

EEG classification and cross-validation using the BCILAB toolbox: practicum

Christian A. Kothe
SCCN/UCSD

Download the latest toolbox version from:

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

System requirements

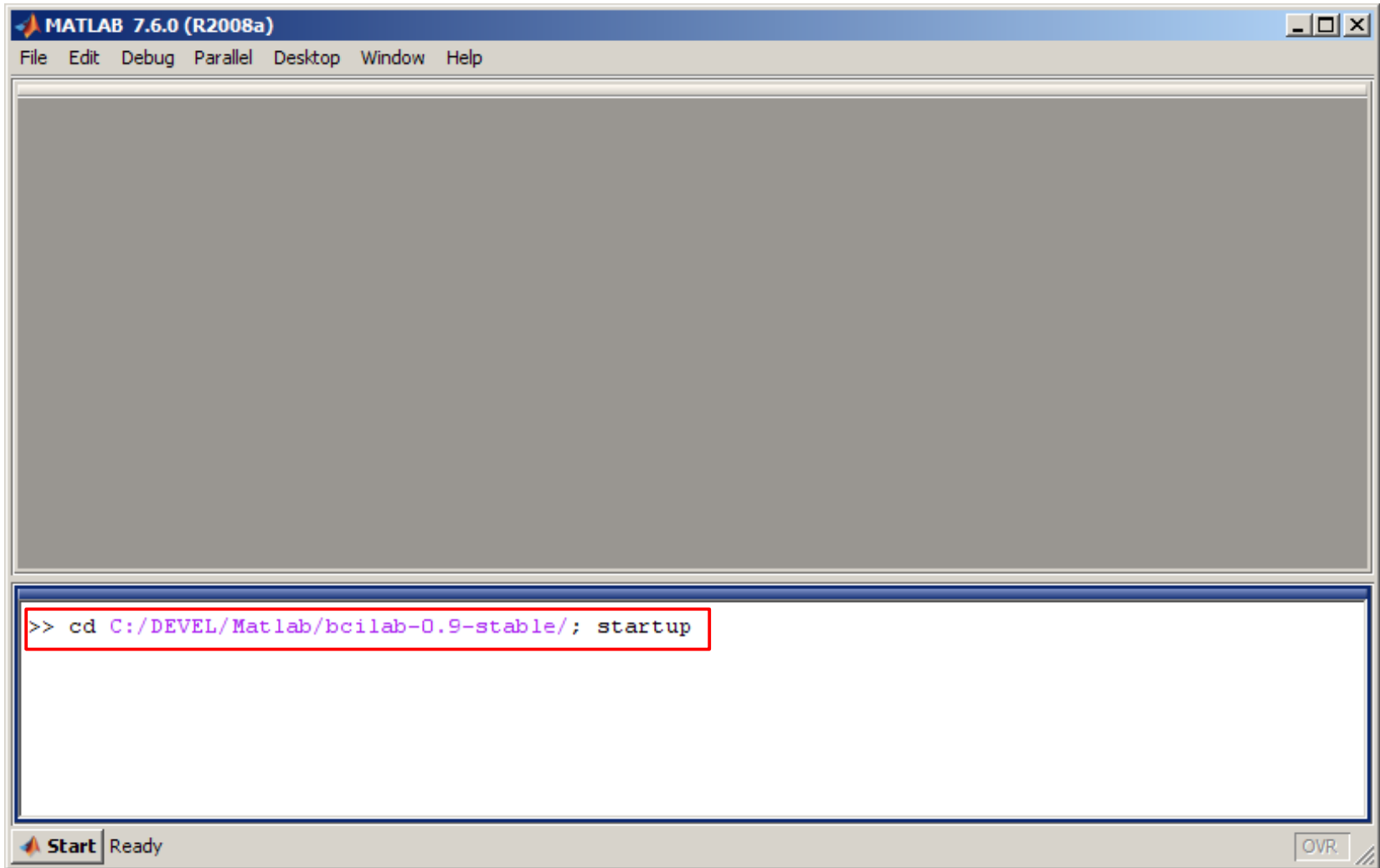
- MATLAB 2008a+ (scripts will run on 7.1+ (2005), but not this version)
- 1GB+ RAM (better: 2GB+)
- Windows, Linux, or Mac
- For smooth workshop: **No** toolboxes in MATLAB path other than Mathworks toolboxes (or EEGLAB)
- To use certain additional features (not covered today):
Signal Processing Toolbox, 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)

Installation

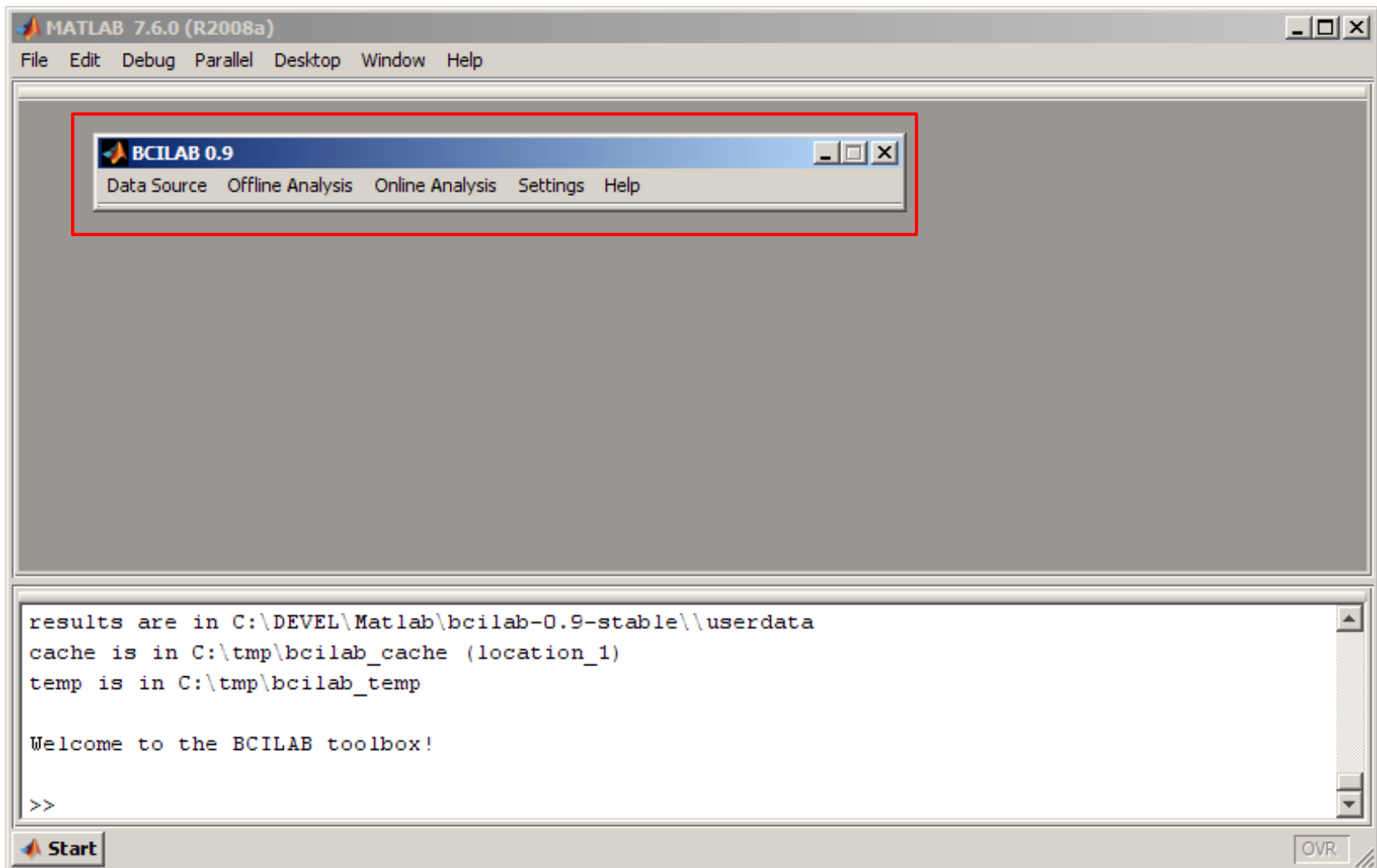
- Extract the .zip file (it contains one folder)
- Open MATLAB, type

```
cd /your/directory/bcilab-0.9-stable  
startup
```

