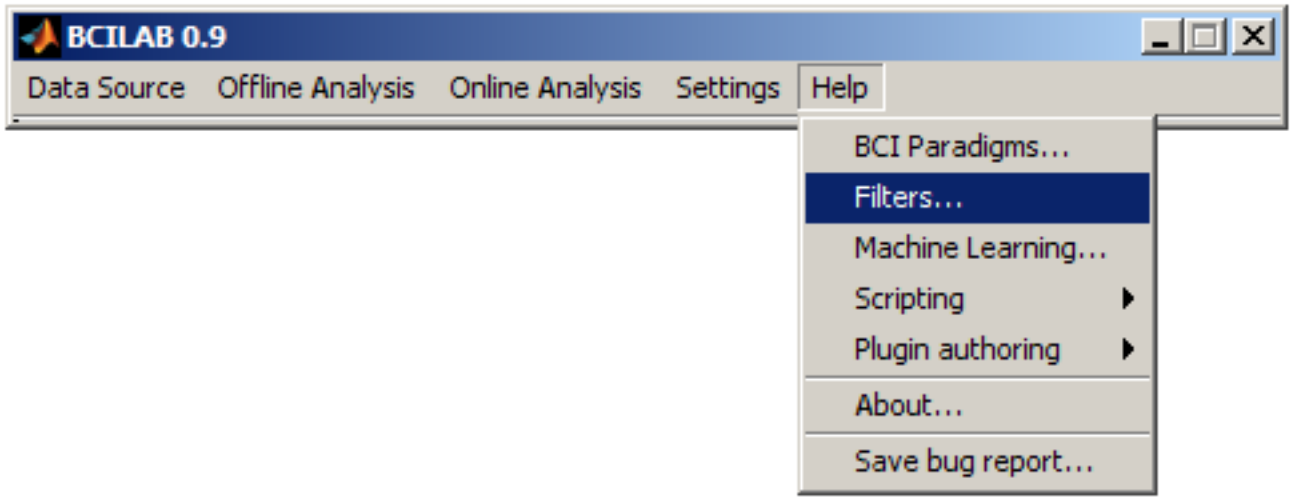
```
pop_saveset(rmfield(exp_eval(io_loadset('data:/workshop/A01T.set')),'tracking'),'filepath','A01T_fixed.set')
```

Fixing the Data ☺

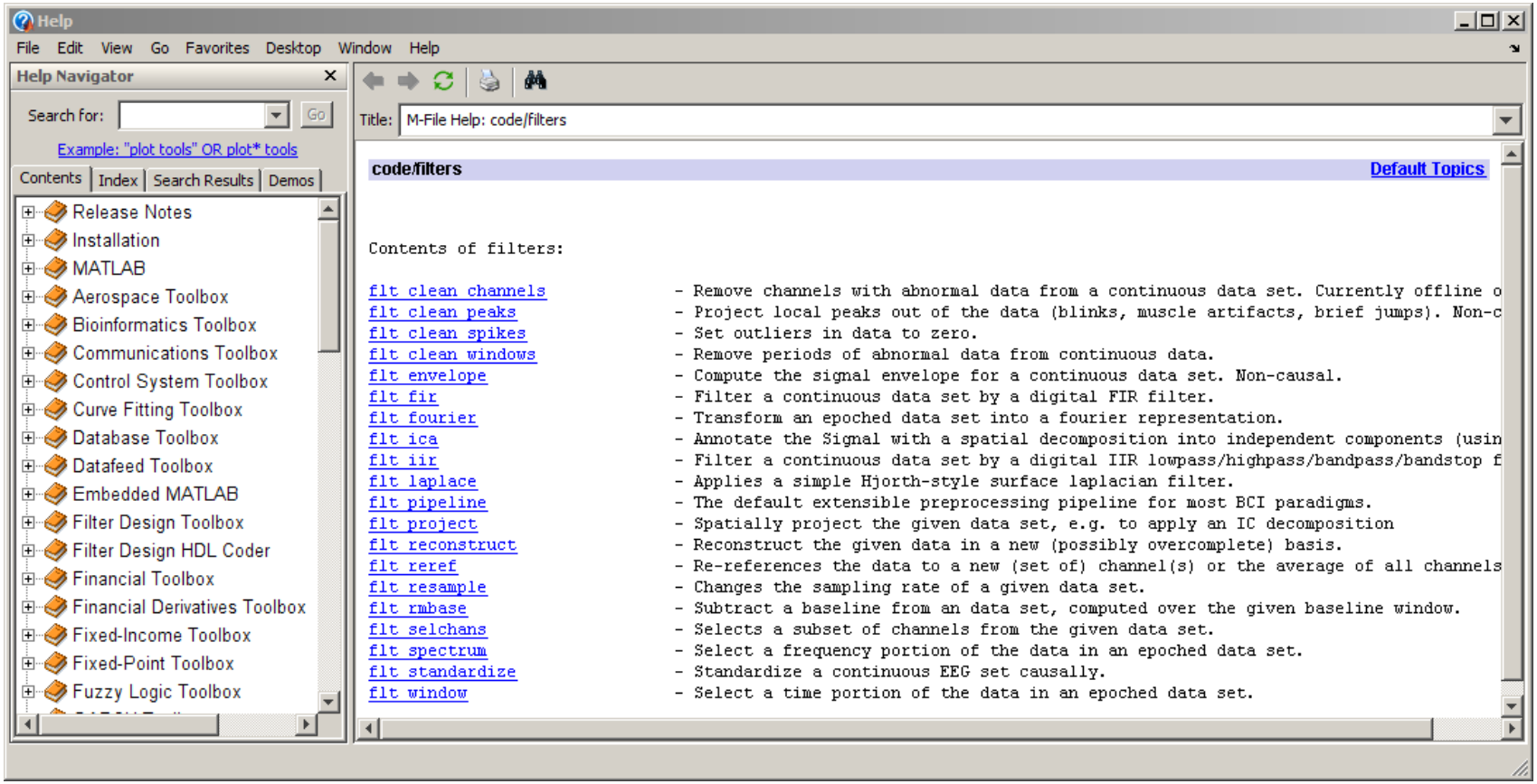
- … and the same for the three other data sets:

```
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')
```

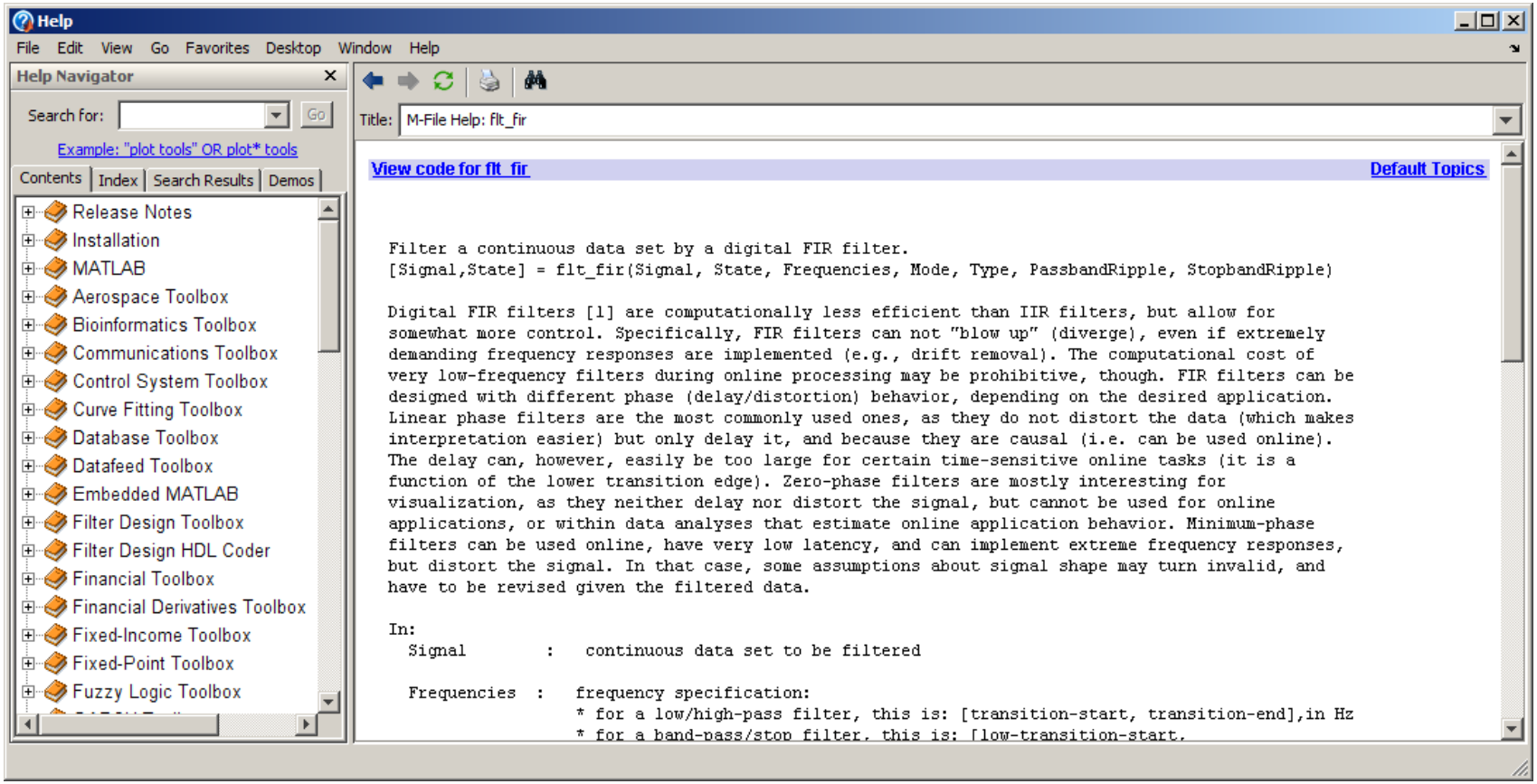
Getting help (if needed)



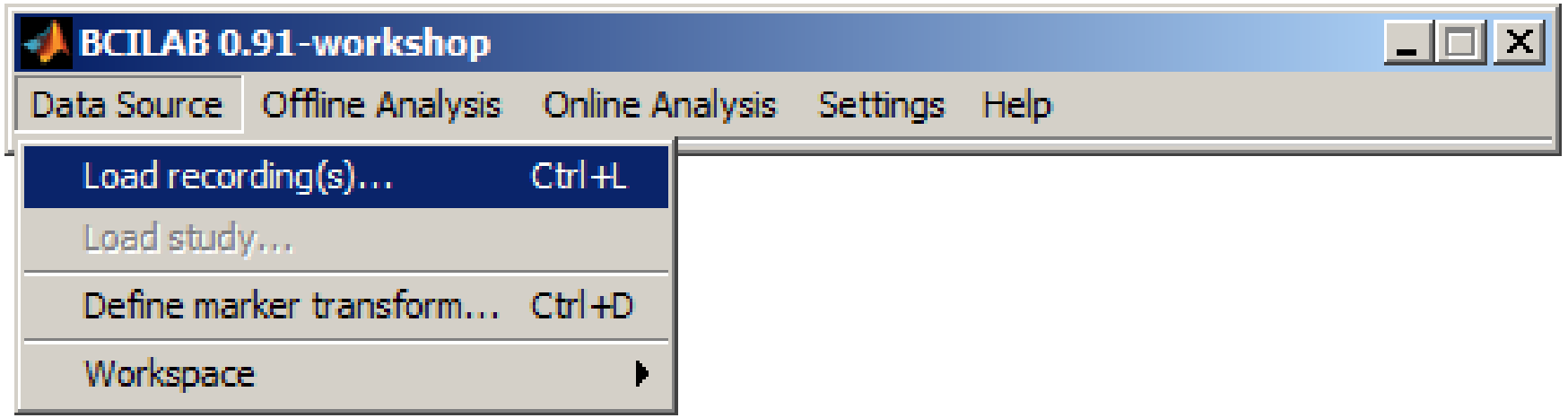
Getting help (if needed)



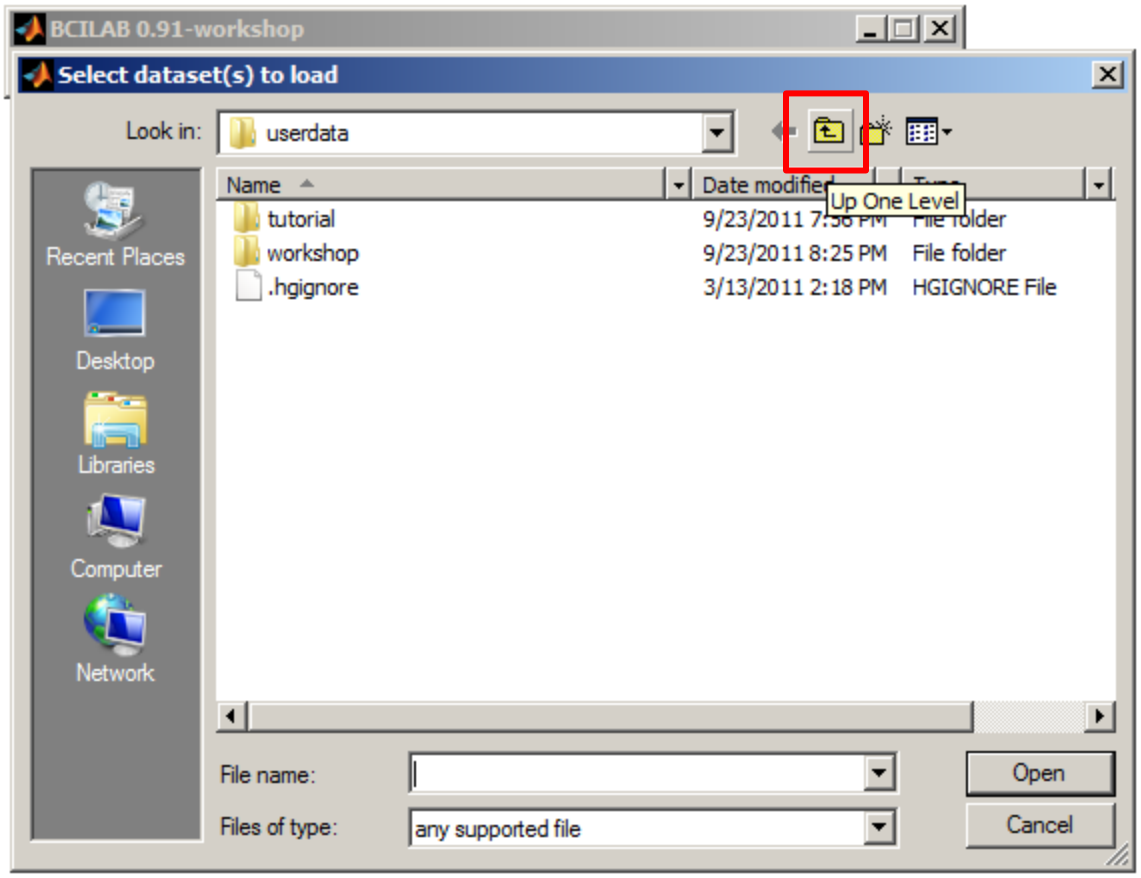
Getting help (if needed)