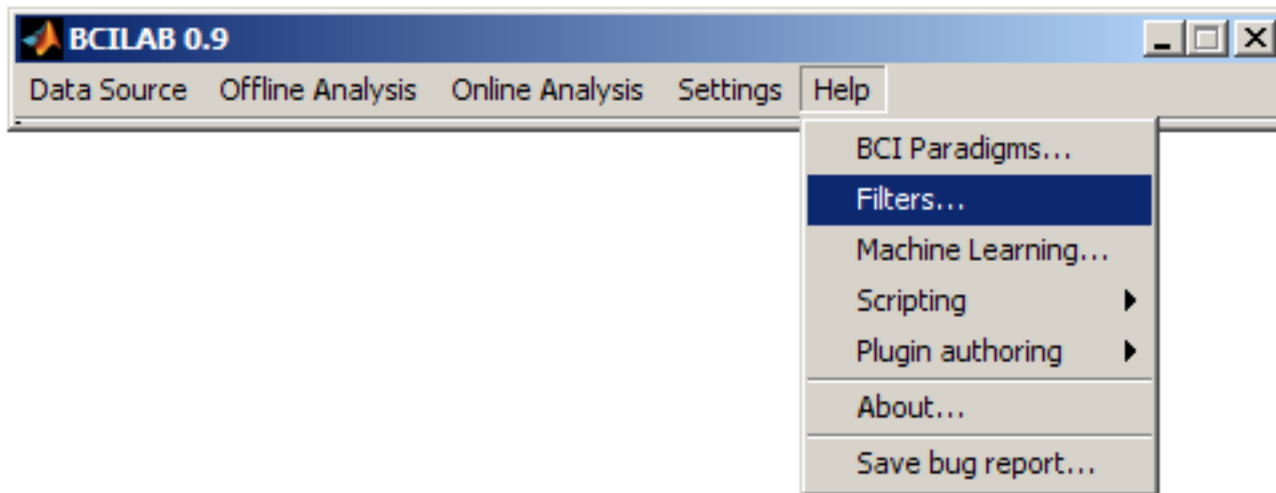
Startup



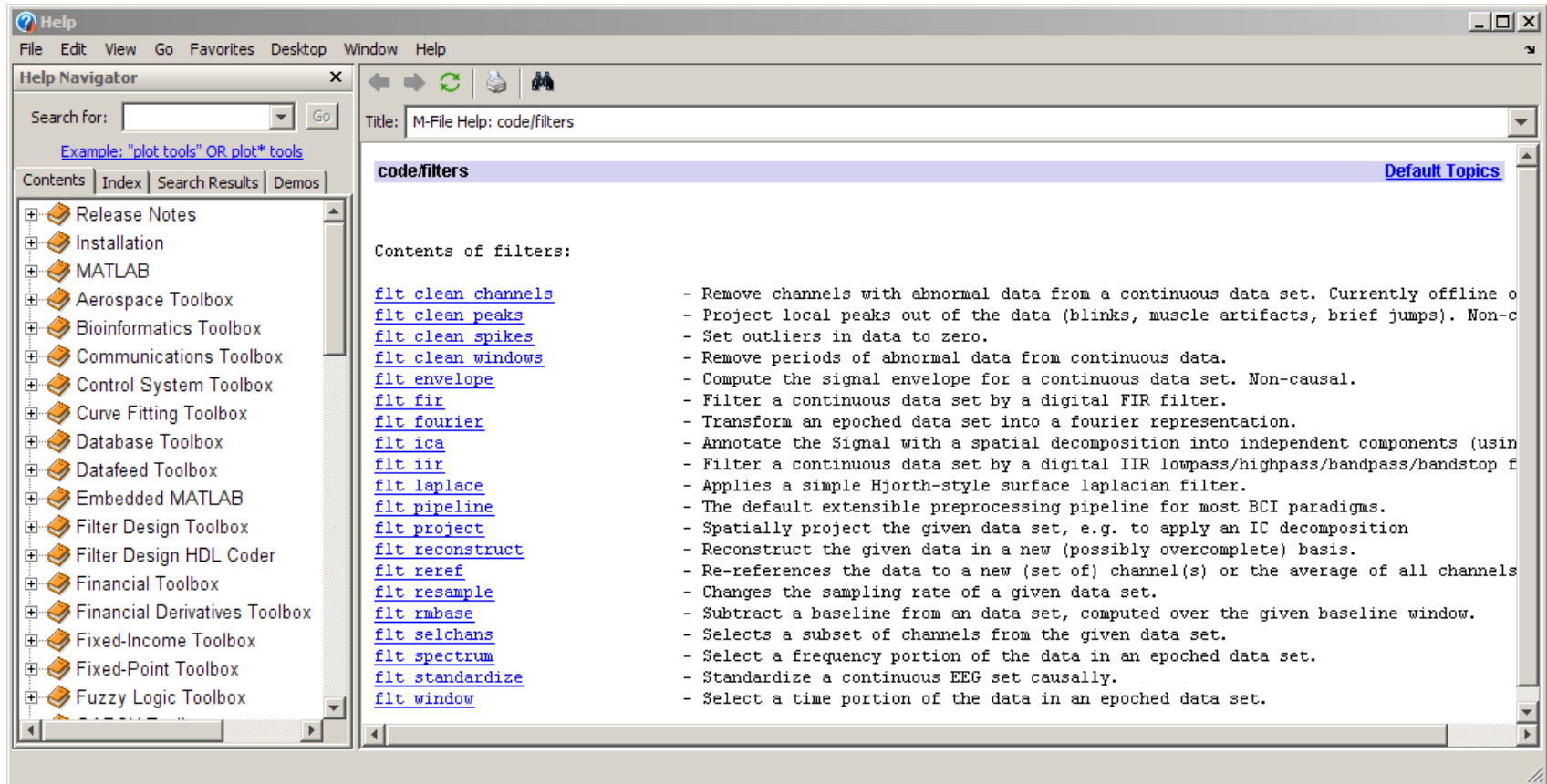
Toolbox GUI



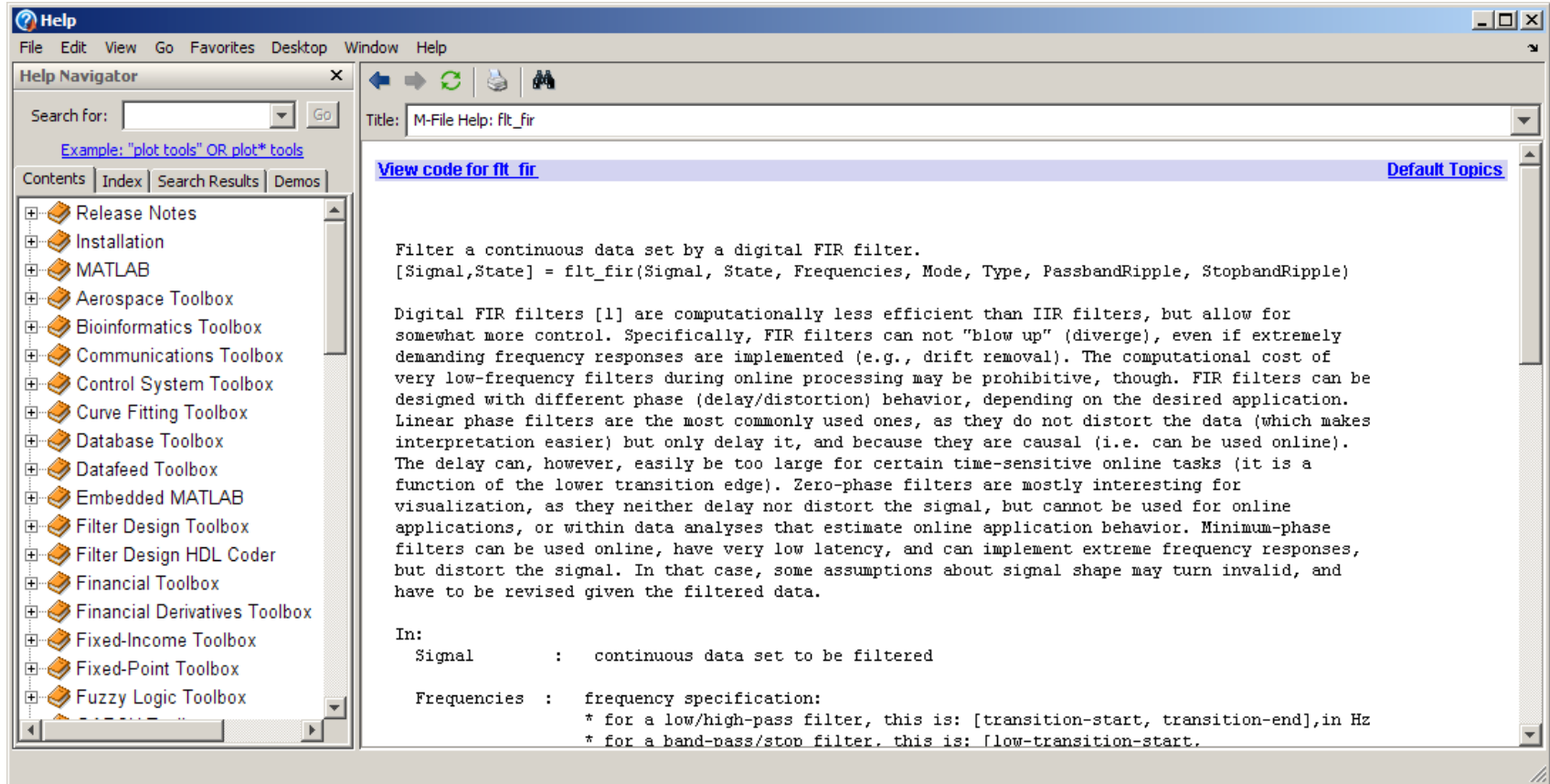
Getting help (if needed)



Getting help (if needed)



Getting help (if needed)



Use case 1

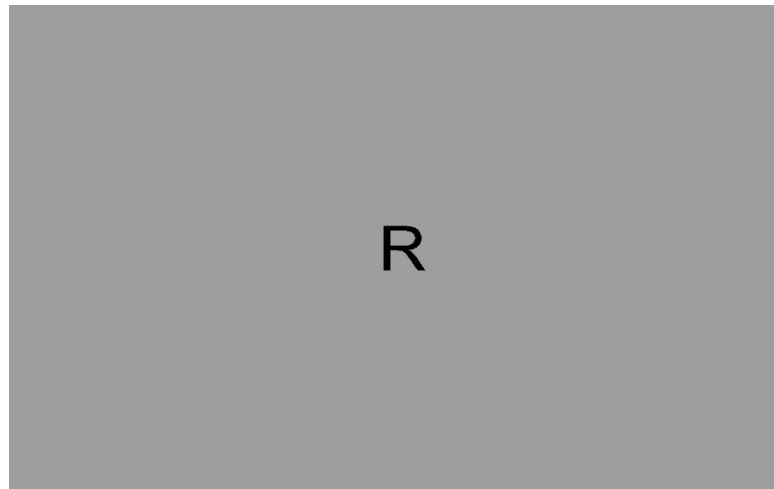
- You just recorded pilot data for some new study
- The idea is to try to estimate a certain aspect of cognitive state
- The question is what method works best, and what accuracies can be achieved

Use case 1

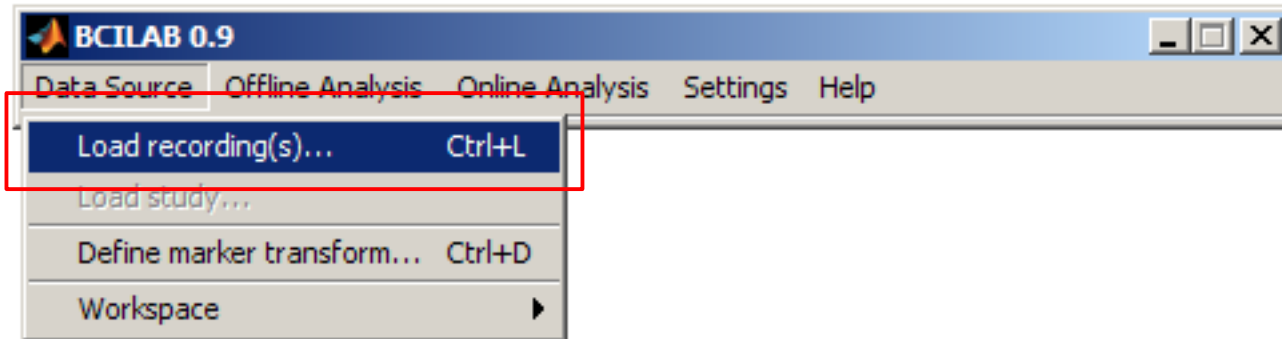
- Scenario: Subject is instructed to imagine a hand movement, either **left** hand or **right** hand (standard BCI case)
- Task: Estimate, from raw data, which hand movement was imagined
- Experimental data: EEG, 32 channels, 2 sessions (each ~30 min.), 2 sub-blocks per session with intermittent pause

Experimental task

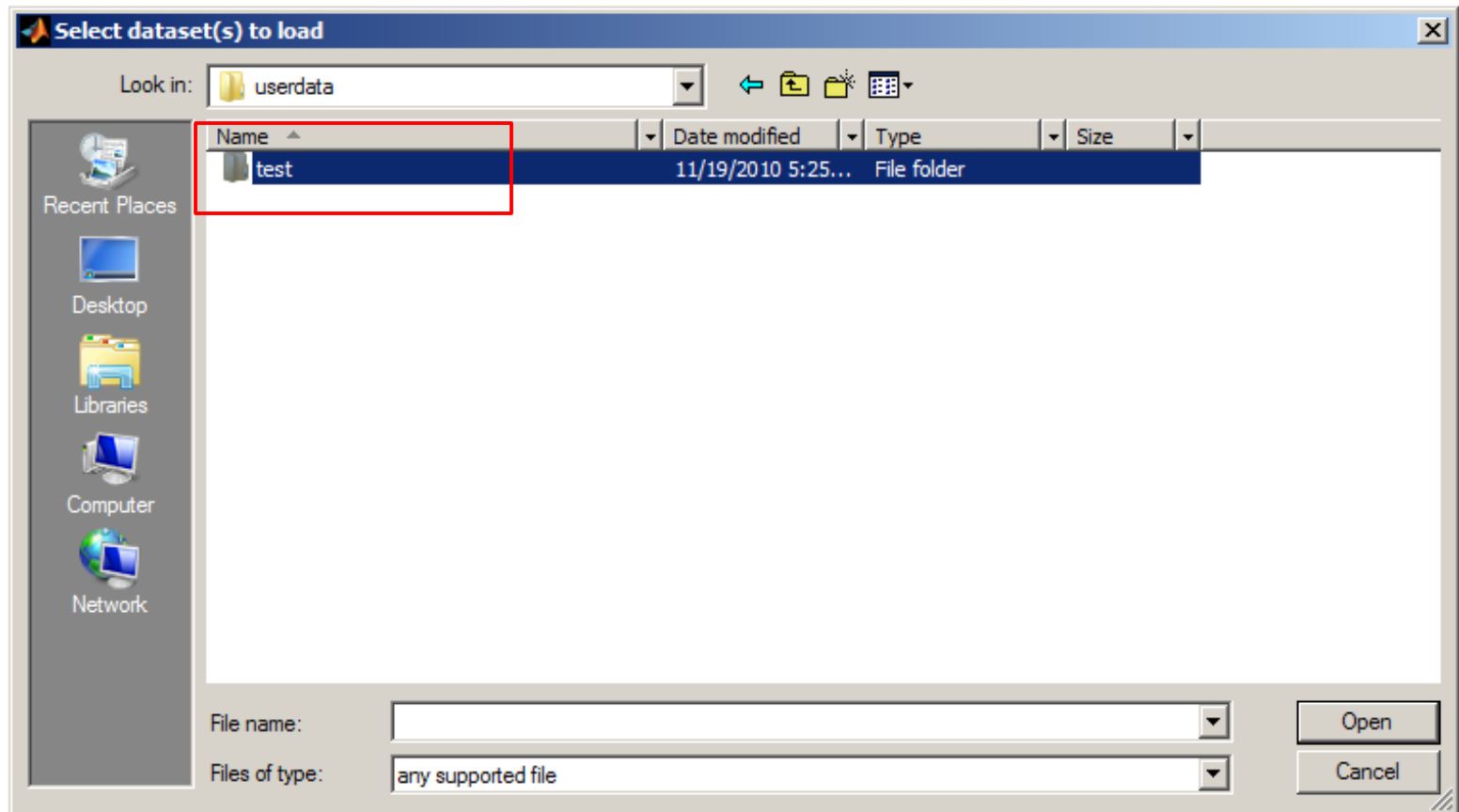
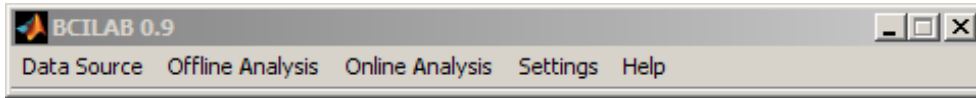
- 160 trials
- Randomized Instruction: L or R
- Displayed for 3s, followed by blank screen for 3.5s
- Sample:



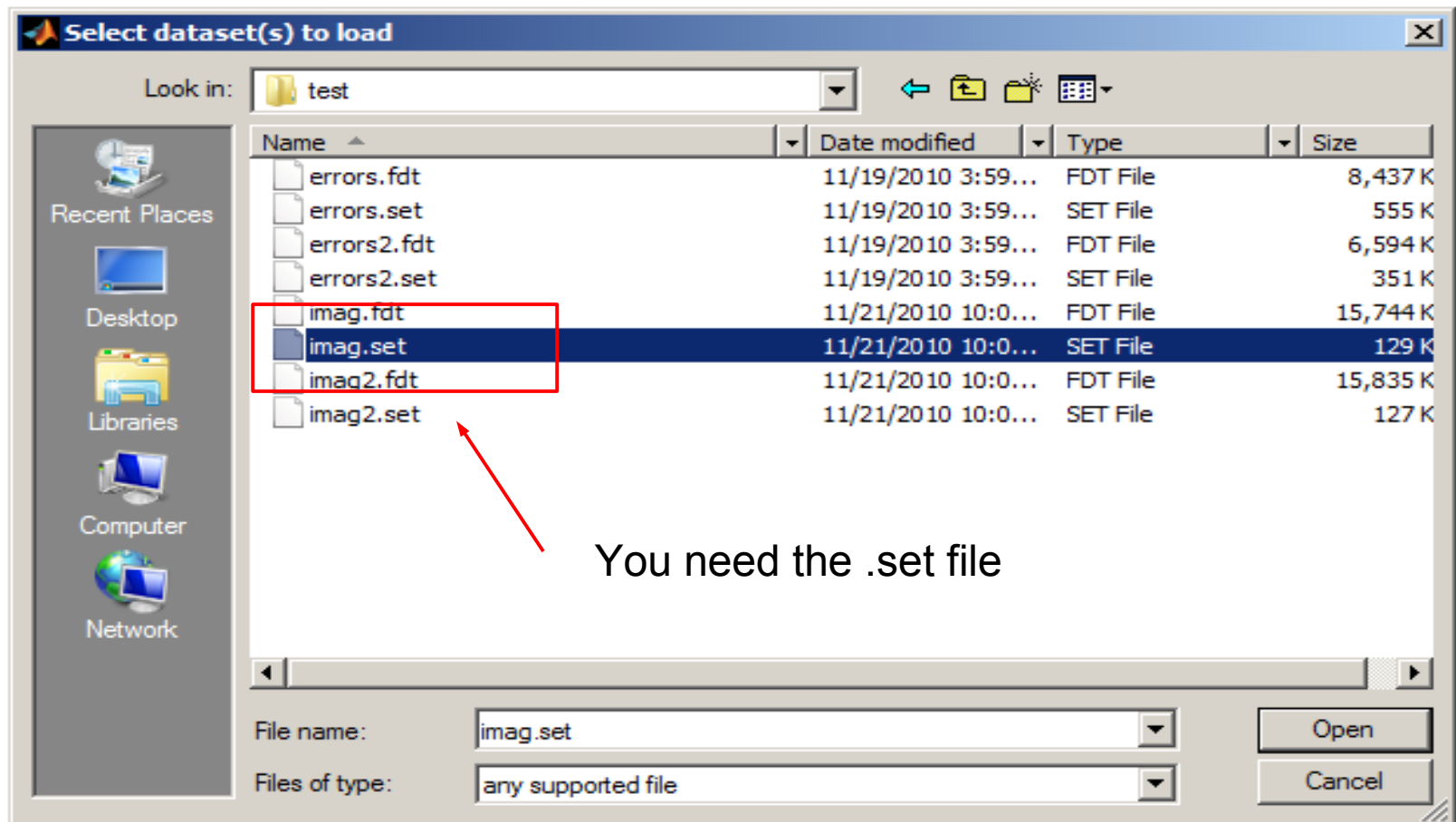
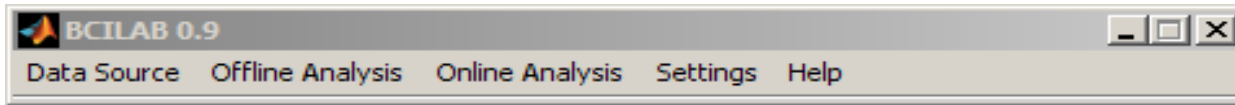
Load the data