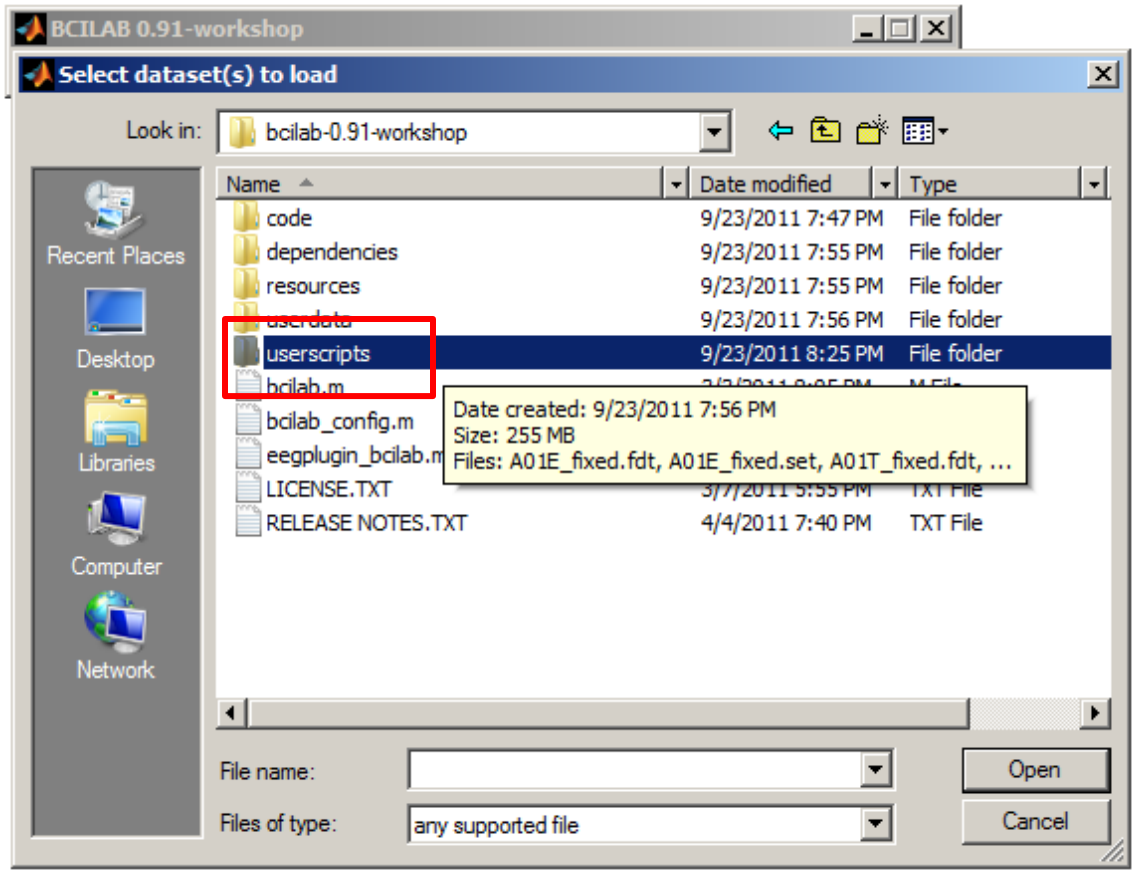
Loading the Data



Loading the Data

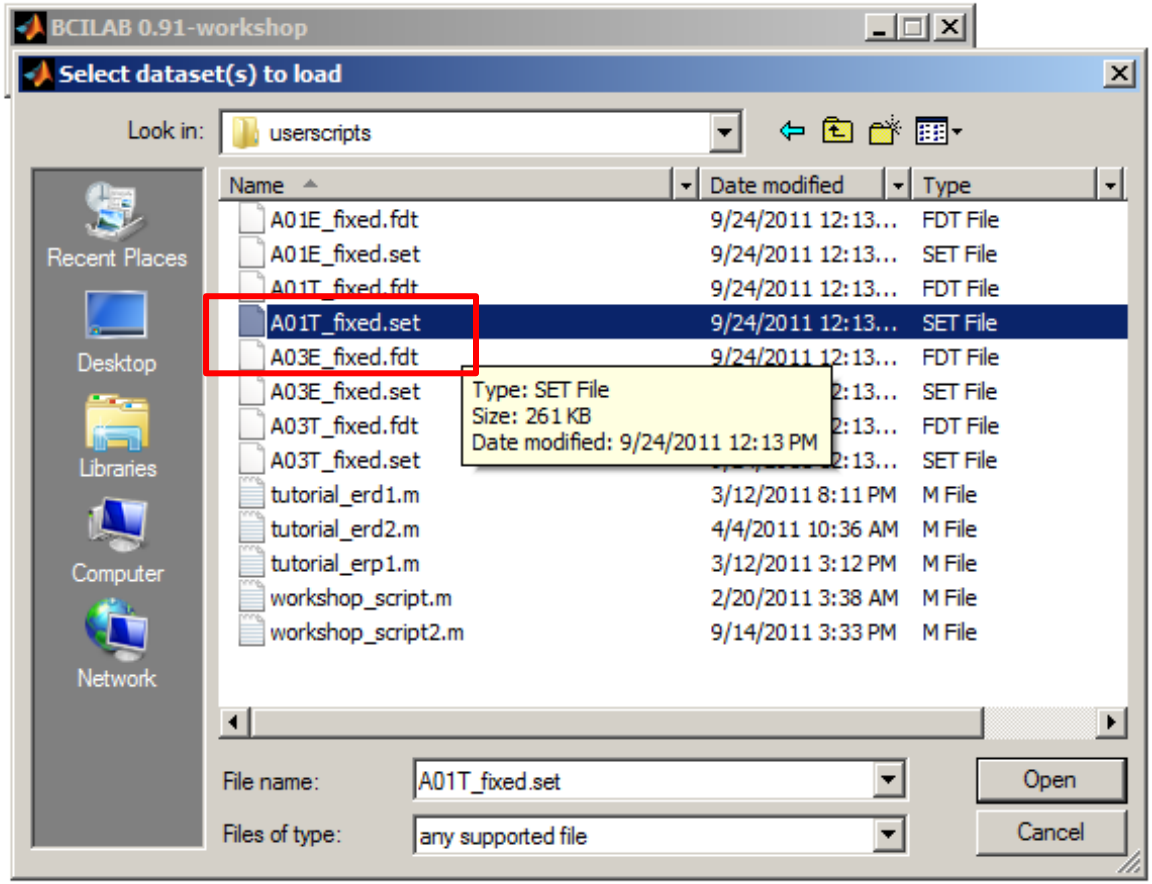


Loading the Data

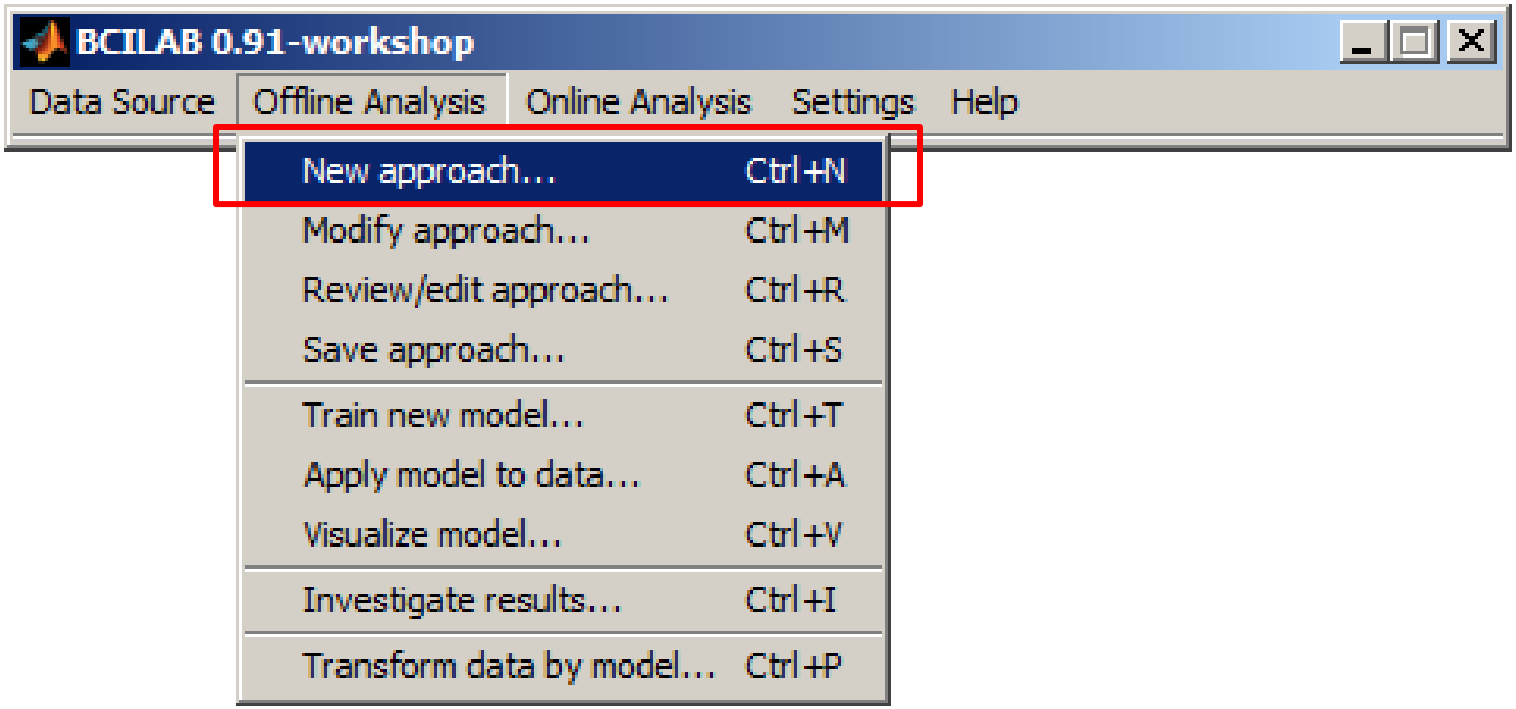


Loading the Data

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

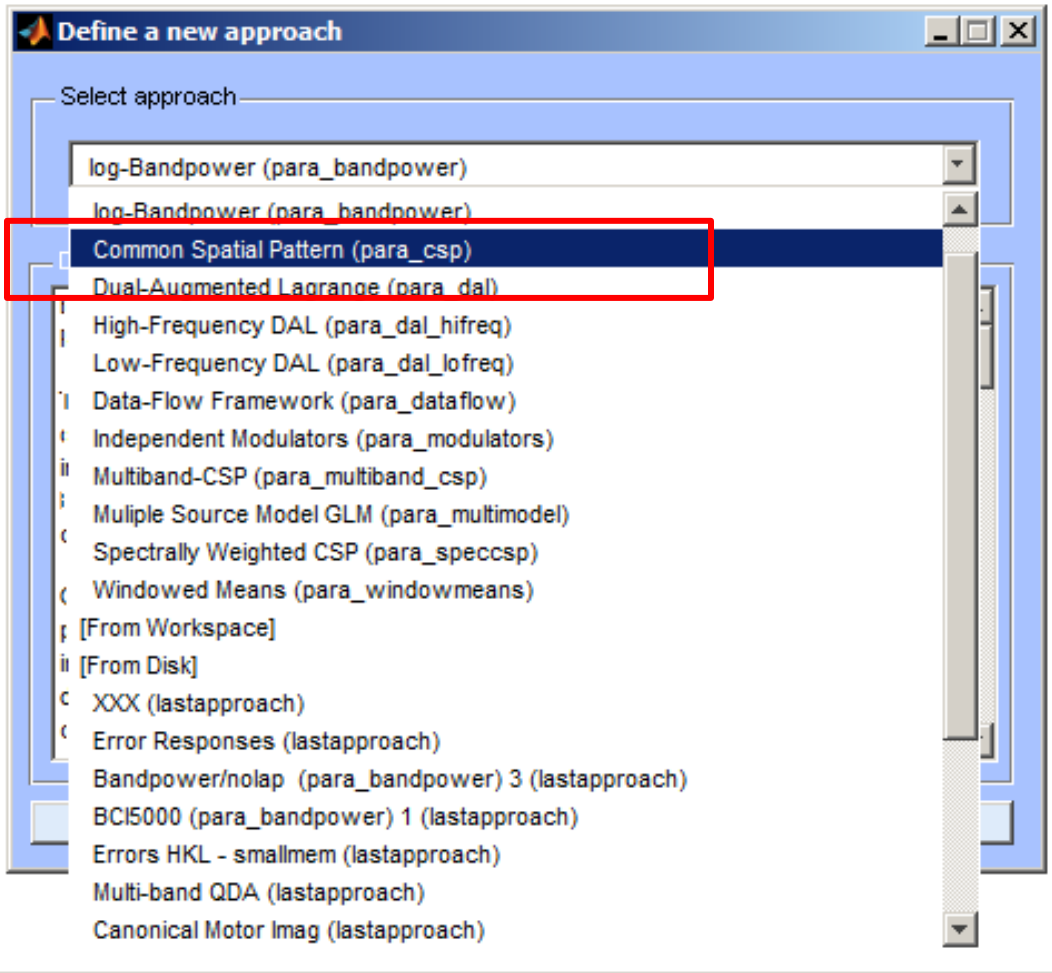


Defining a new Approach

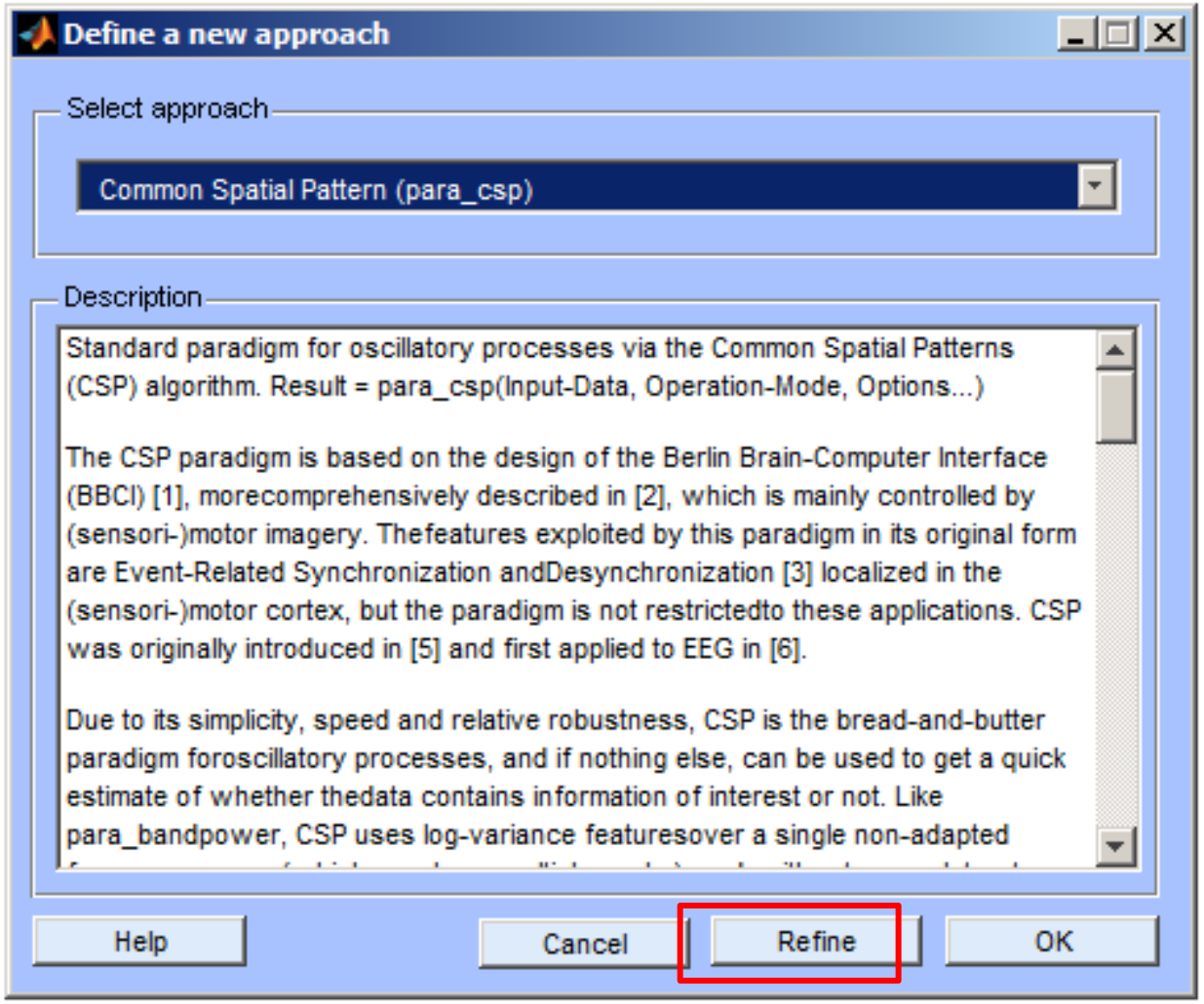


Defining a new Approach

- Let's start with Common Spatial Patterns:

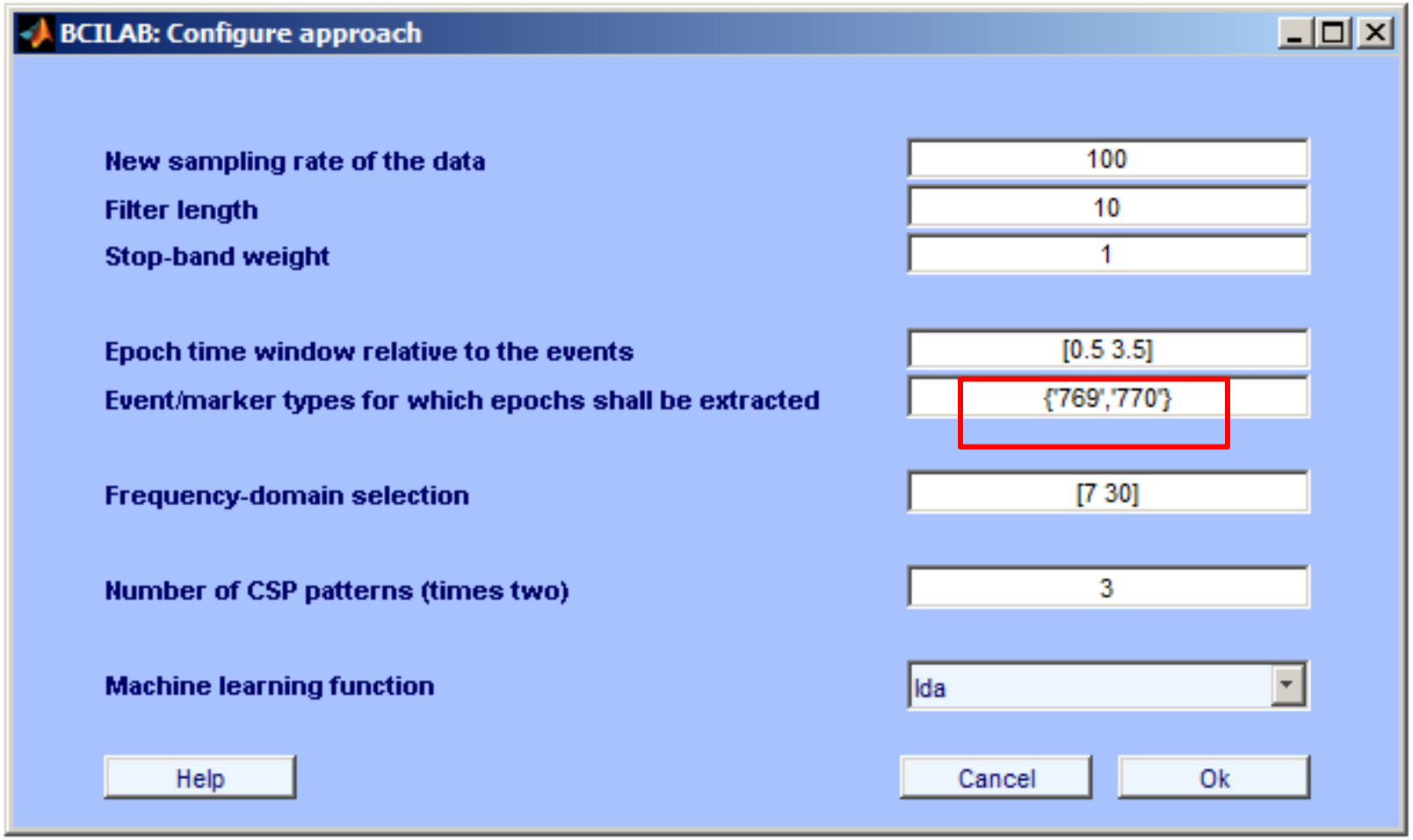


Defining a new Approach

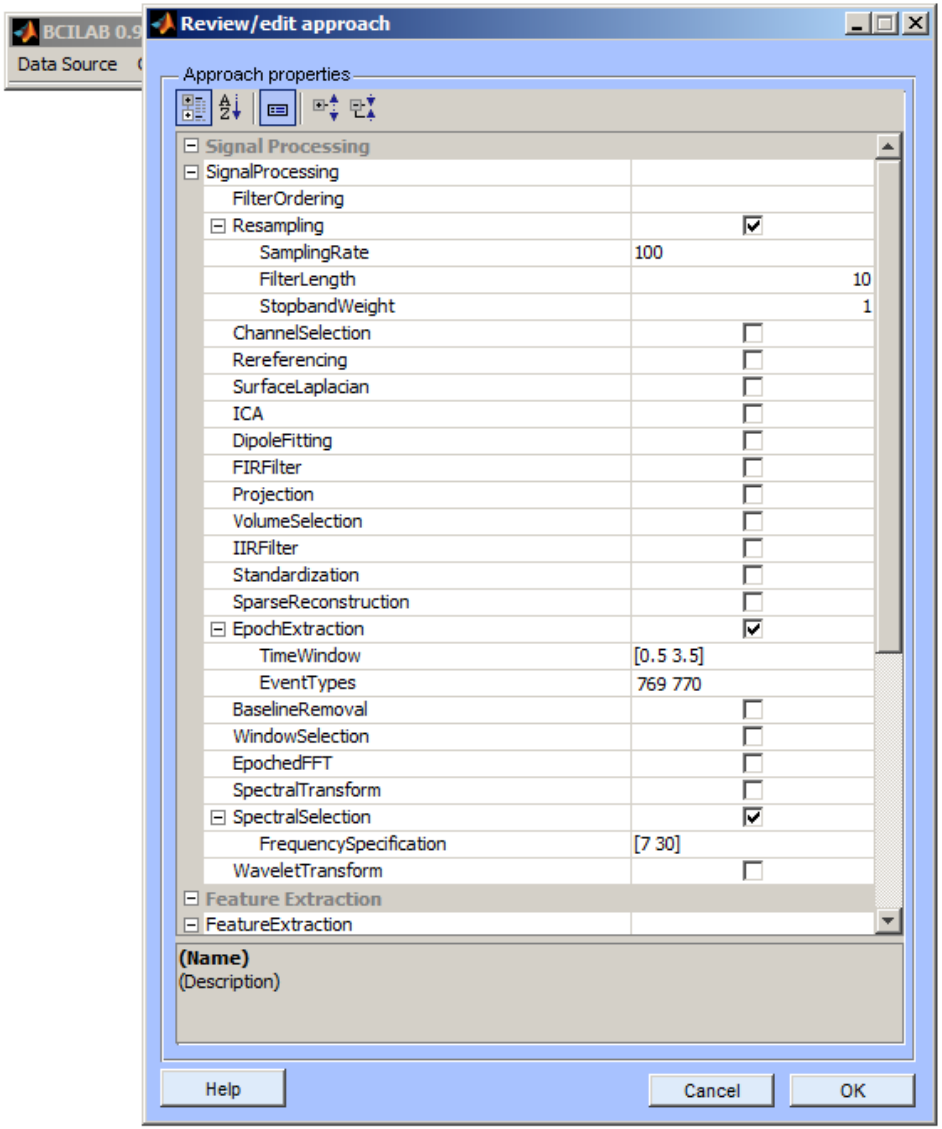


Adapting an Approach

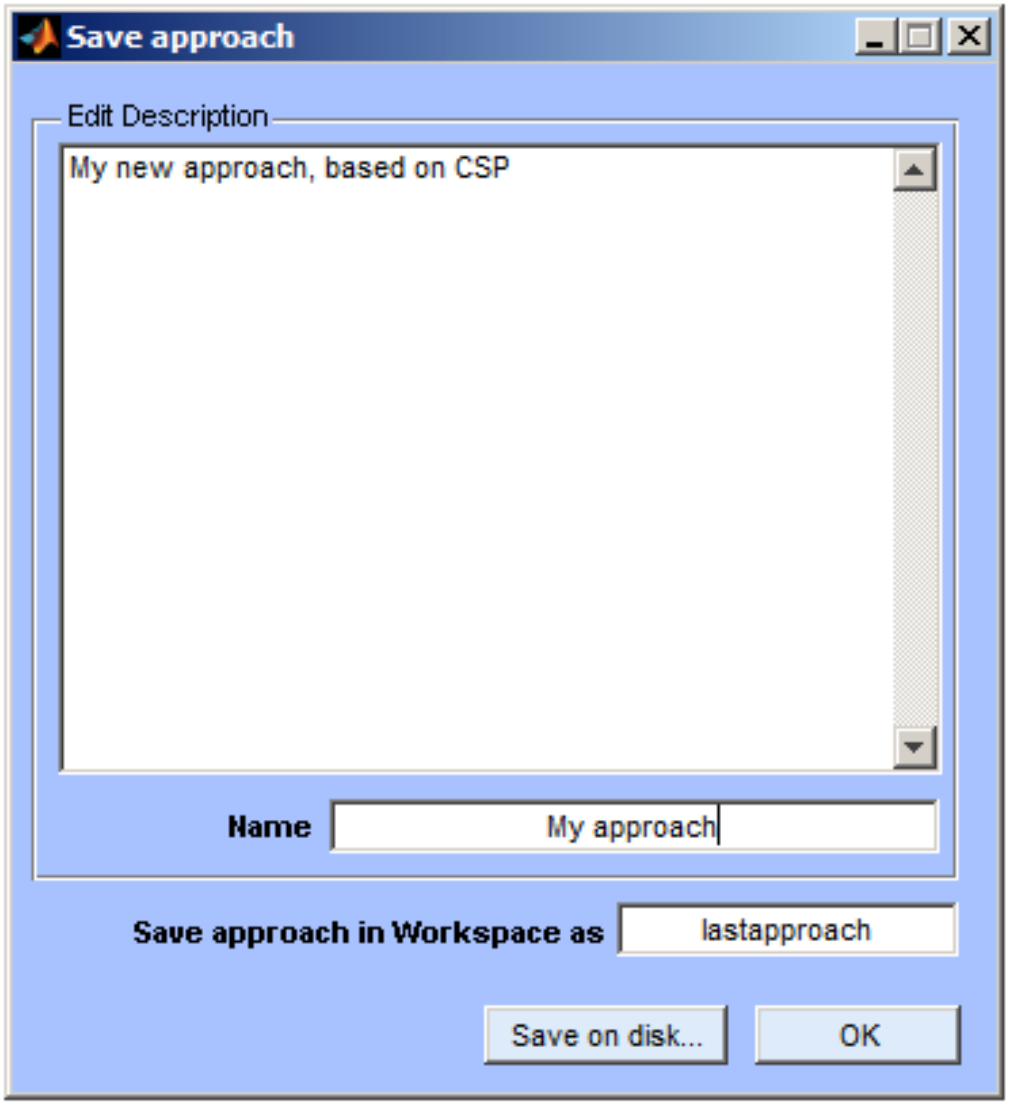
- Marker names are {‘769’,’770’} in this set:



Further Editing an Approach

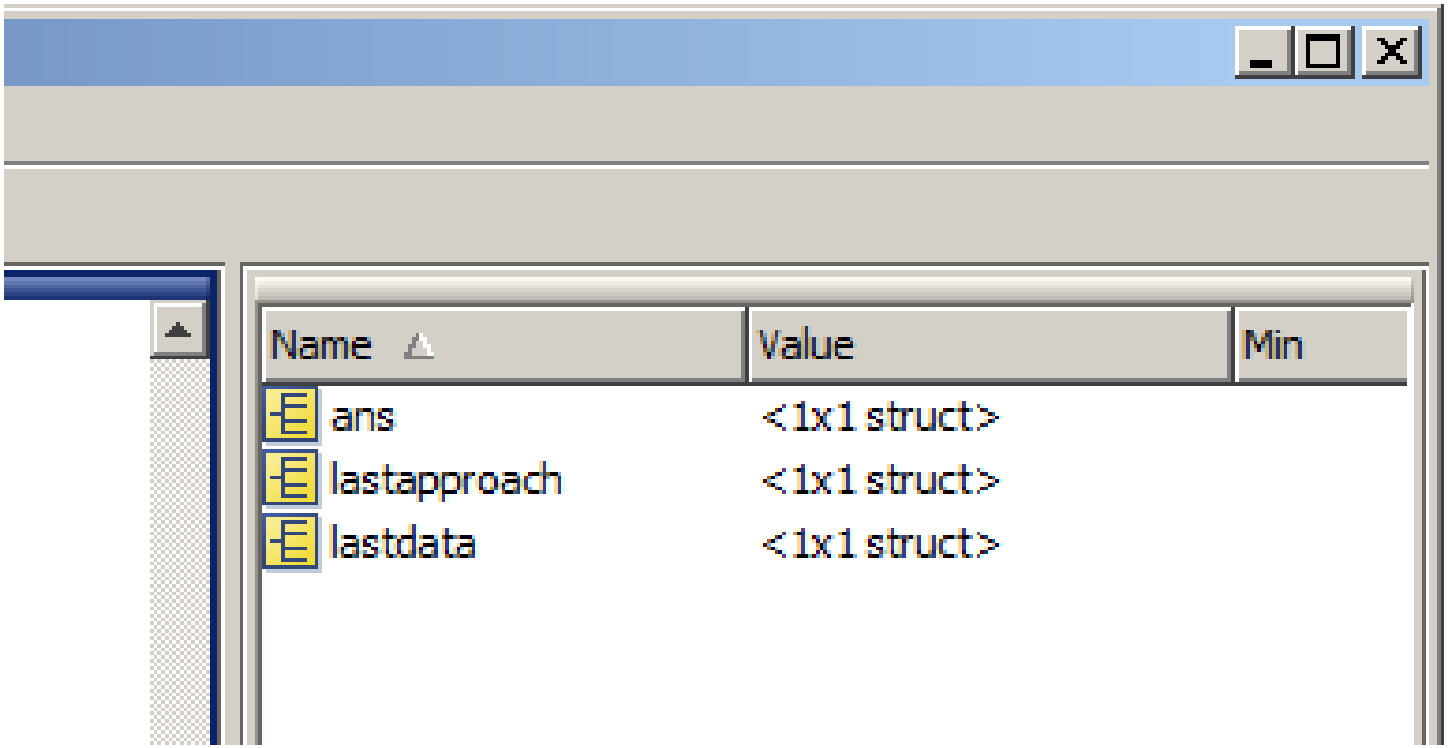


Saving the Approach

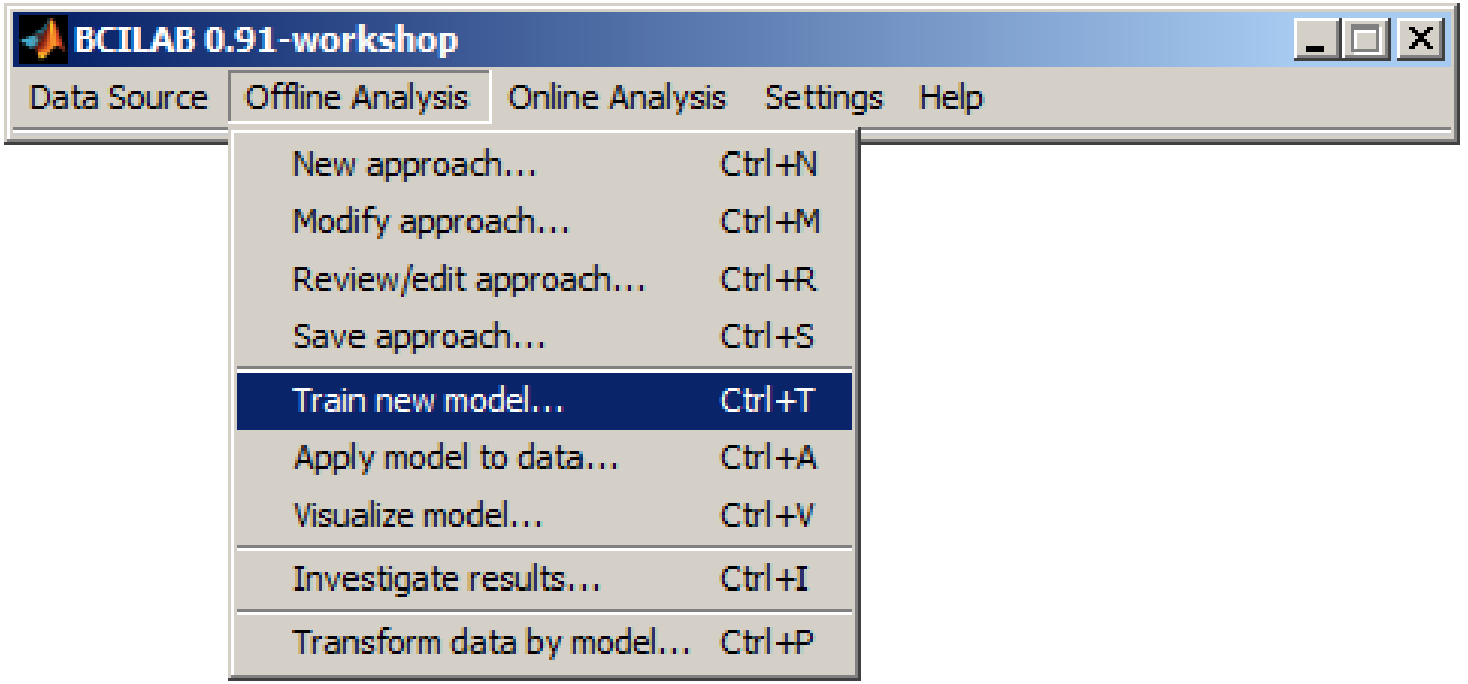


See also: MATLAB's Workspace

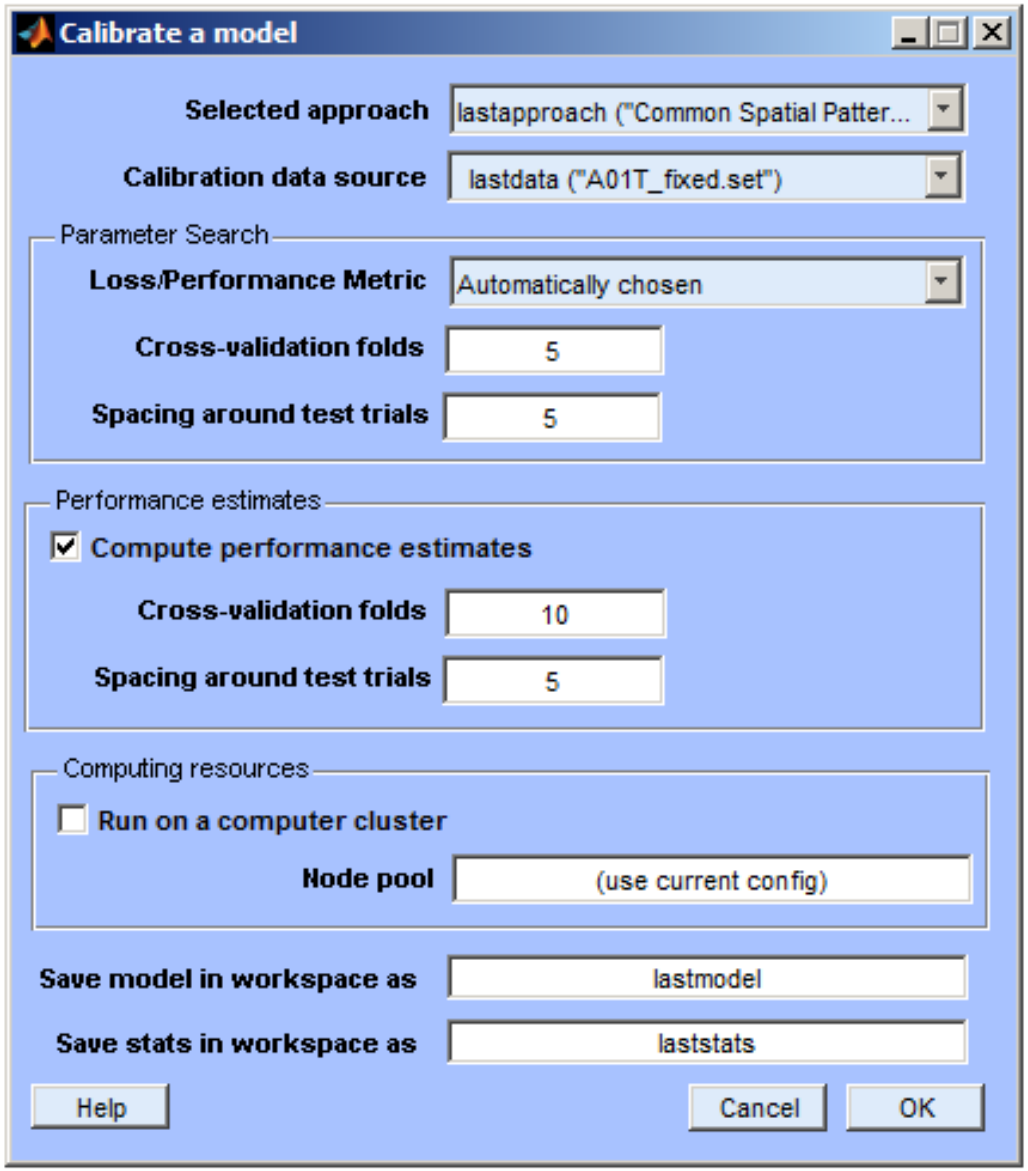
- Can be enabled via "Desktop" menu



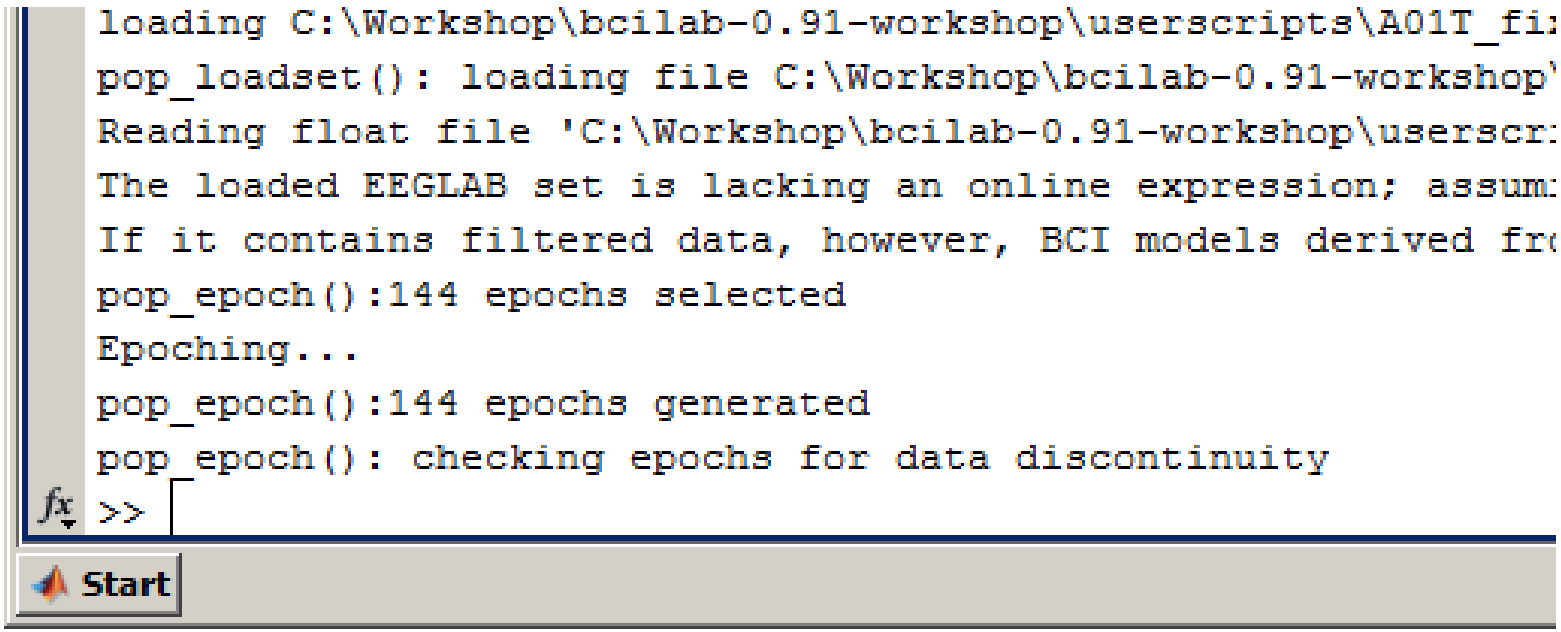
Now Hitting the Data…



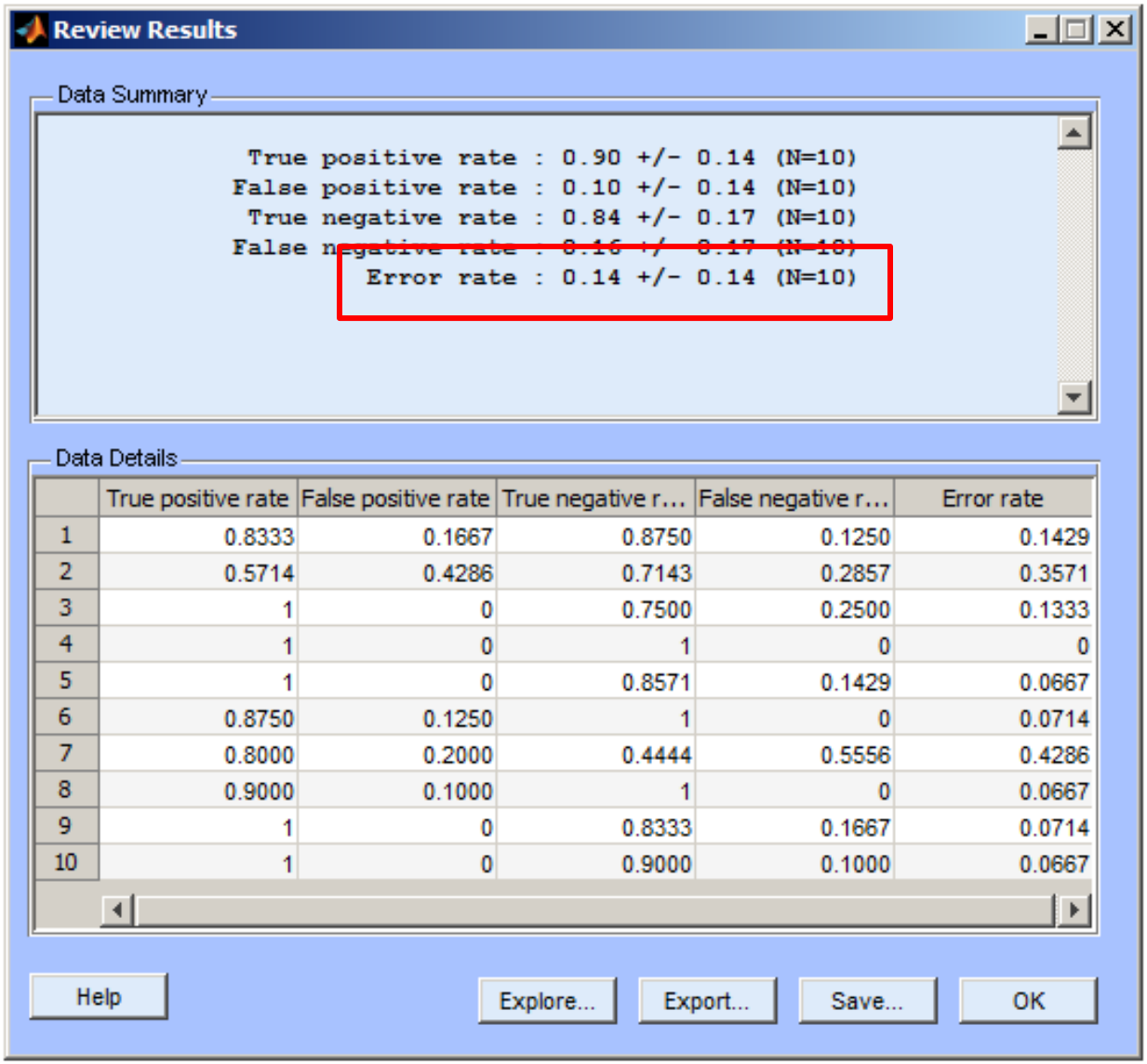
Now Hitting the Data…



Watching the Computation…

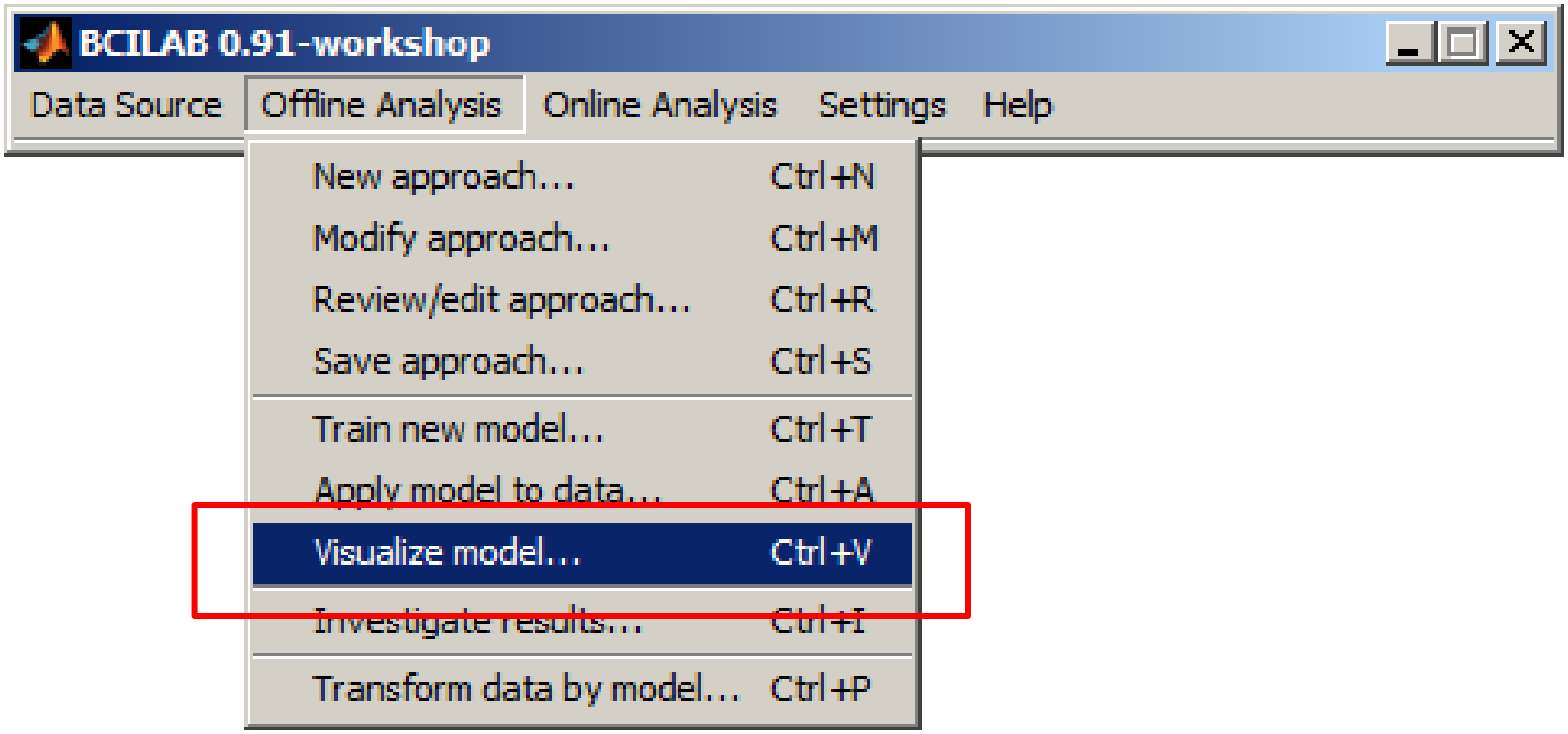