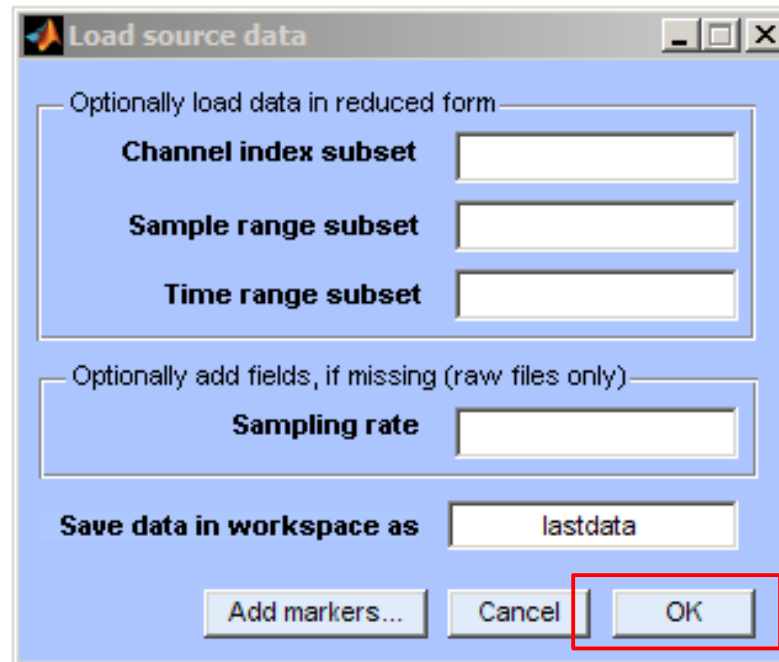
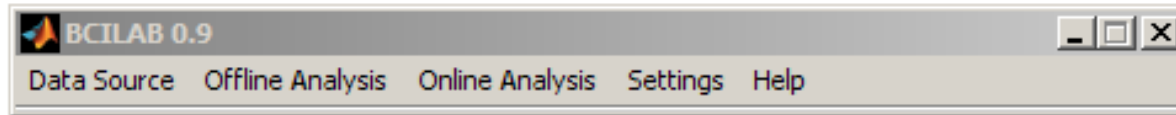
Load the data



Load the data

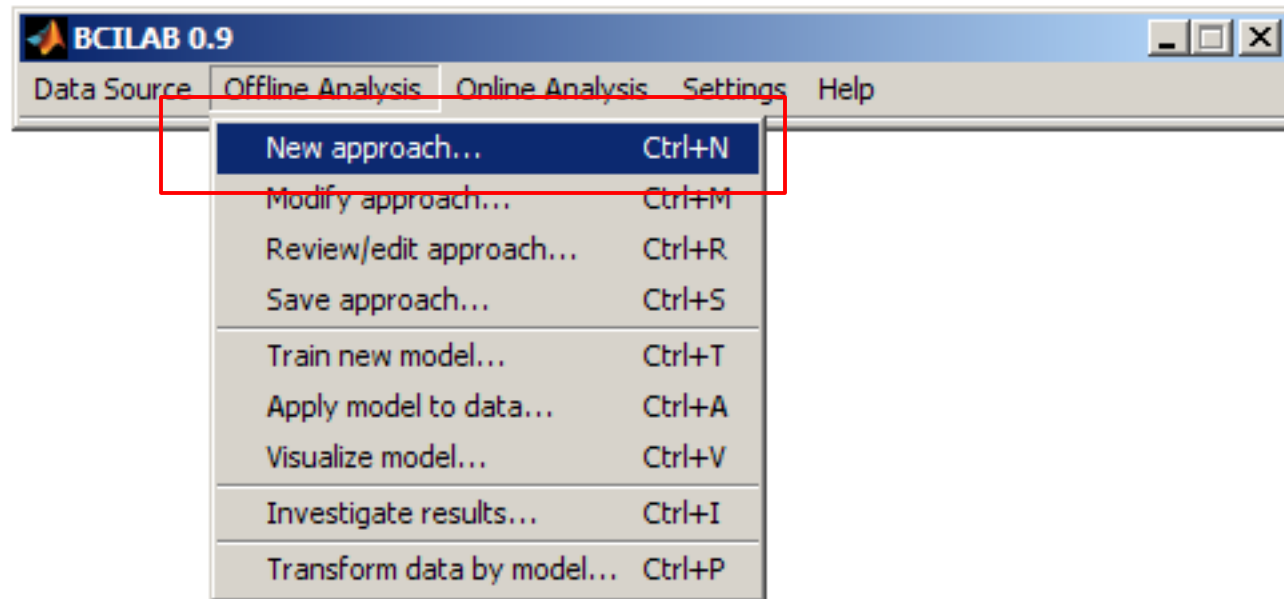


Load the data



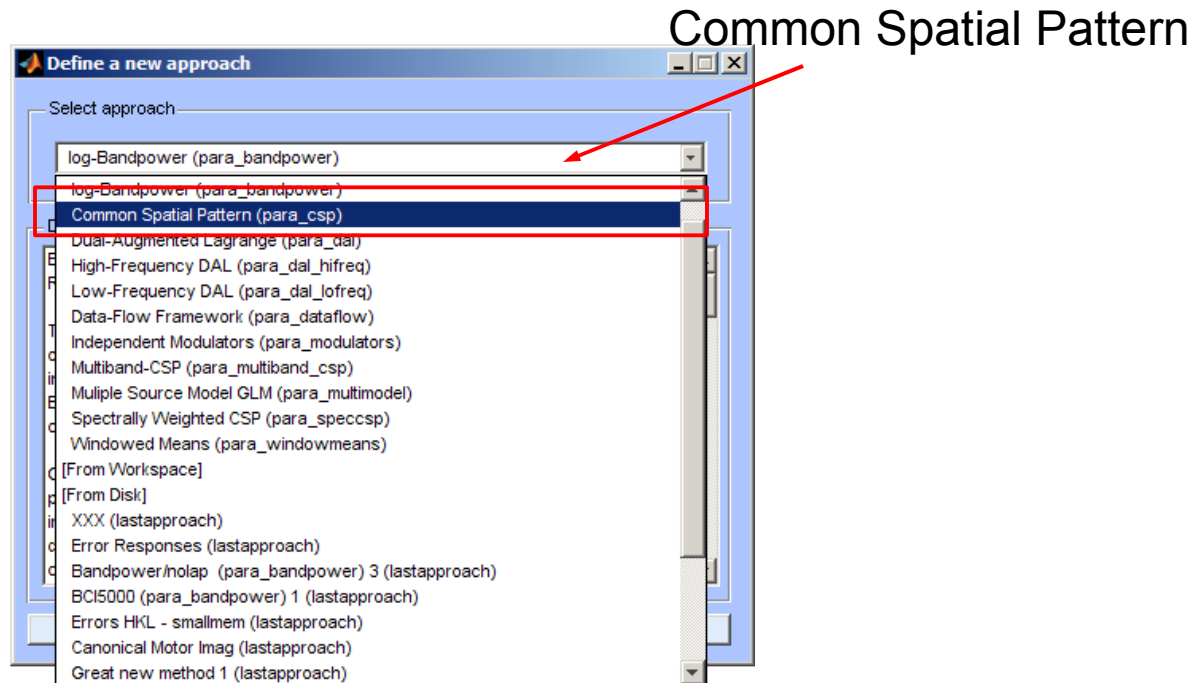
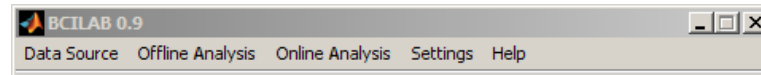
Define an approach

- An approach addresses both parts of the BCI problem: Mapping from observed signals to predictions, and learning the unknown parameters



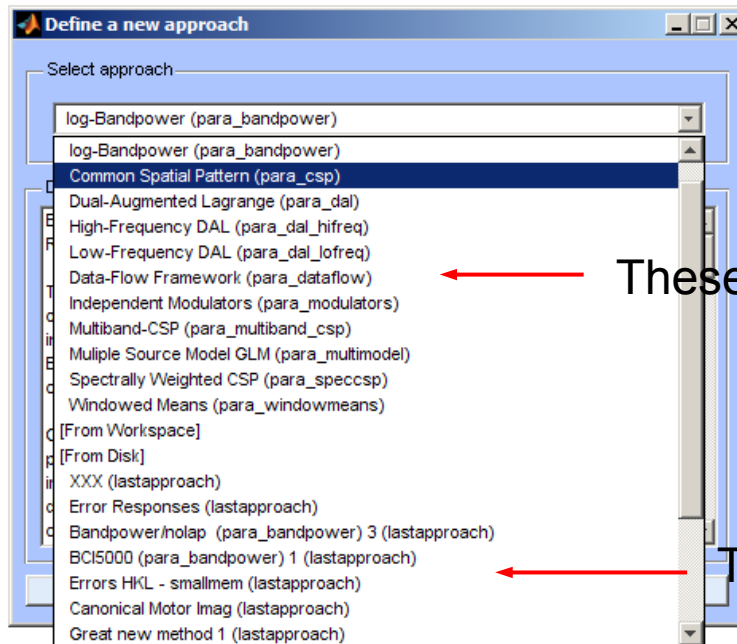
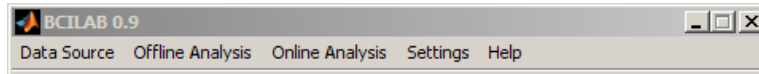
Define an approach

- You never start completely from scratch, but on the basis of what is known to work



Define an approach

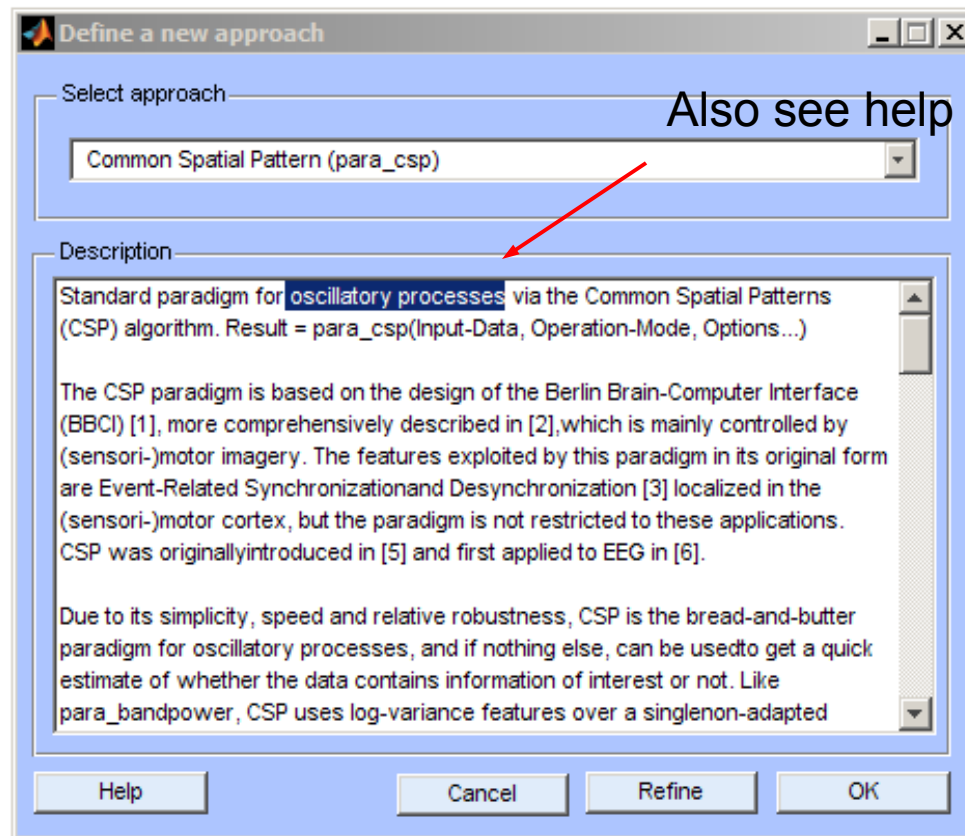
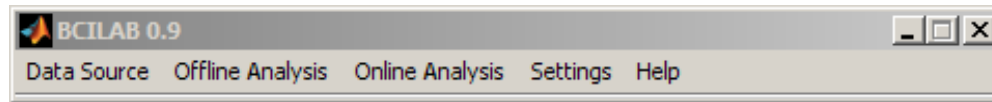
- Some of these work best for oscillatory processes, others for ERP-like features, etc.



These are pre-defined templates

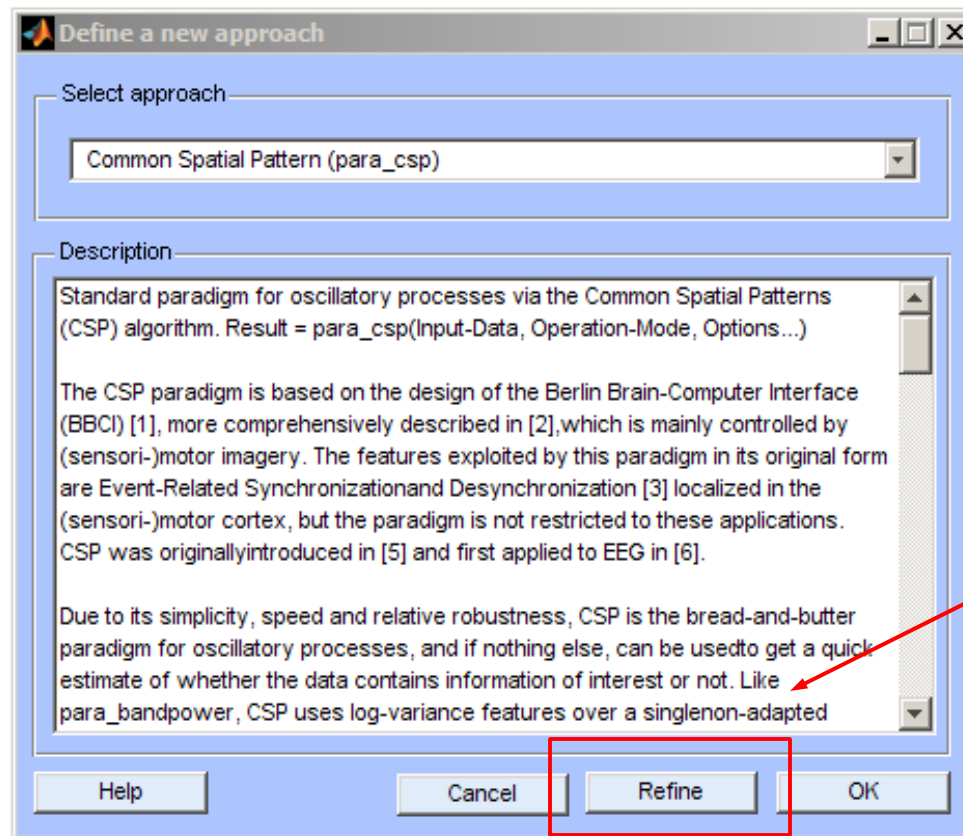
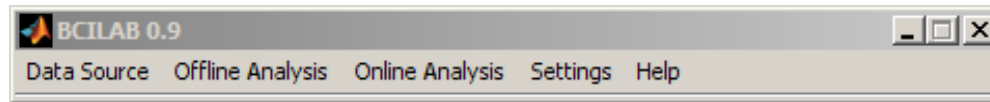
These are user-defined

Define an approach



Also see help text

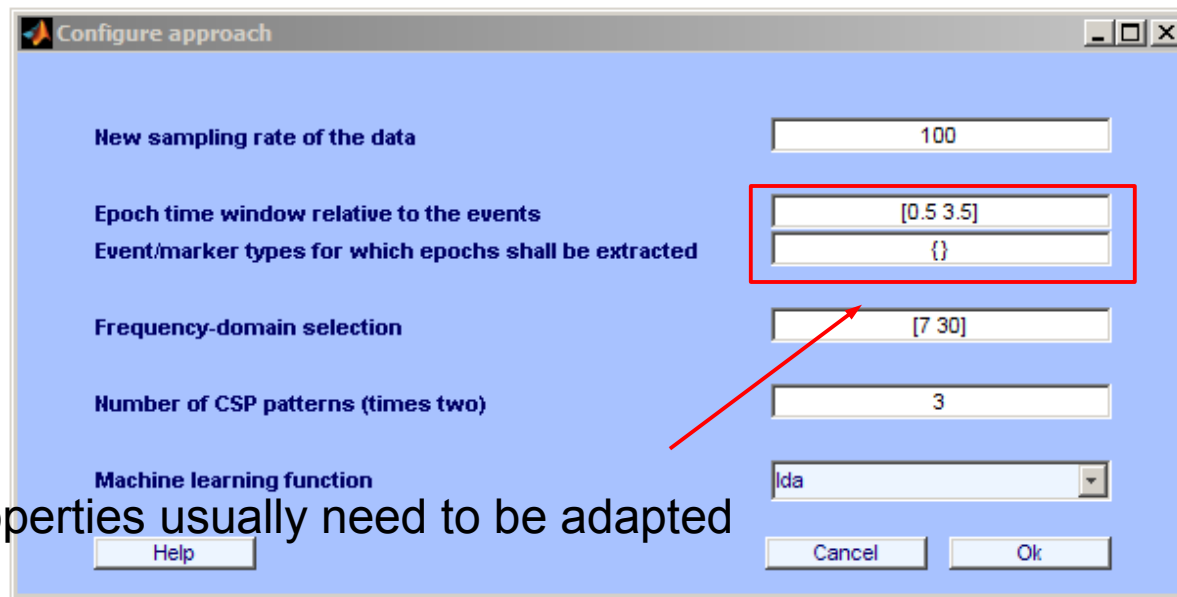
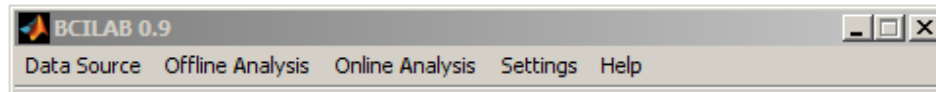
Define an approach



Adapt the template to your experiment

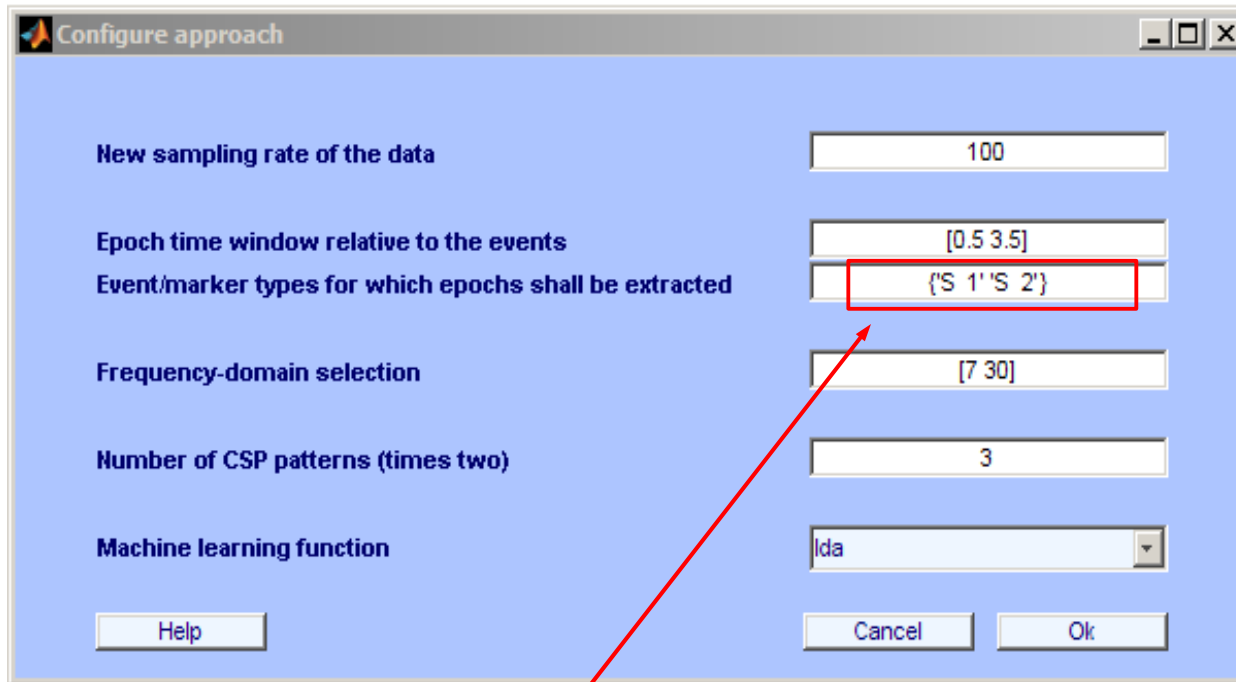
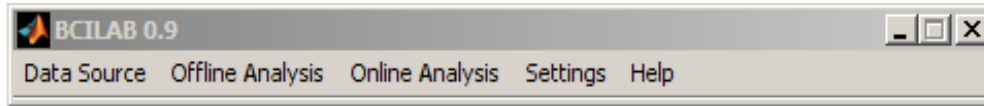
Configure approach

- Key properties can be configured in this dialog



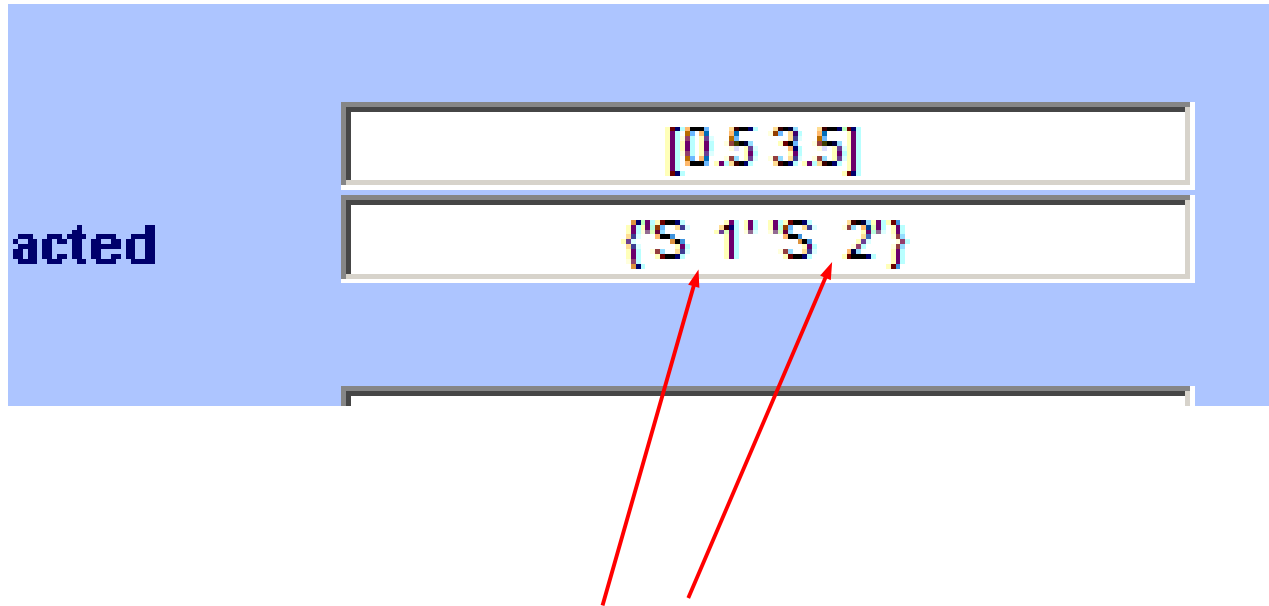
Trial epoch properties usually need to be adapted

Configure approach



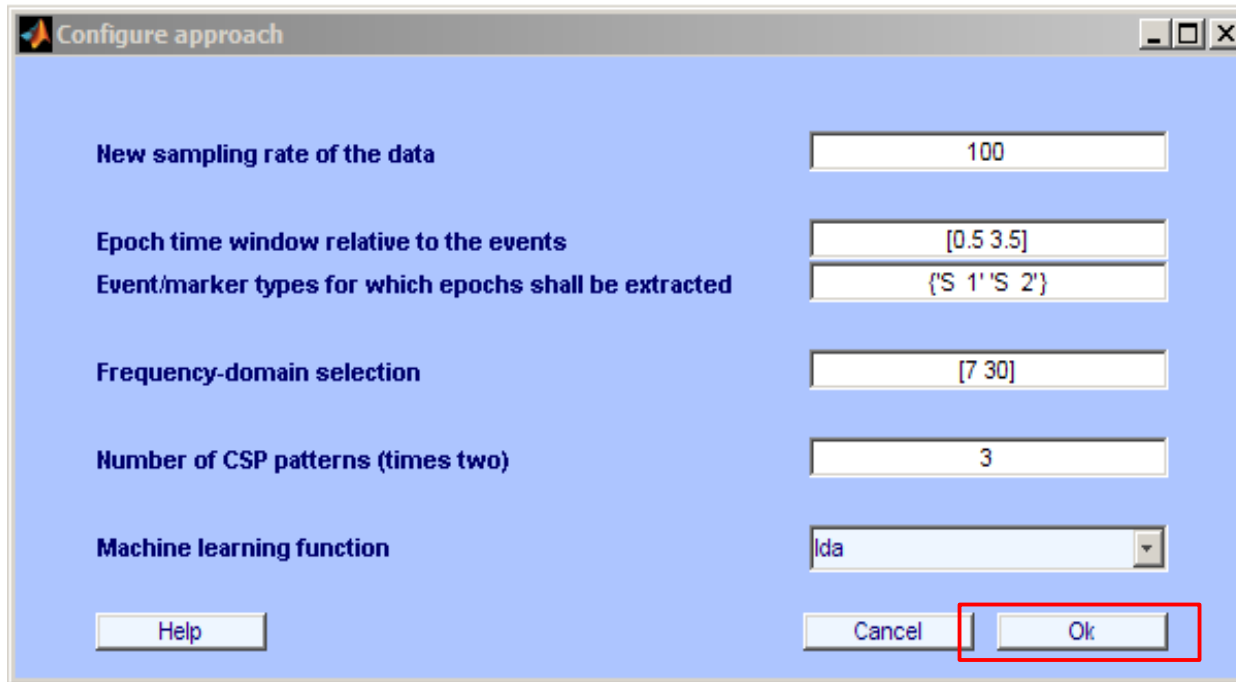
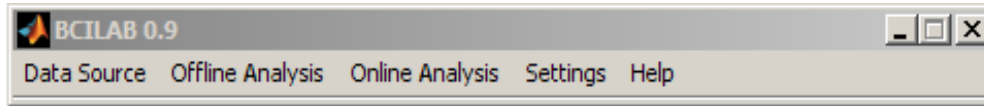
Fill in the 2 event types for this dataset; Stimulus 1 & 2, called 'S 1' and 'S 2' (Brain Products names)

Configure approach



Note the **two** spaces between the S and the number!