Cross-validation Results:

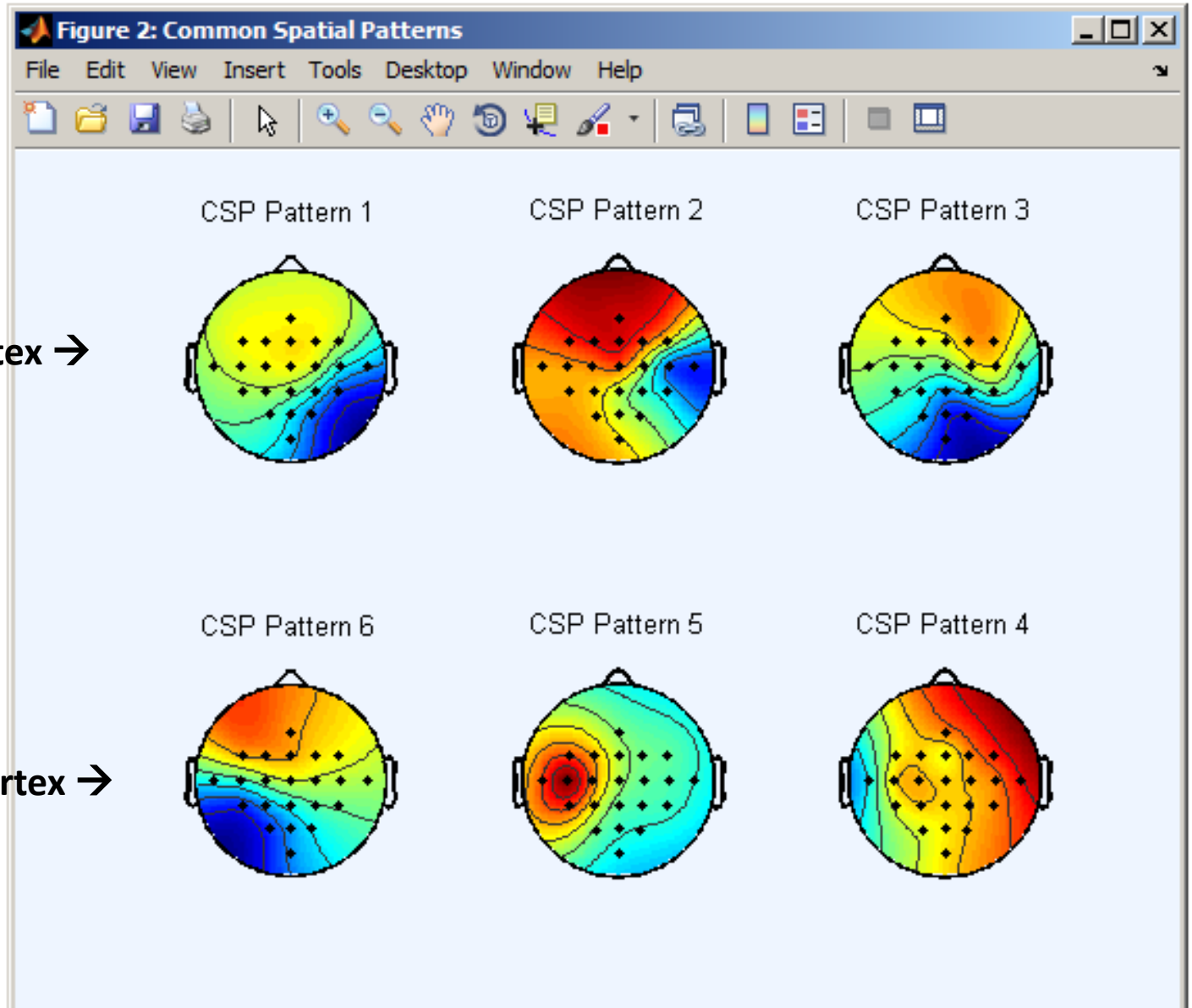


	True positive rate	False positive rate	True negative r...	False negative r...	Error rate
1	0.8333	0.1667	0.8750	0.1250	0.1429
2	0.5714	0.4286	0.7143	0.2857	0.3571
3	1	0	0.7500	0.2500	0.1333
4	1	0	1	0	0
5	1	0	0.8571	0.1429	0.0667
6	0.8750	0.1250	1	0	0.0714
7	0.8000	0.2000	0.4444	0.5556	0.4286
8	0.9000	0.1000	1	0	0.0667
9	1	0	0.8333	0.1667	0.0714
10	1	0	0.9000	0.1000	0.0667