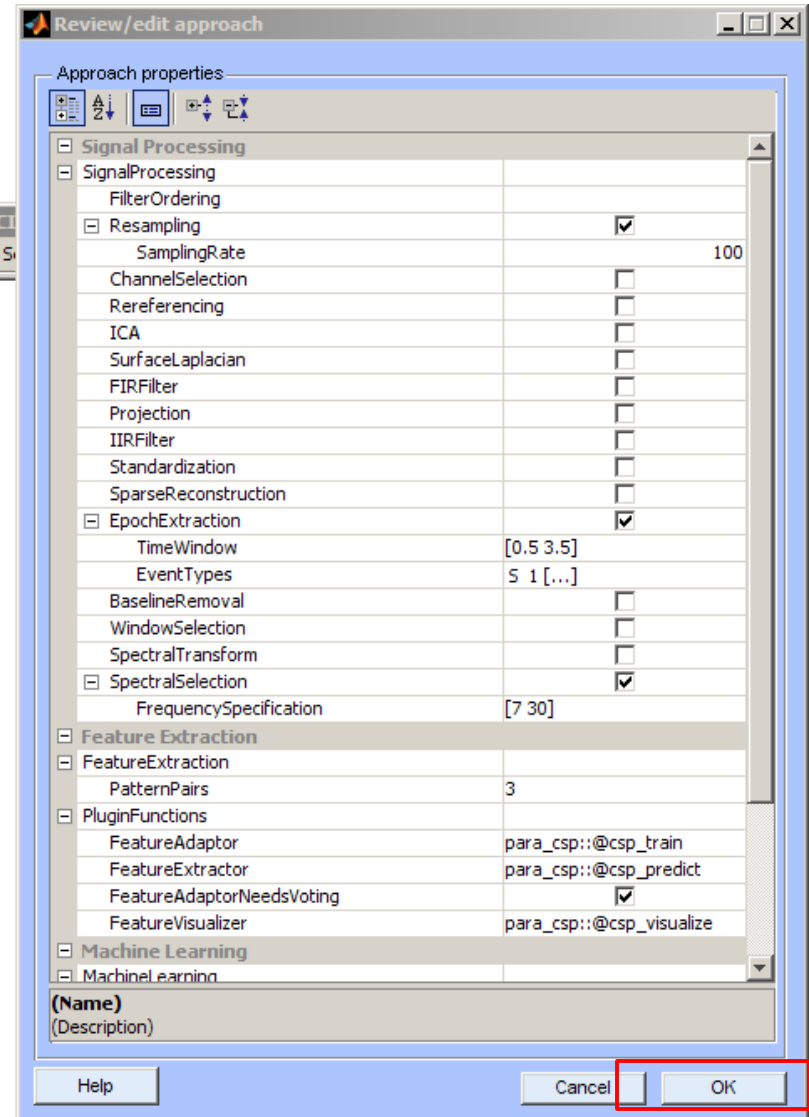
Configure approach



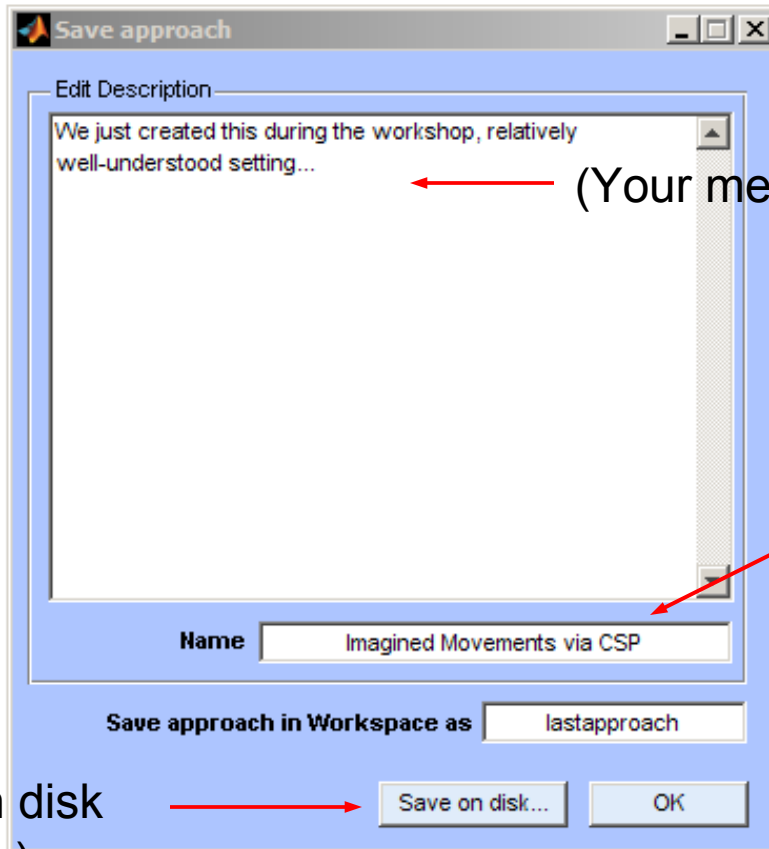
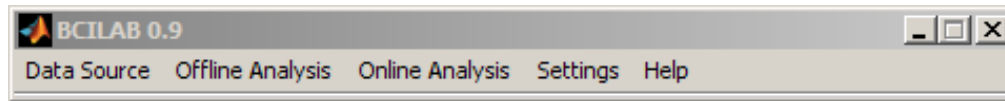
(Takes a while after clicking OK)

Review/edit approach

- The next panel allows to edit all properties of the approach
- Filter stages can be added and configured
- Feature extraction can be configured
- Machine learning components can be selected and configured
- For now, nothing to do



Save approach



← (Your method description)

(Your executive summary strongly recommended to name this properly)

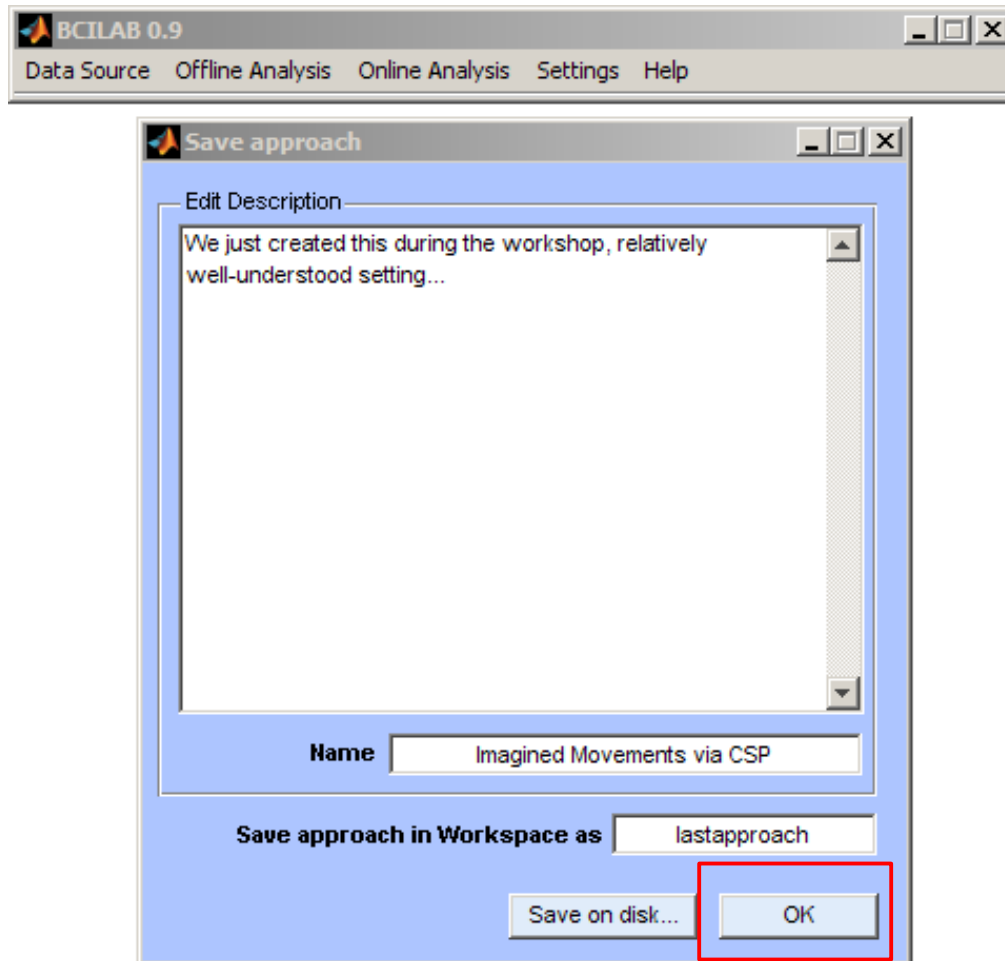
(You can save on disk for later reuse)



Save on disk...

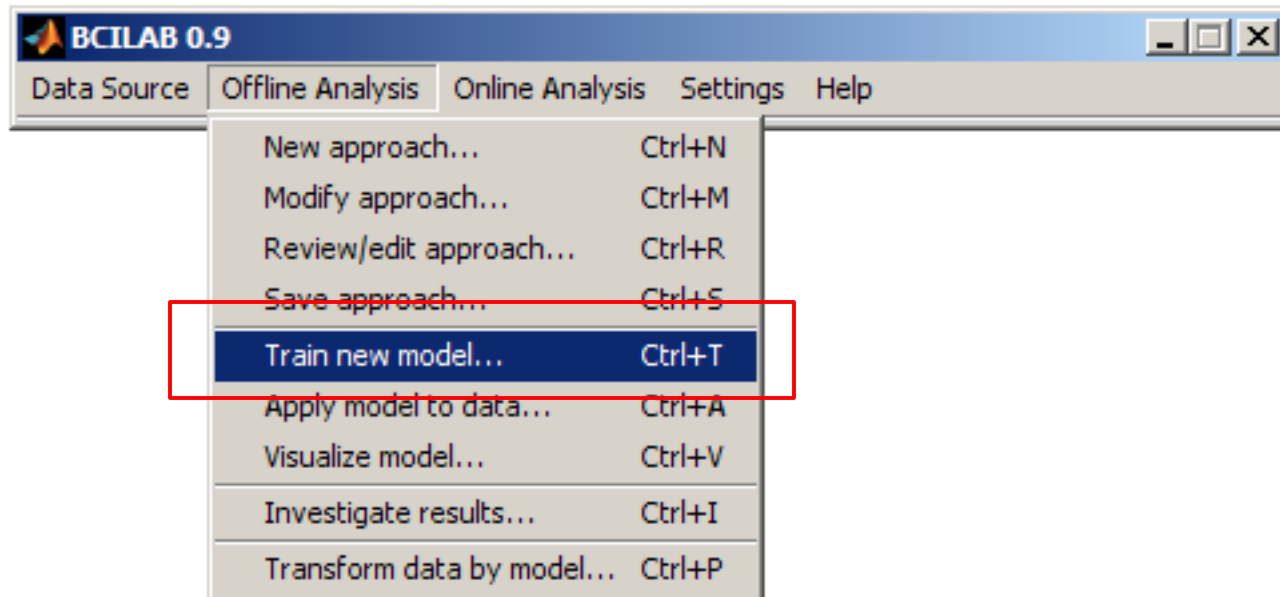
OK

Save approach



Learn a predictive model

- Put the method to the test...



Learn a predictive model

Calibrate a model

Selected approach: lastapproach ("Imagined Movements vi...")

Calibration data source: lastdata ("imag.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

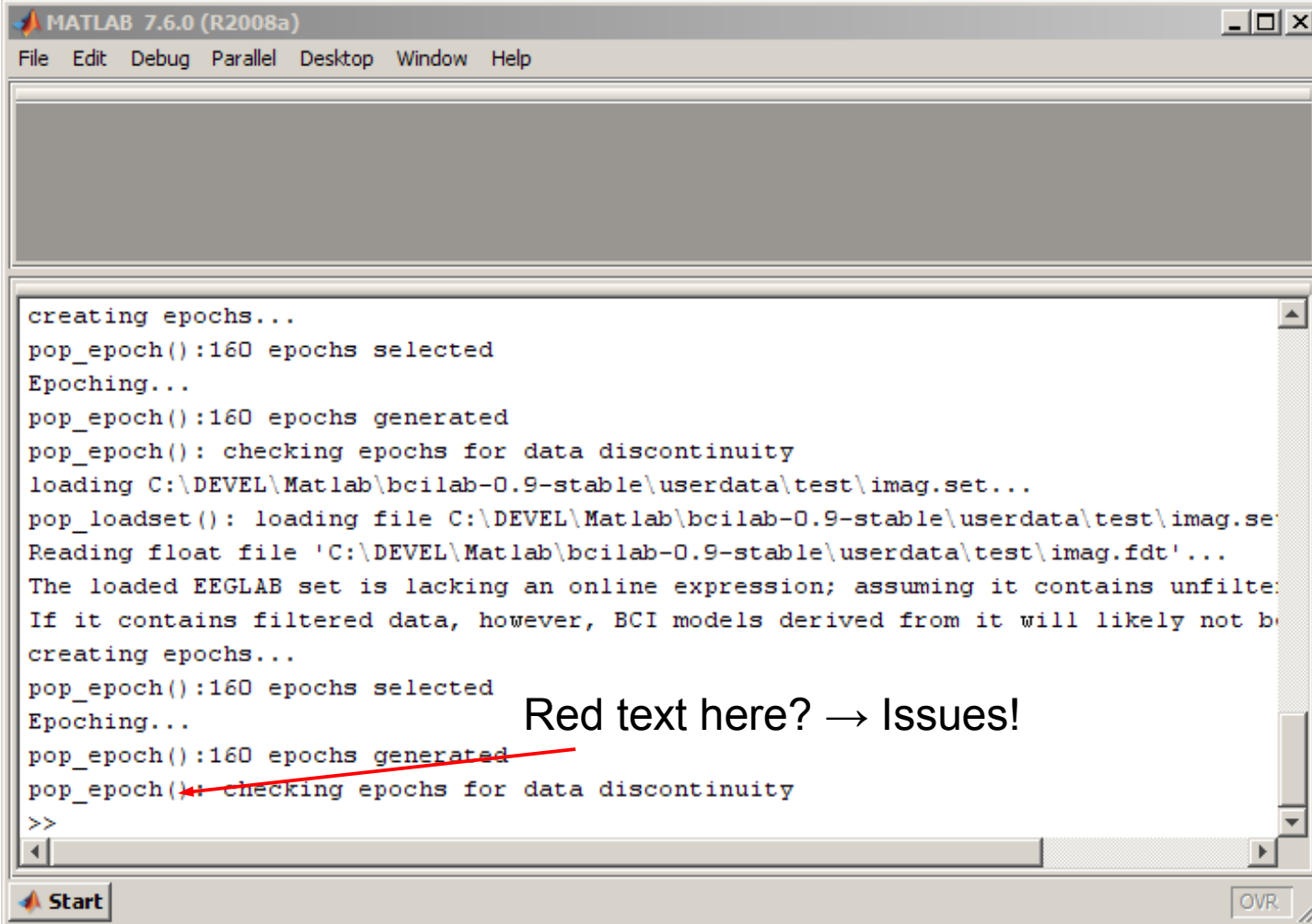
Buttons: Help, Cancel, OK

(Defines the performance metric)

(smaller number: faster, but lower quality estimates)

(If this is checked, you get performance estimates)

Wait for results



```
MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help

creating epochs...
pop_epoch():160 epochs selected
Epoching...
pop_epoch():160 epochs generated
pop_epoch(): checking epochs for data discontinuity
loading C:\DEVEL\Matlab\bcilab-0.9-stable\userdata\test\imag.set...
pop_loadset(): loading file C:\DEVEL\Matlab\bcilab-0.9-stable\userdata\test\imag.se
Reading float file 'C:\DEVEL\Matlab\bcilab-0.9-stable\userdata\test\imag.fdt'...
The loaded EEGLAB set is lacking an online expression; assuming it contains unfiltered
If it contains filtered data, however, BCI models derived from it will likely not be
creating epochs...
pop_epoch():160 epochs selected
Epoching...
pop_epoch():160 epochs generated
pop_epoch(): checking epochs for data discontinuity
>>
```

Red text here? → Issues!

Start OVR

Review results

The screenshot shows a 'Review Results' dialog box with two main sections: 'Data Summary' and 'Data Details'.

Data Summary: Displays the text "Error rate : 0.11 +/- 0.15 (N=10)". A red arrow points to the value "0.11".

Data Details: A table with 10 rows and 2 columns. The column headers are "Error rate". The values for each row are: 0.0625, 0.1875, 0.5000, 0, 0.0625, 0.1250, 0.1250, 0.0625, 0, 0. A red arrow points to the value "0.5000" in the third row.

At the bottom of the dialog box, there are five buttons: "Help", "Explore...", "Export...", "Save...", and "OK". The "OK" button is highlighted with a red border.

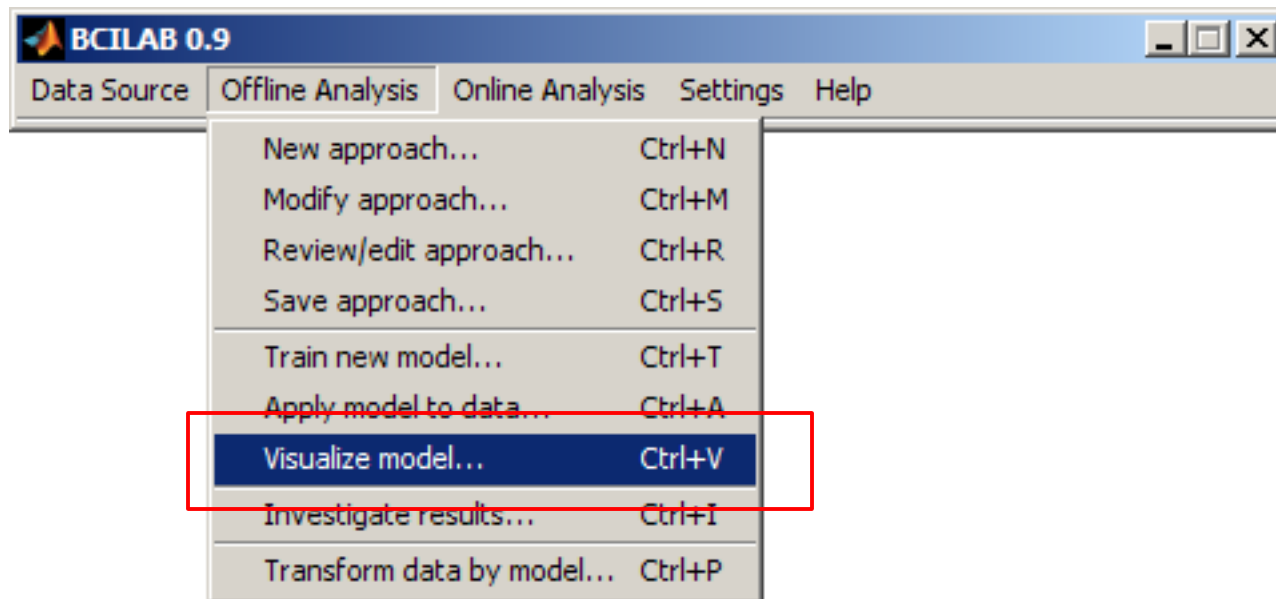
Annotations on the slide:

- "Mean loss estimate (fraction of mis-classified trials)" with a red arrow pointing to the "0.11" in the Data Summary.
- "Loss for every partition of the data set" with a red arrow pointing to the "0.5000" in the Data Details table.

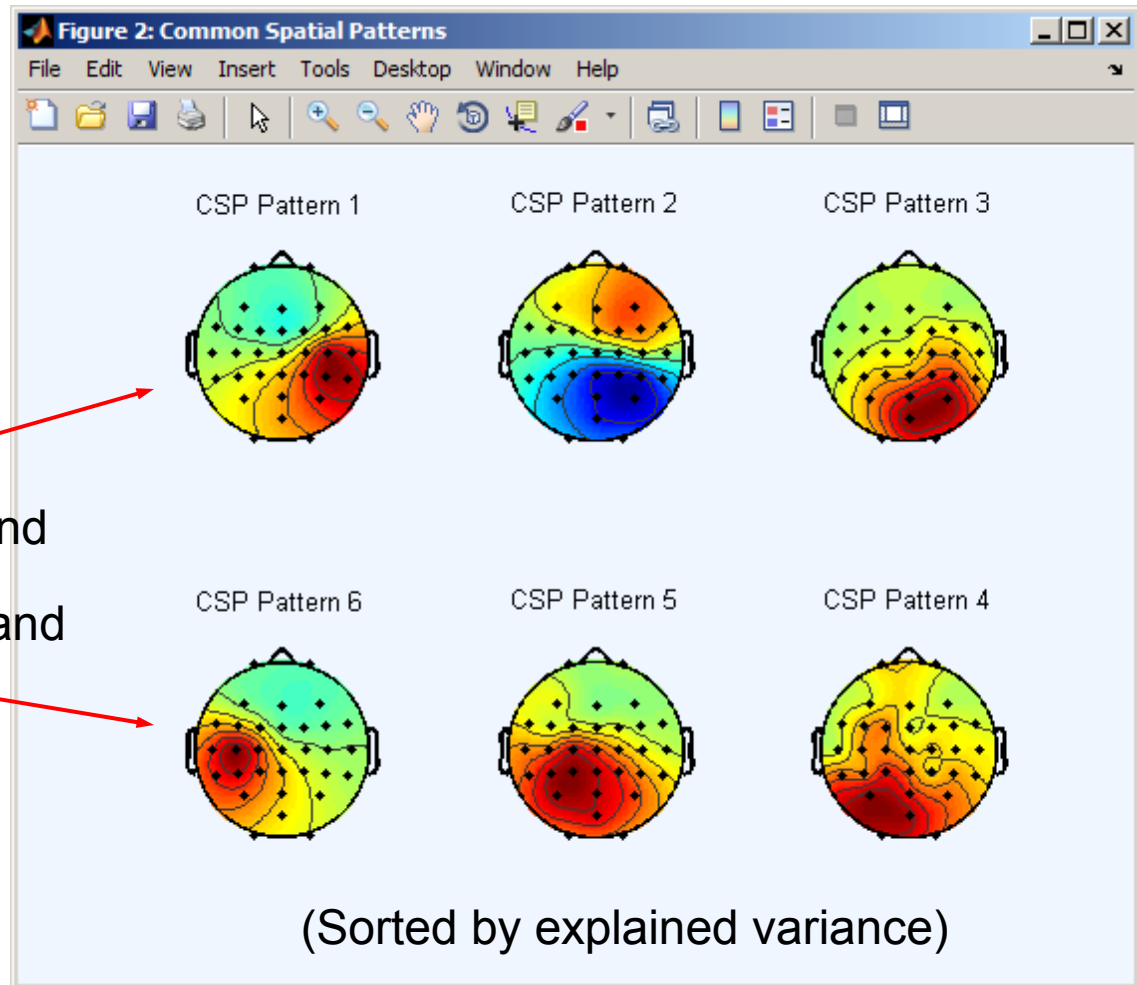
Review results

- 11% error rate is quite good for imagined movements; mean across studies & methods is probably closer to 25%
- chance level is here 50% (keep that in mind when evaluating)
- You may get multiple outputs (e.g., false positives, true positives, which show up in the table), depending on loss measure

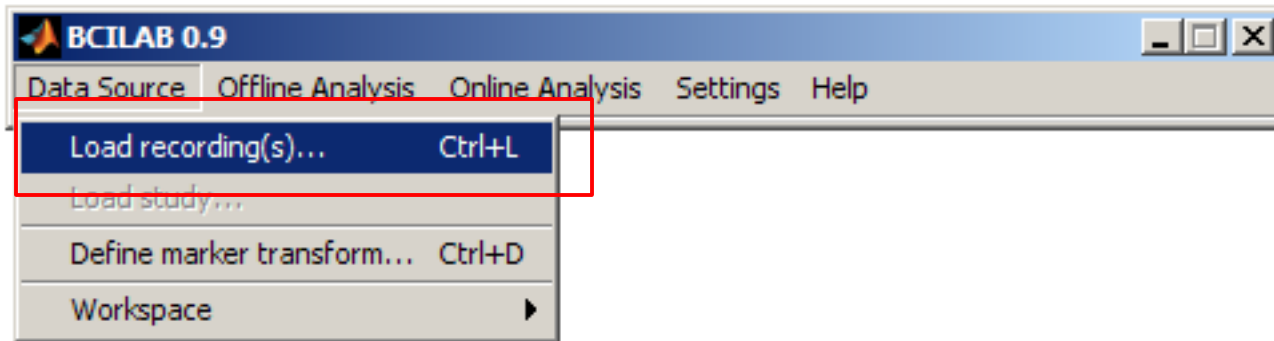
Visualize model properties



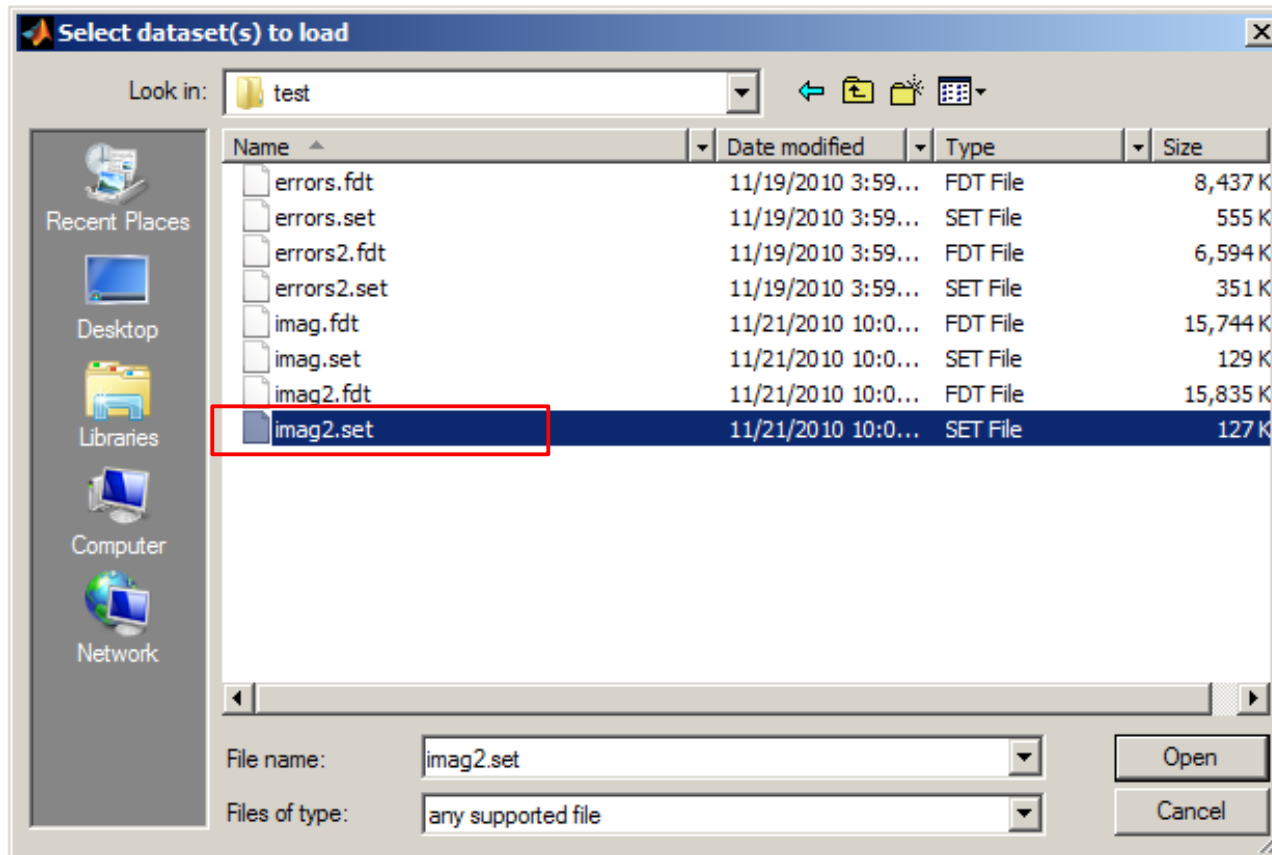
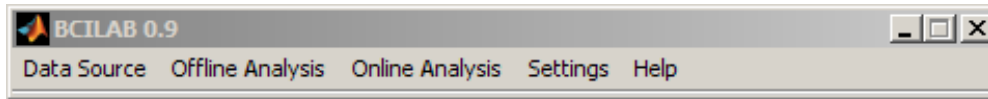
Visualize model properties



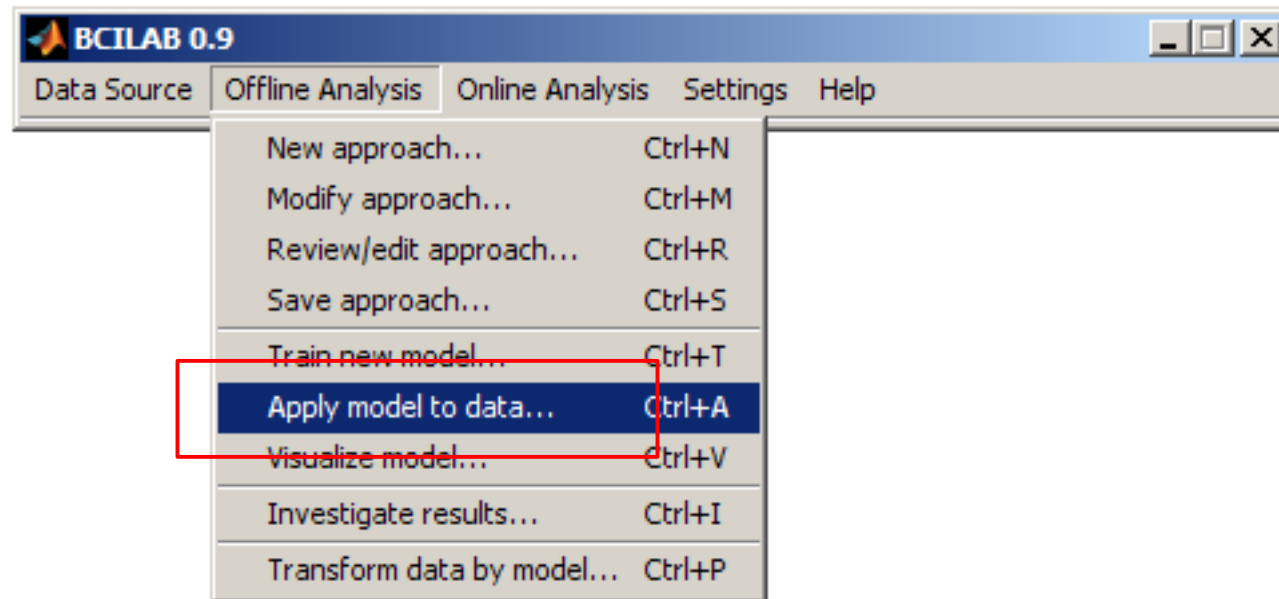
Apply model to 2nd session



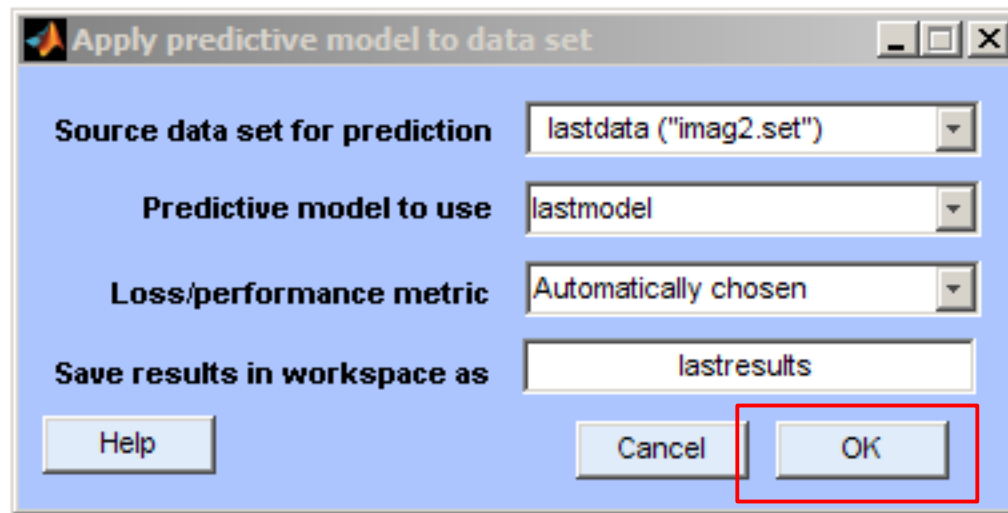
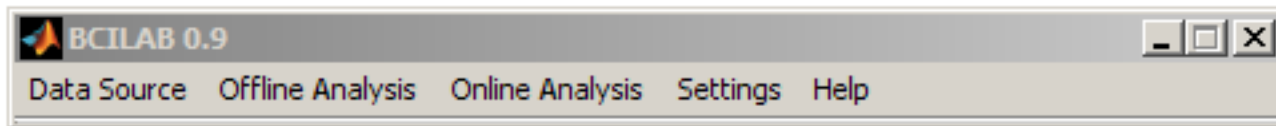
Apply model to 2nd session



Apply model to 2nd session



Apply model to 2nd session



Review results

The image shows a software dialog box titled "Review Results". It is divided into two main sections: "Data Summary" and "Data Details".