Visualizing a Trained Model



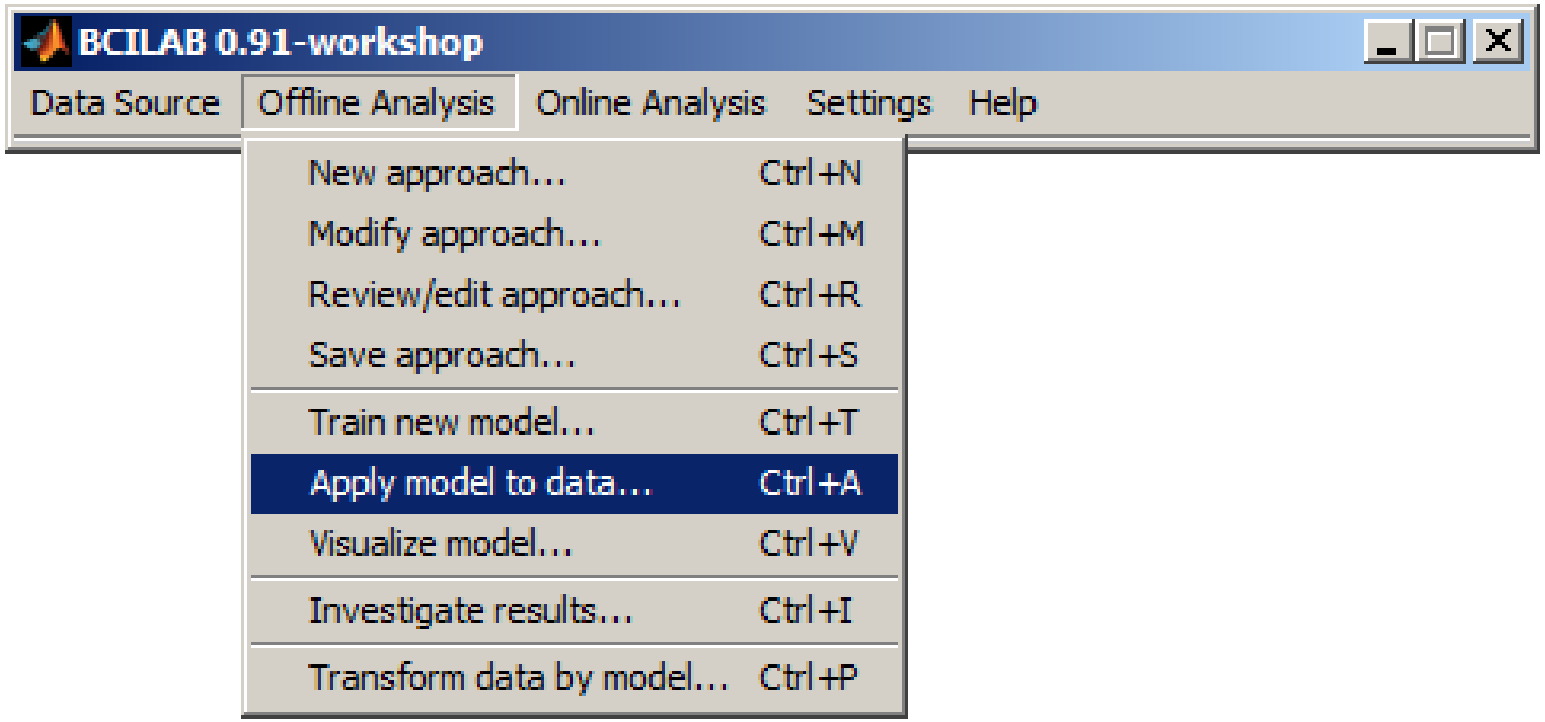
Visualizing a Trained Model



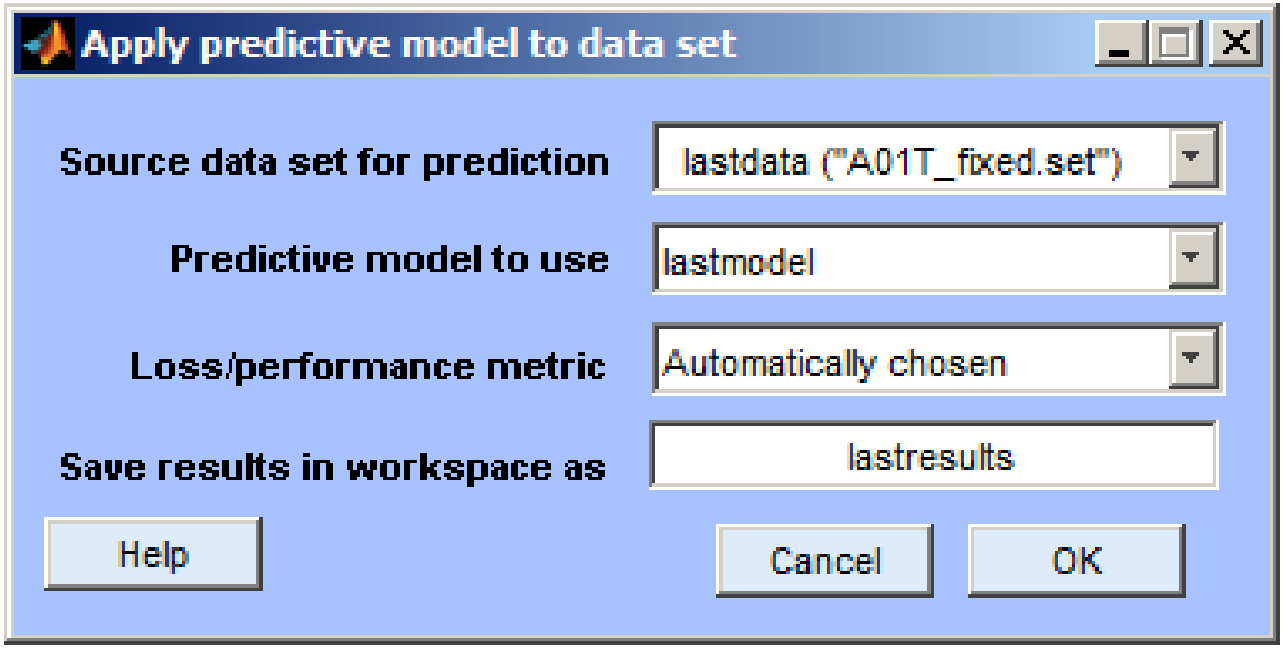
Left hand motor cortex →

Right hand motor cortex →

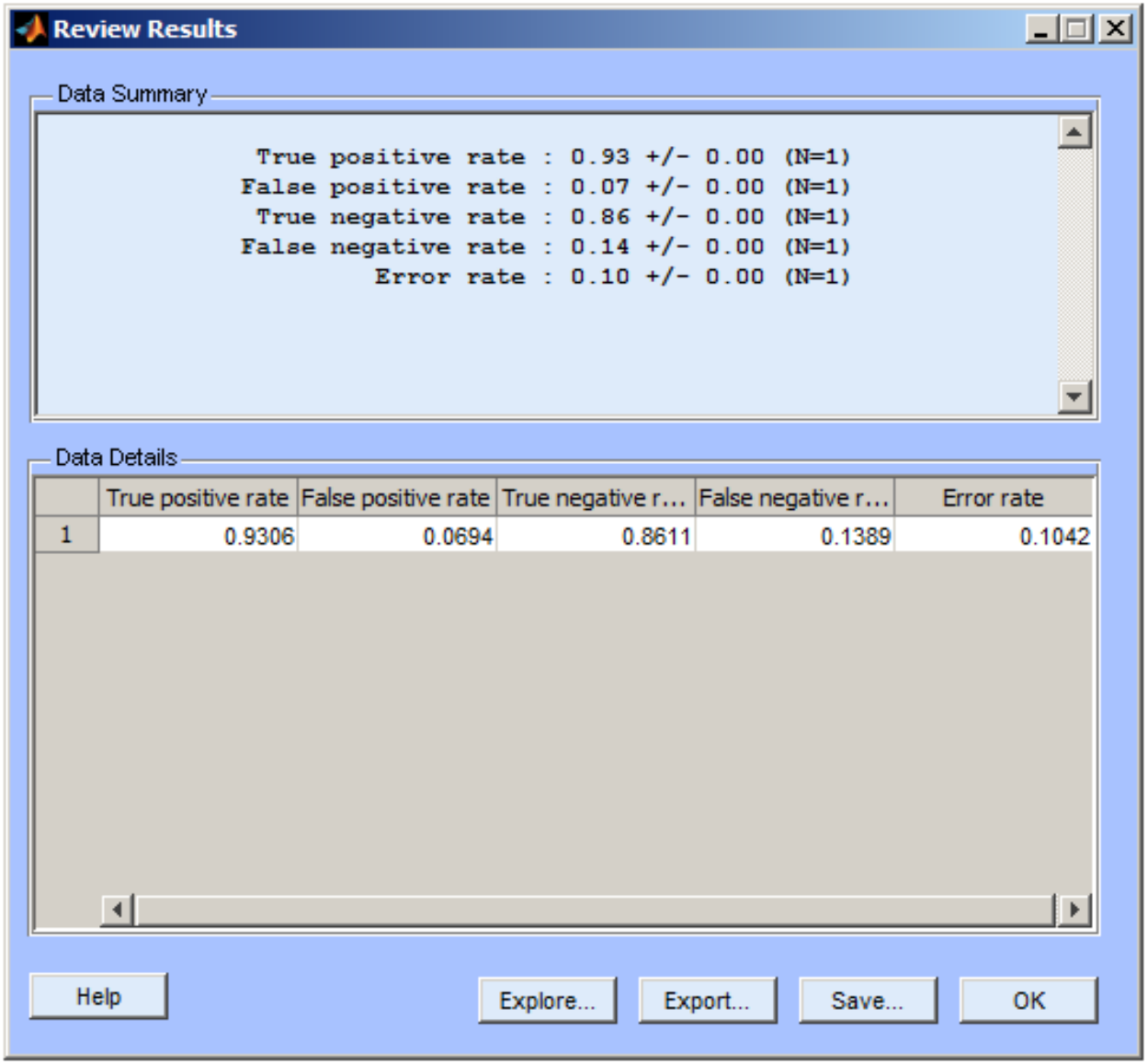
Applying a Model to New Data



Applying a Model to New Data

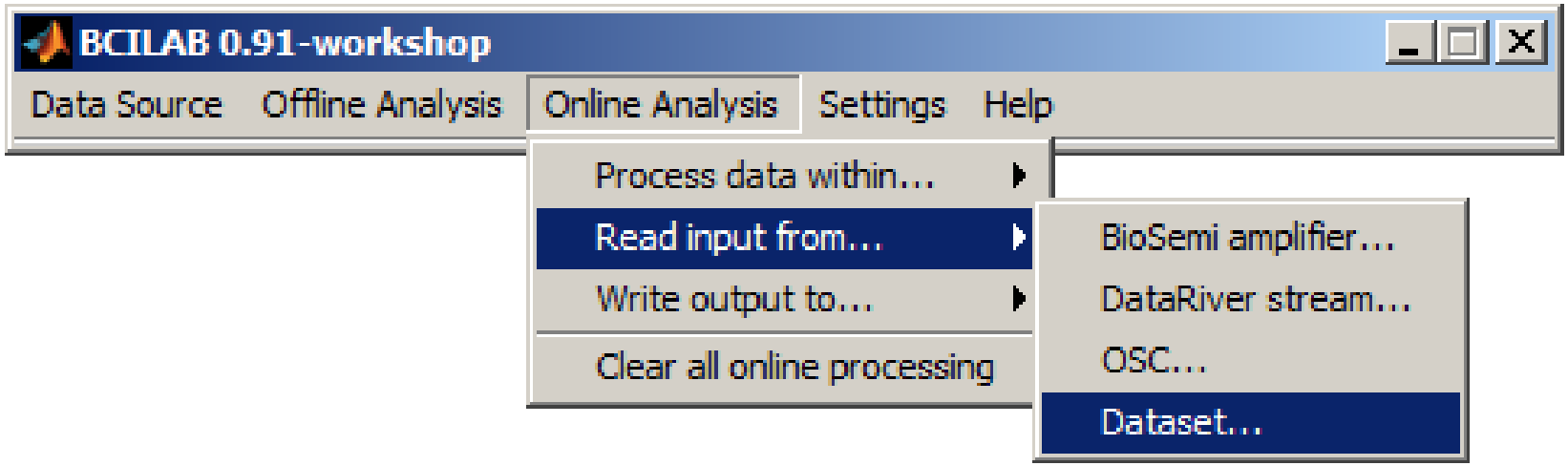


Results: Training-Set Error



Online Processing

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

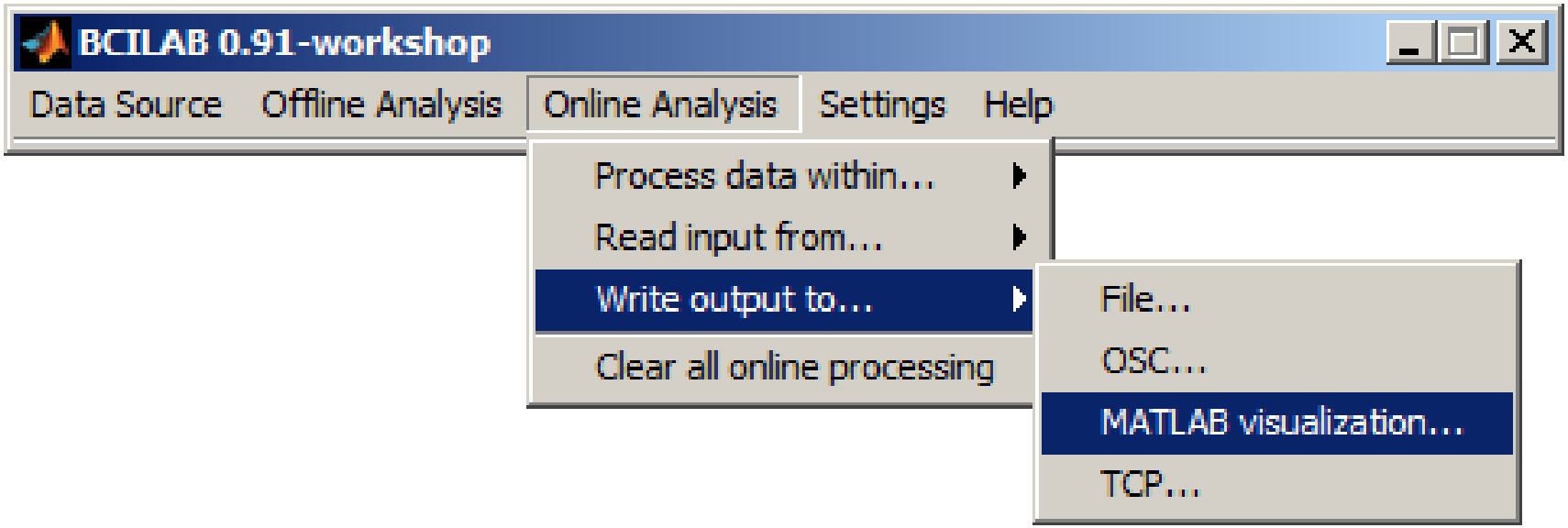


Online Processing

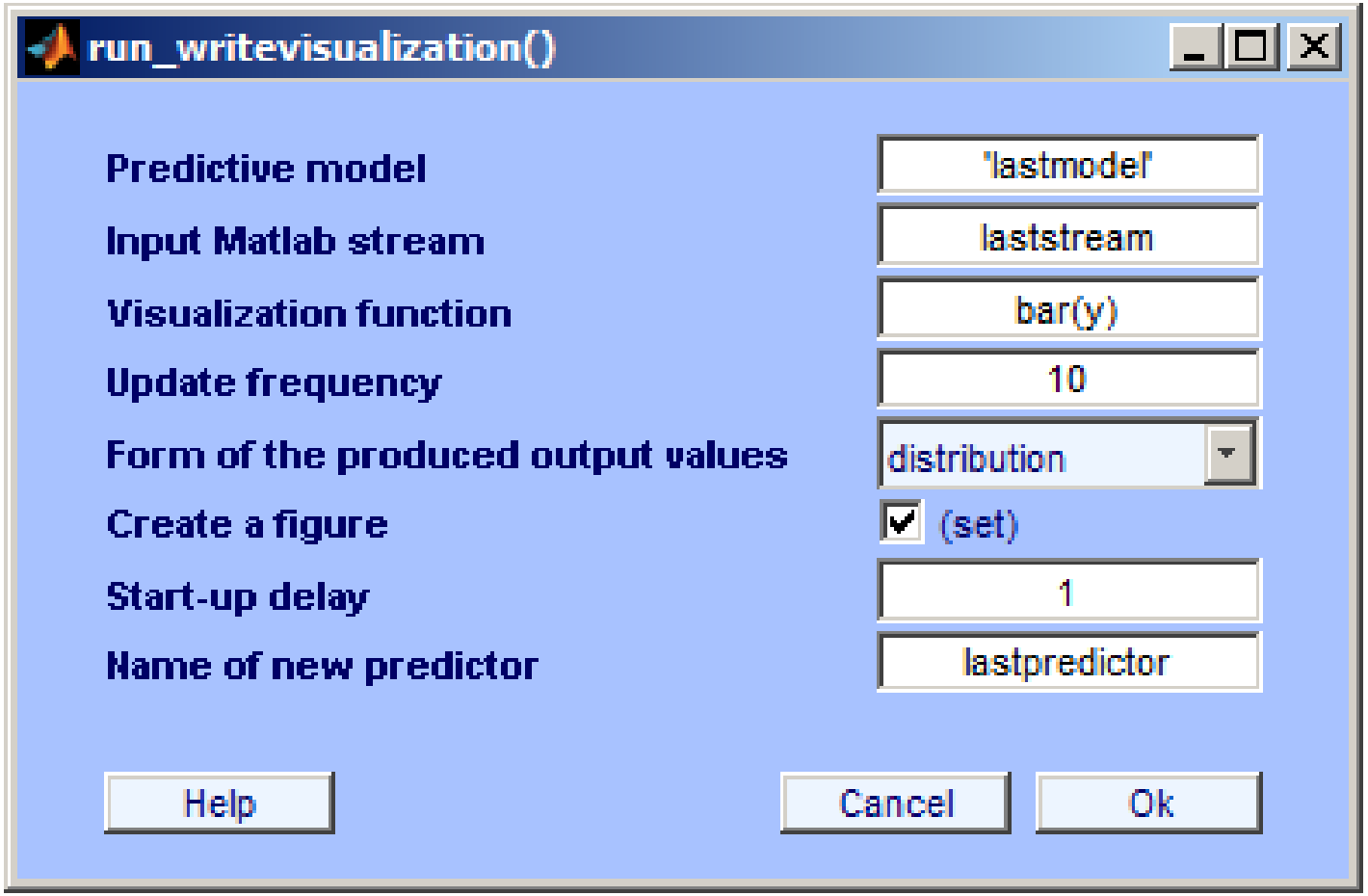
- For lack of EEG hardware, play back a data set in real time:

Online Processing

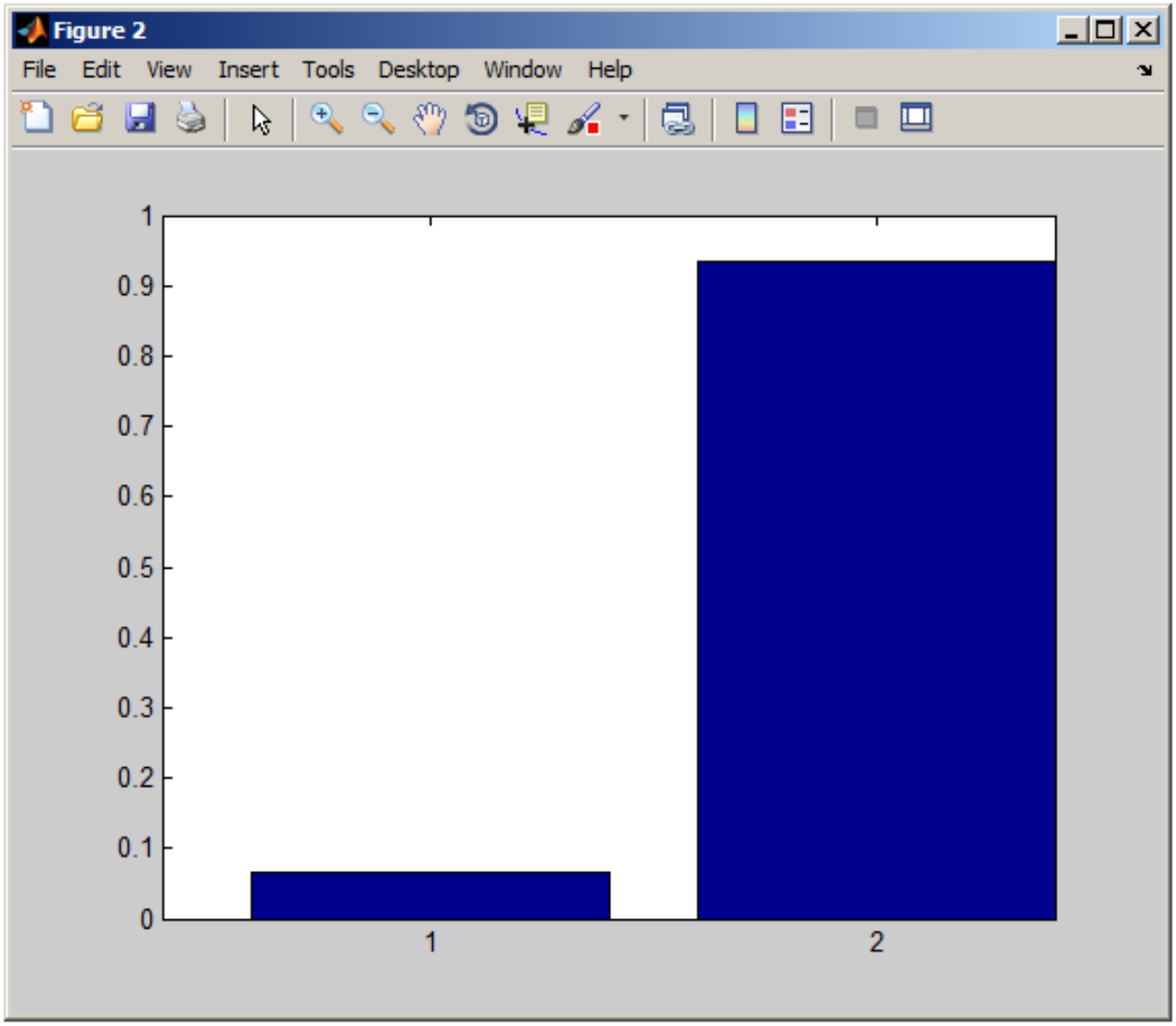
- Choosing a destination for outputs:



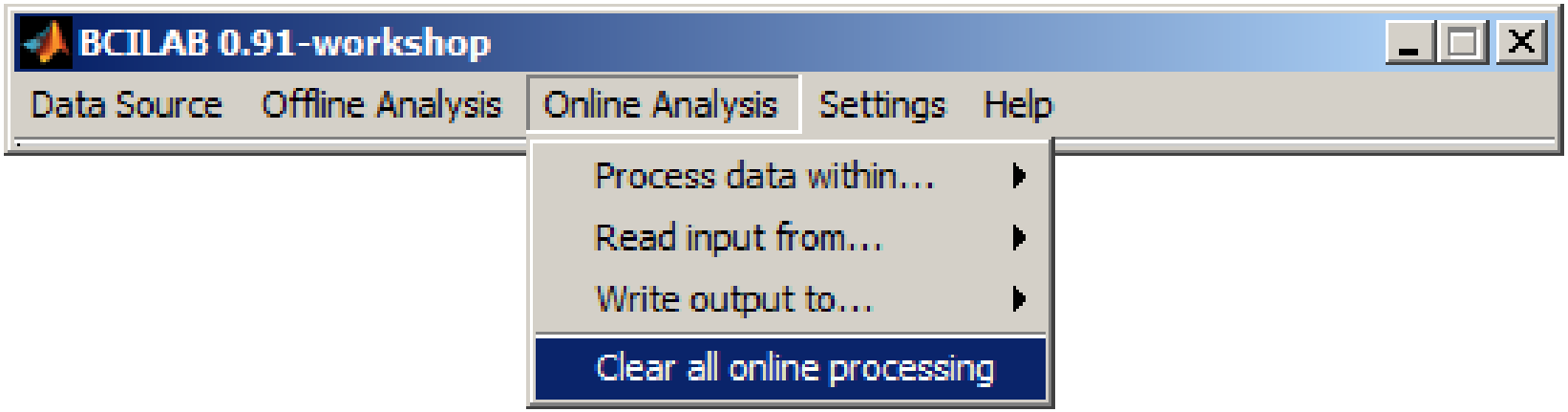
Setting up a Visualization



Real-time Class Probabilities



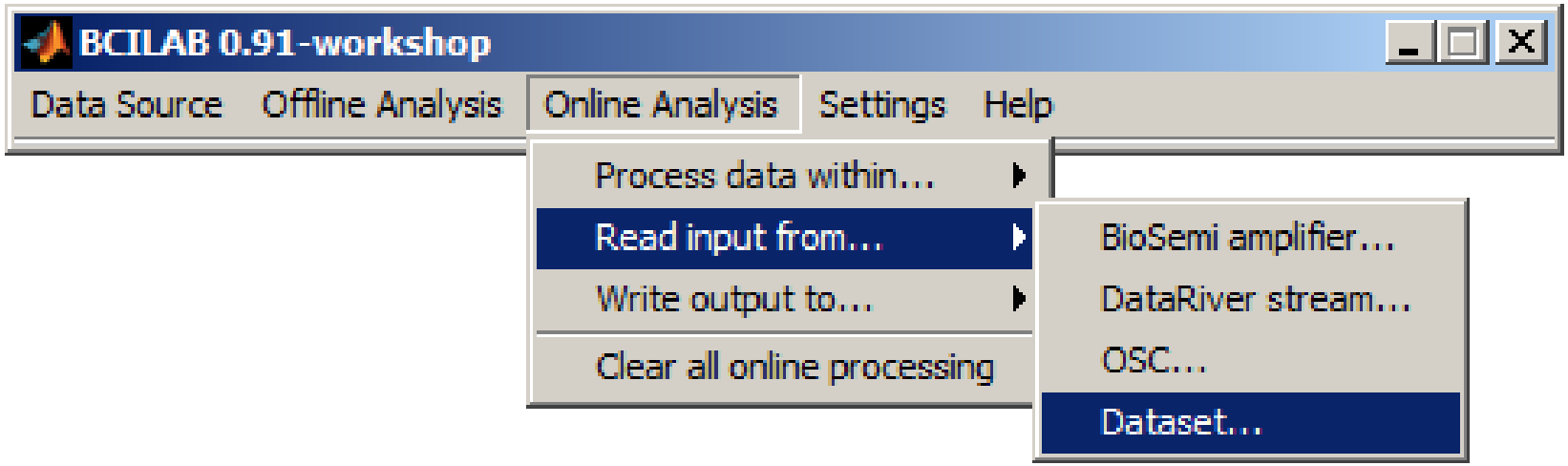
Clearing the Online Processes



Fancier Visualization

Online Processing

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

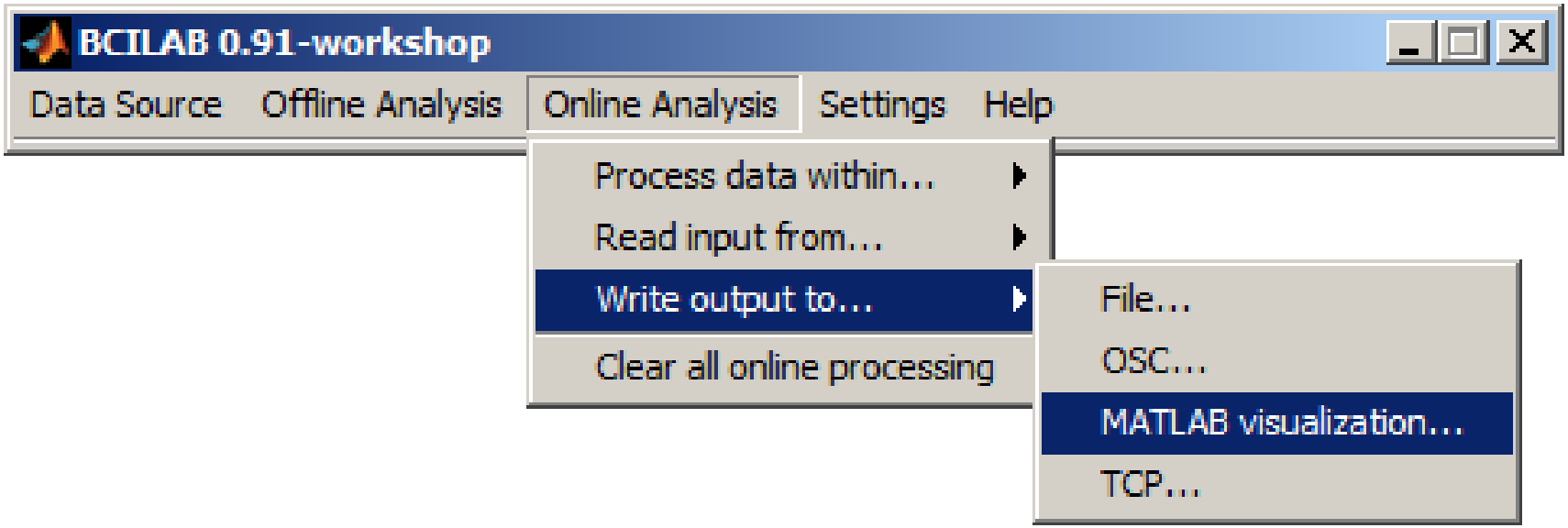


Online Processing

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

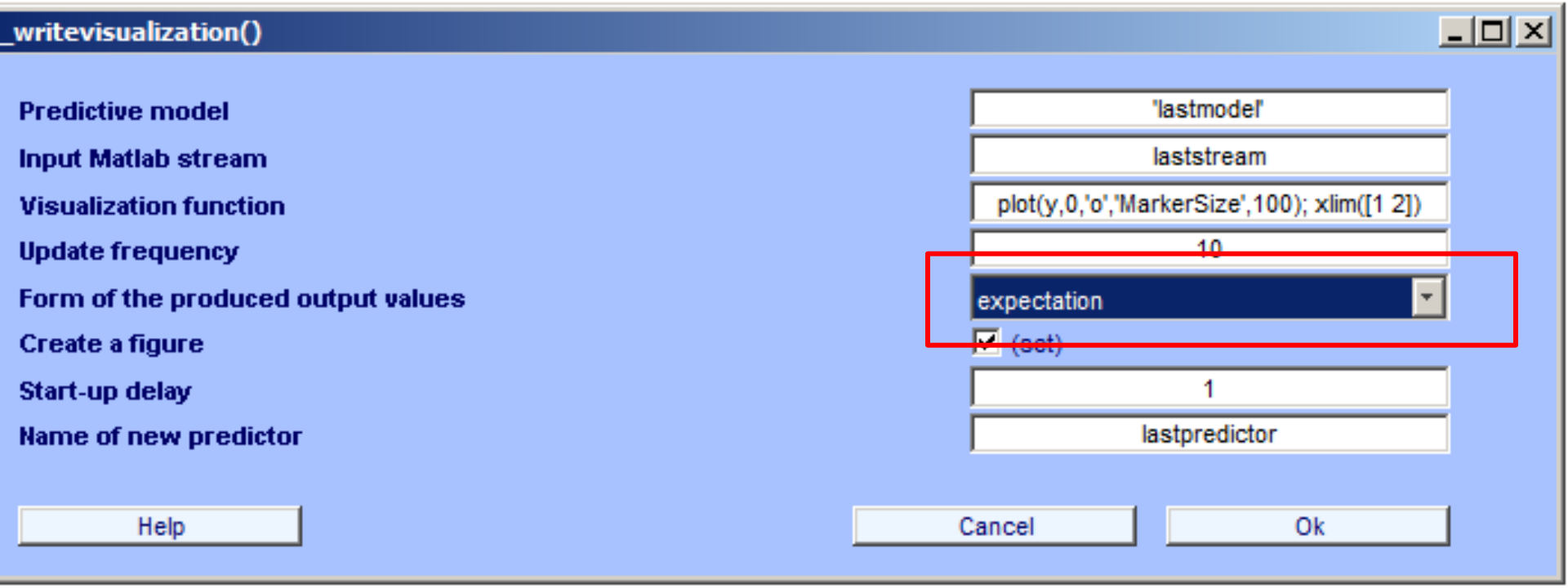
Online Processing

- Choosing a destination for outputs:



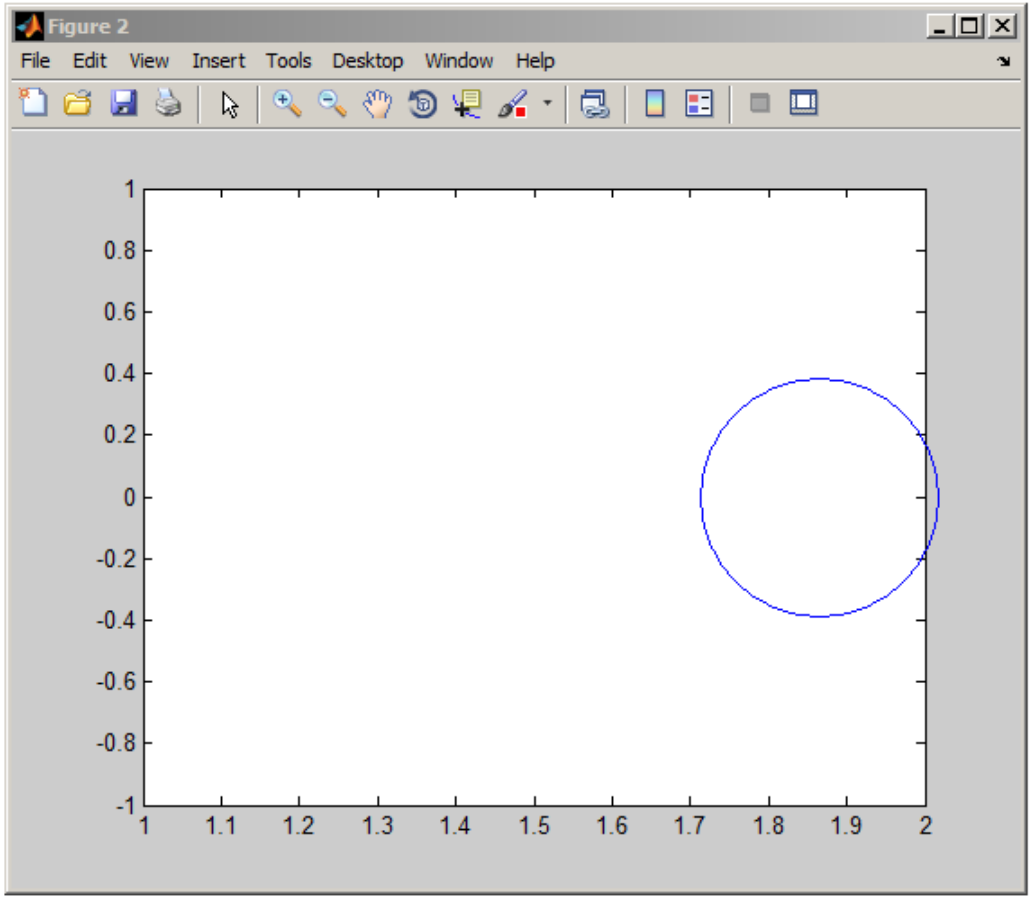
Setting up a Visualization

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

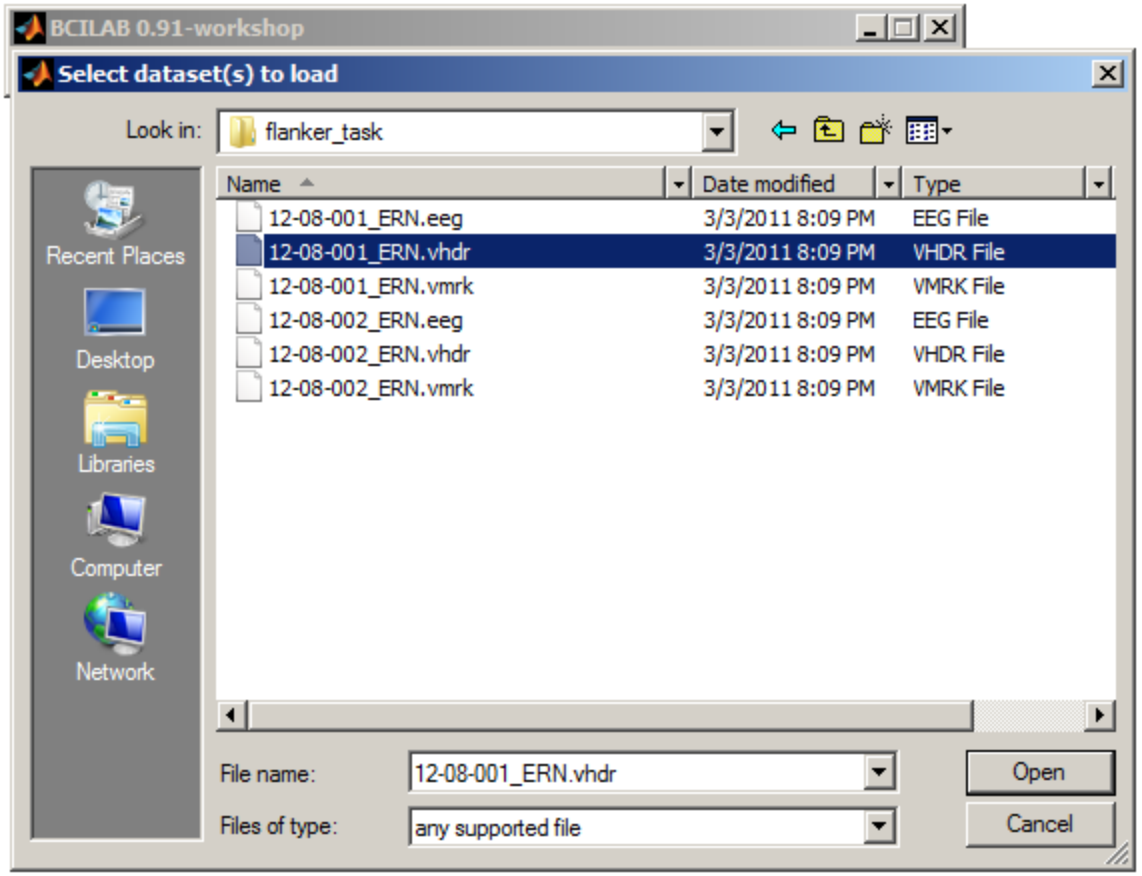


Real-time Class Probabilities

- 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]