Data Summary: A text area containing the text "Error rate : 0.09 +/- 0.00 (N=1)".

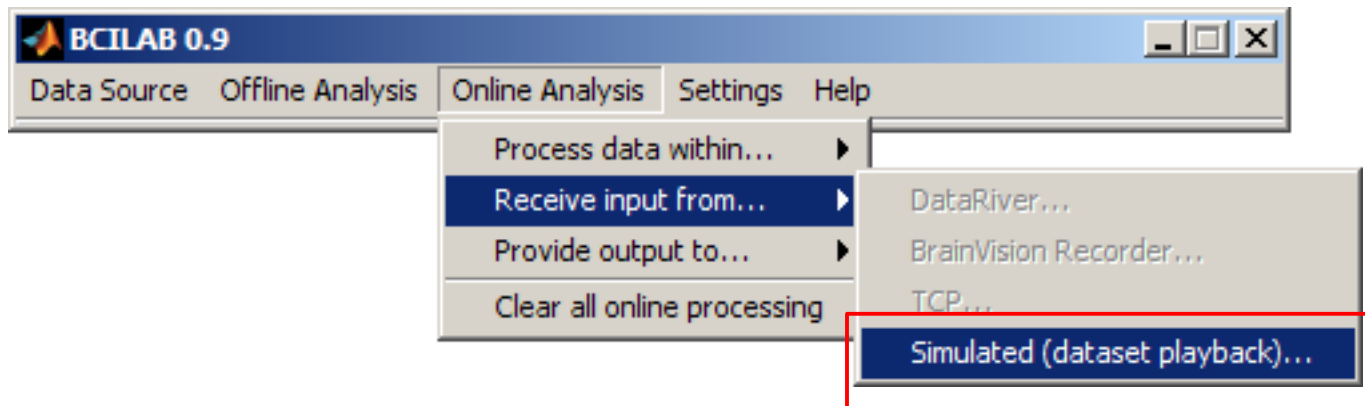
Data Details: A table with one row and one column. The column header is "Error rate" and the row value is "0.0938".

Buttons: At the bottom of the dialog, there are five buttons: "Help", "Explore...", "Export...", "Save...", and "OK". The "OK" button is highlighted with a red rectangular box.

	Error rate
1	0.0938

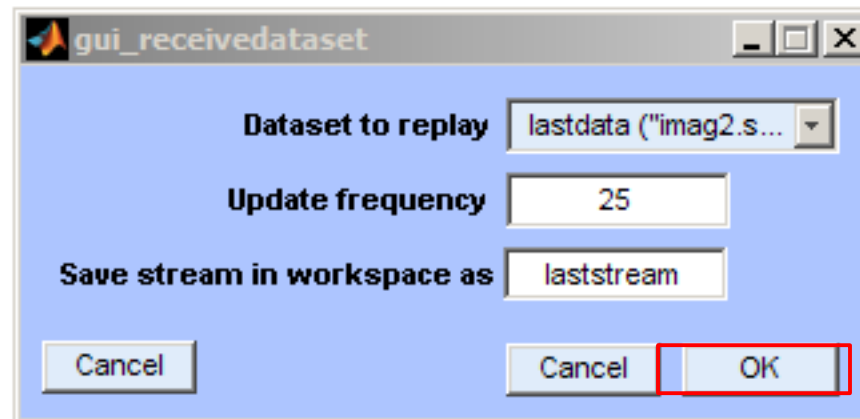
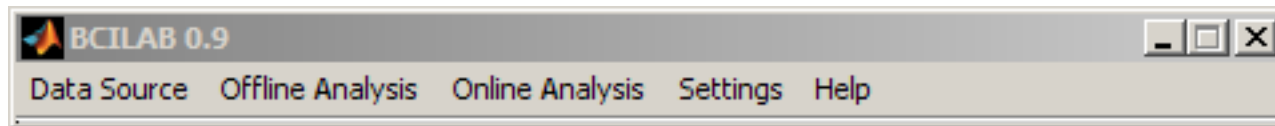
OR: Apply model online

- (if you have a subject sitting next to you)
- Today: use a simulated data source (playing back the 2nd session)

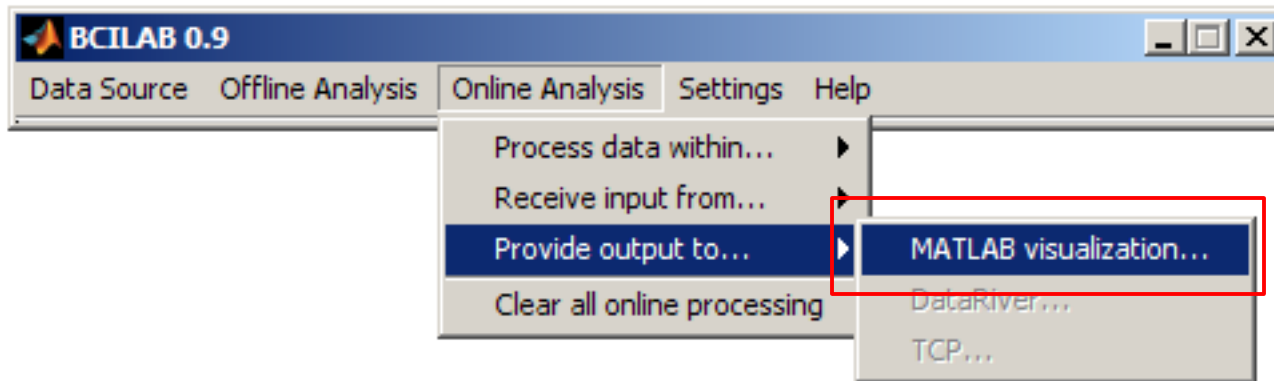


Apply model online

- This adds a data feed process in the background

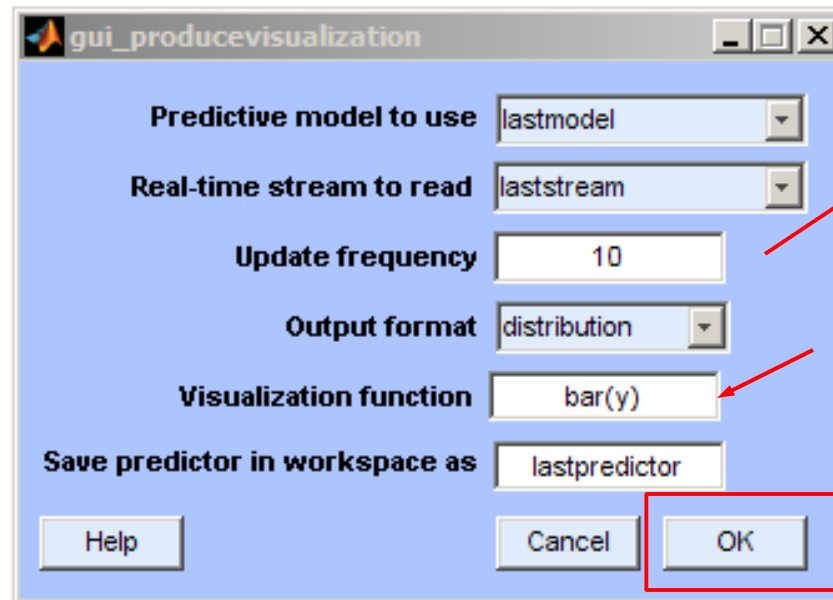
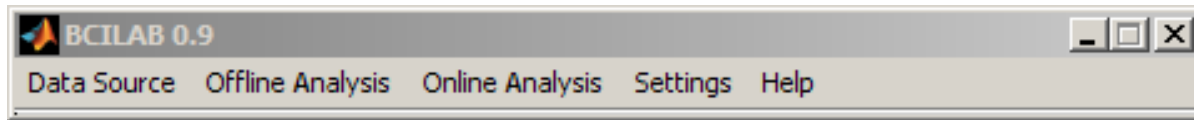


Apply model online



Apply model online

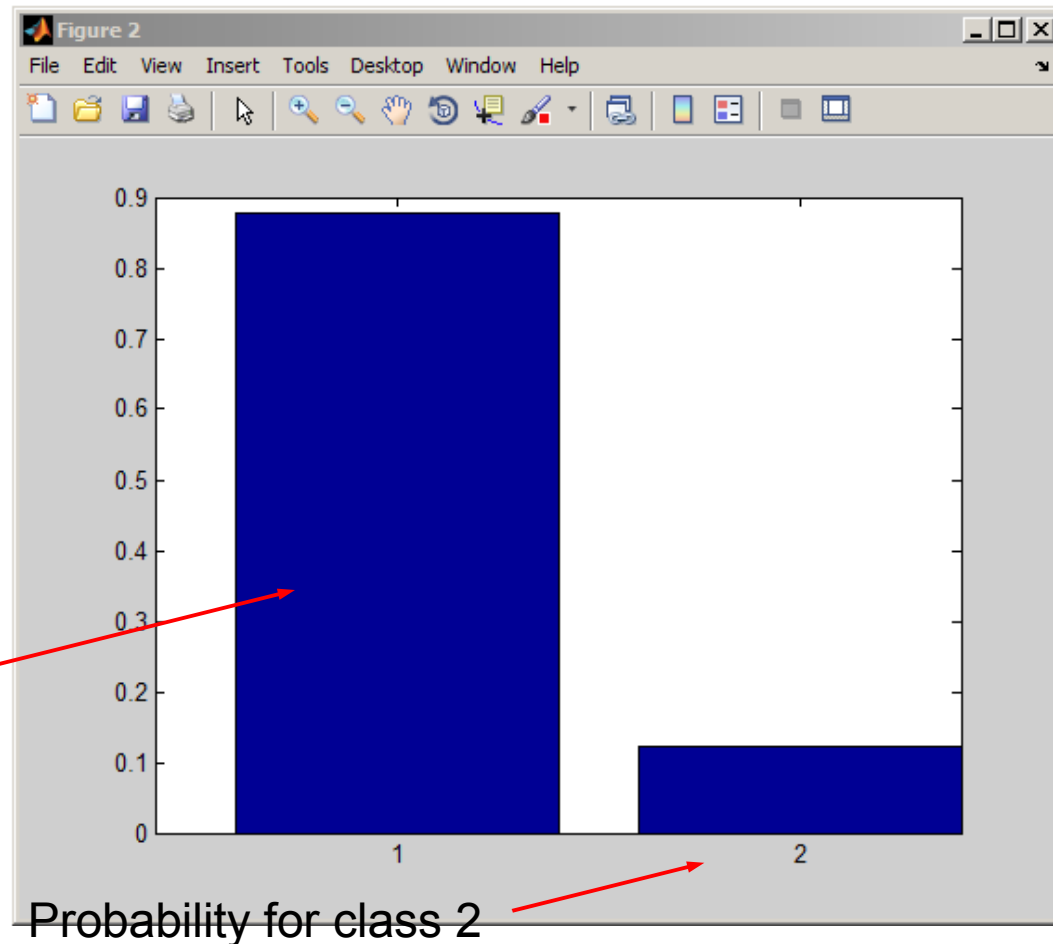
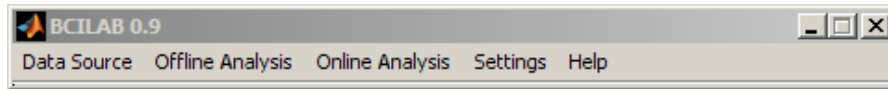
- This adds a real-time inference process in the background



25 Hz if your computer is fast enough

Forwarded to this display command

Real-time output



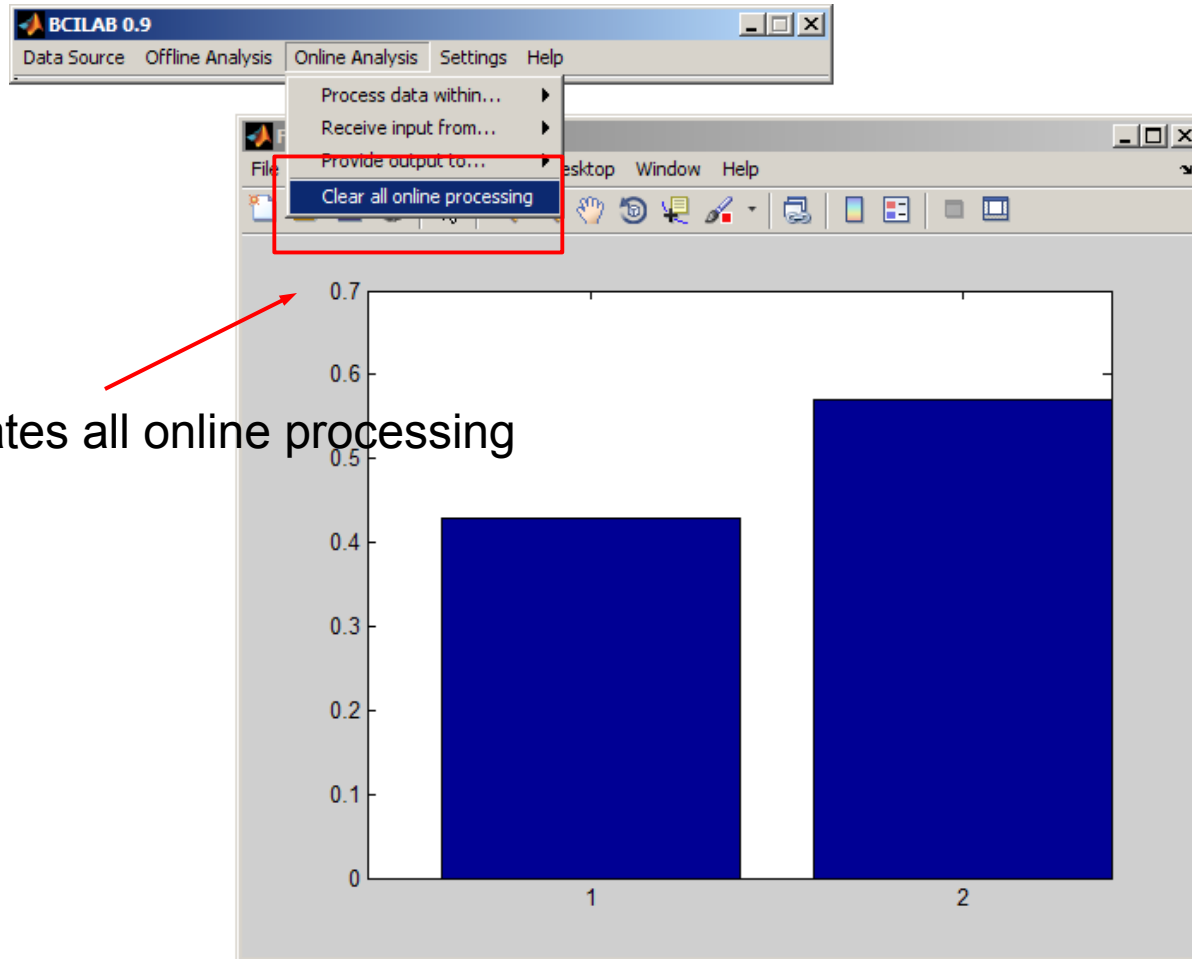
Probability for class 1

Probability for class 2

Real-time output

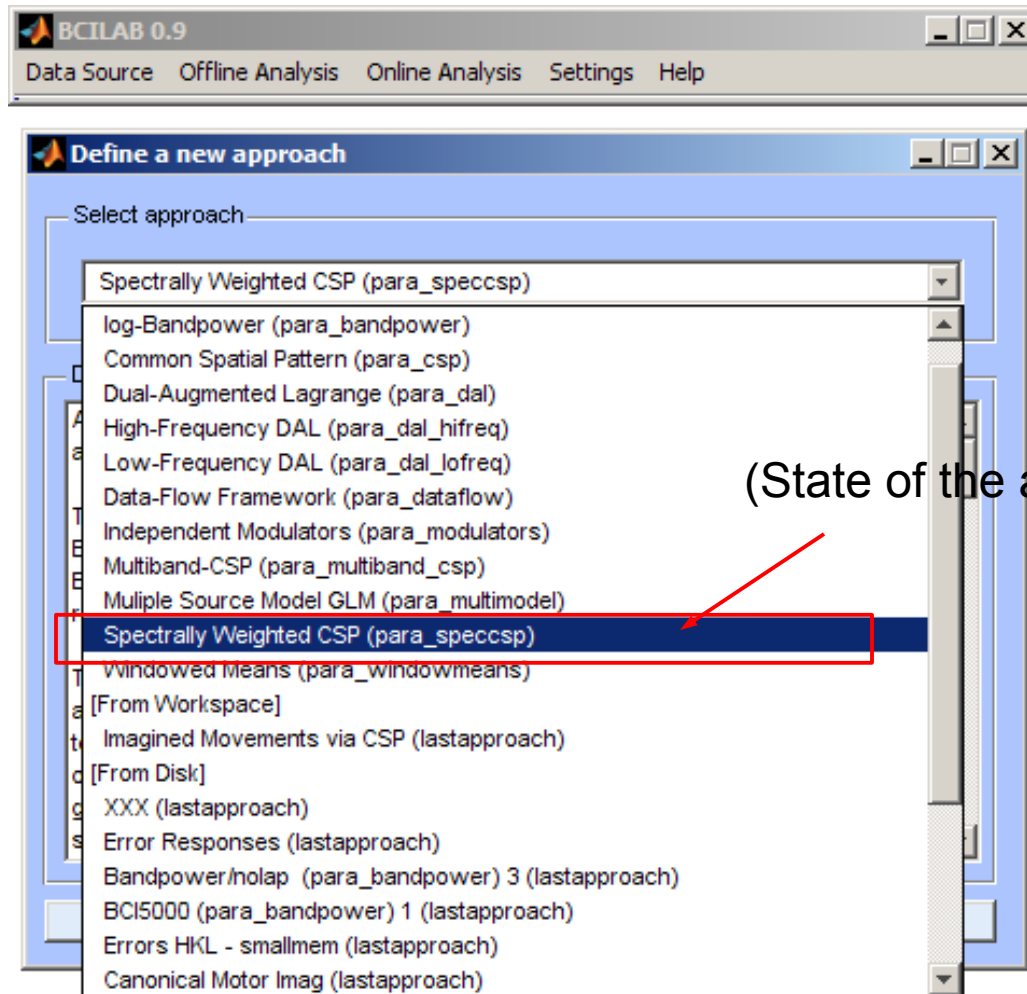
- If you have more classes, you get more bars
- You can also remap to other parameters (e.g. expected value)
- Note: the simple graphics command always renders into the current window

Real-time output

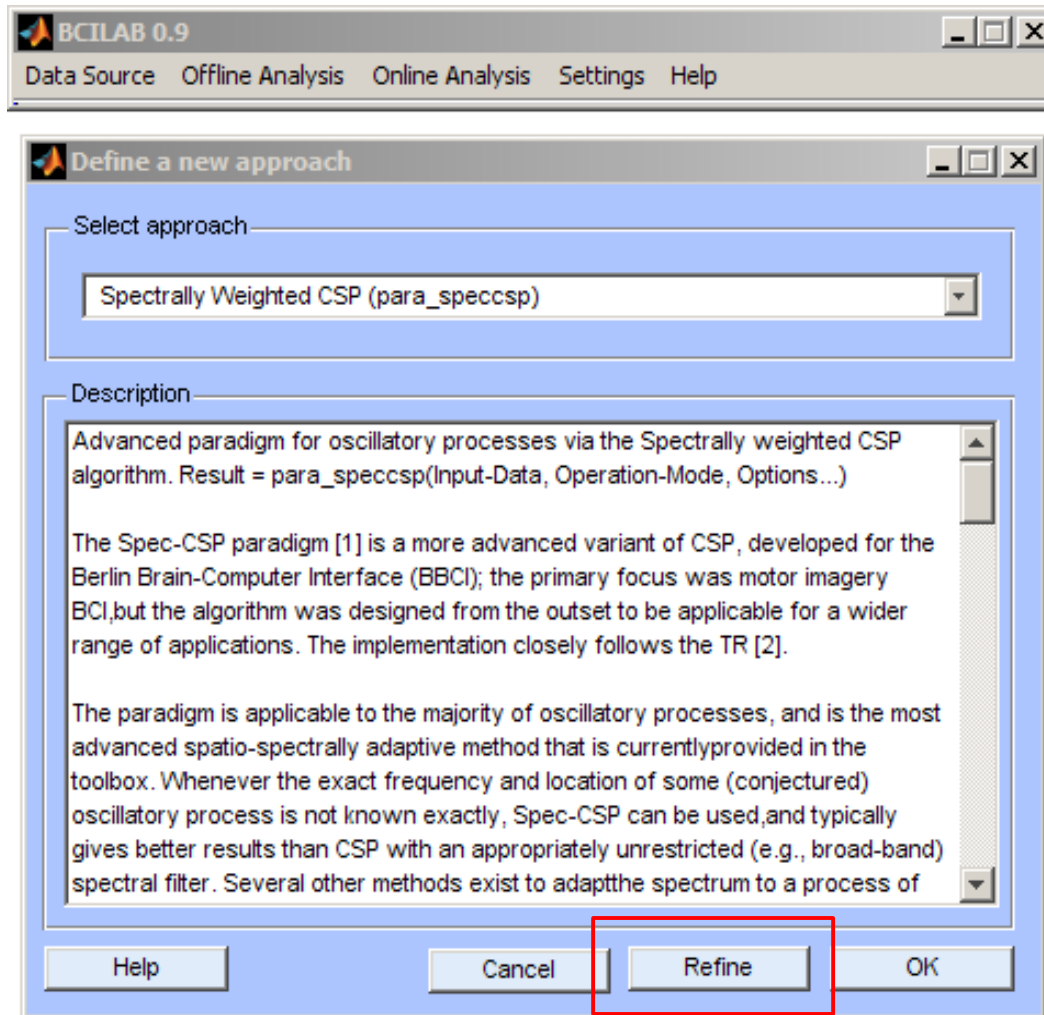


This terminates all online processing

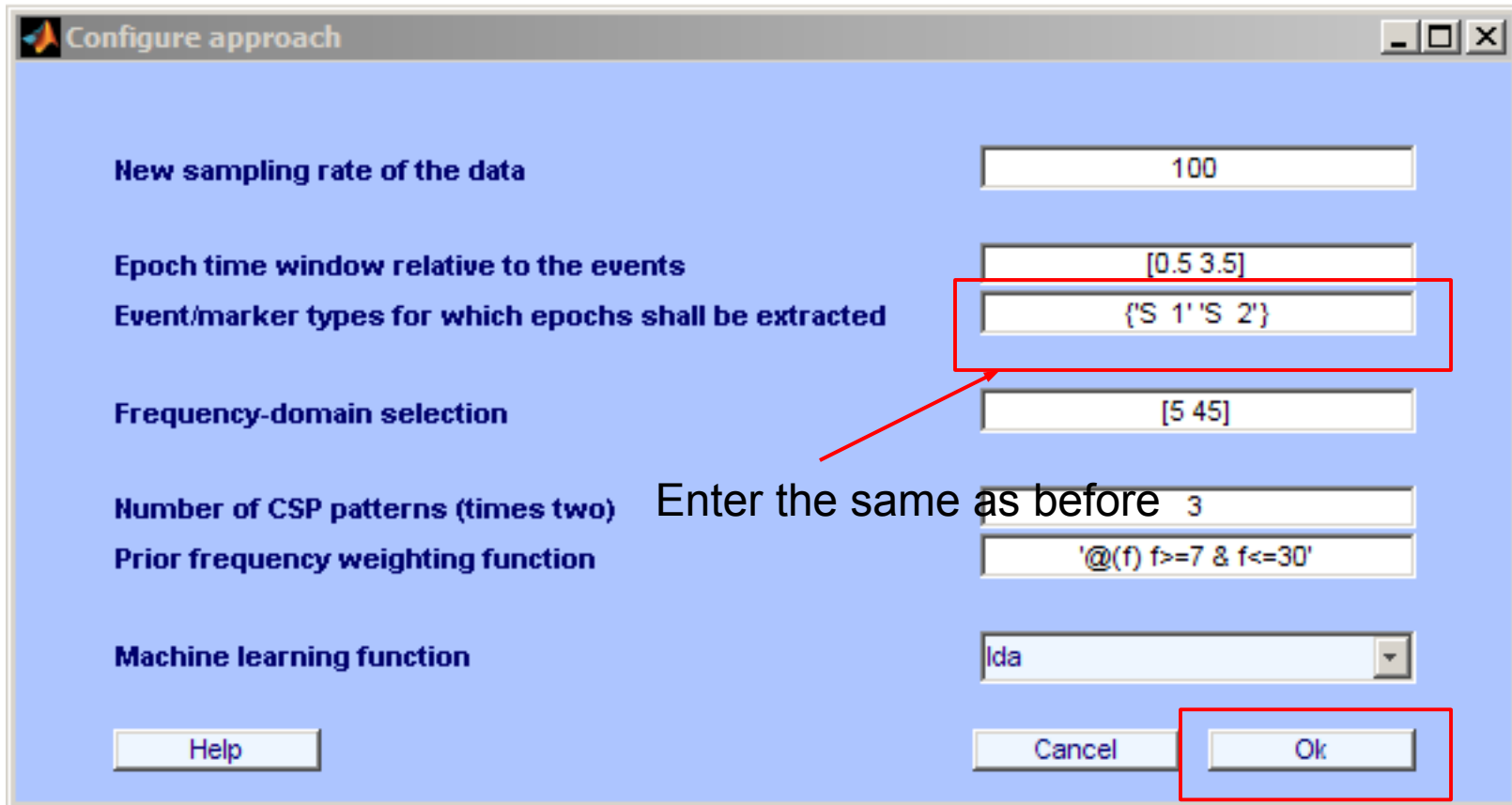
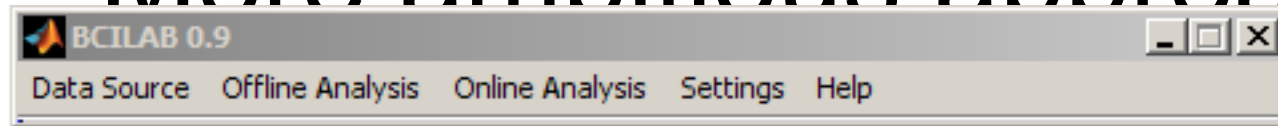
More ambitious approach?



More ambitious approach?



More ambitious approach



Train model, review results

- Note that the model calibration takes longer for Spec-CSP.

The screenshot shows a 'Review Results' dialog box with two main sections: 'Data Summary' and 'Data Details'. The 'Data Summary' section displays the text 'Error rate : 0.05 +/- 0.05 (N=10)' with a red arrow pointing to it and the text '5 percent!' below. The 'Data Details' section contains a table with 10 rows and one column labeled 'Error rate'. The table data is as follows:

	Error rate
1	0
2	0.0625
3	0
4	0
5	0.0625
6	0.0625
7	0.1250
8	0.1250
9	0.0625
10	0

At the bottom of the dialog, there are buttons for 'Help', 'Explore...', 'Export...', 'Save...', and 'OK'. The 'OK' button is highlighted with a red rectangle. Below the main dialog, there is a smaller dialog box with fields for 'Save model in workspace as' (containing 'lastmodel') and 'Save stats in workspace as' (containing 'laststats'), along with 'Help', 'Cancel', and 'OK' buttons.

Next steps

- Run online, apply to dataset, edit parameters, try to improve results, ...

Use case 2

- Question: Can we predict whether the user perceives an event as being an error?
- Experimental data: EEG, 32 channels, 2 sessions

Experimental task

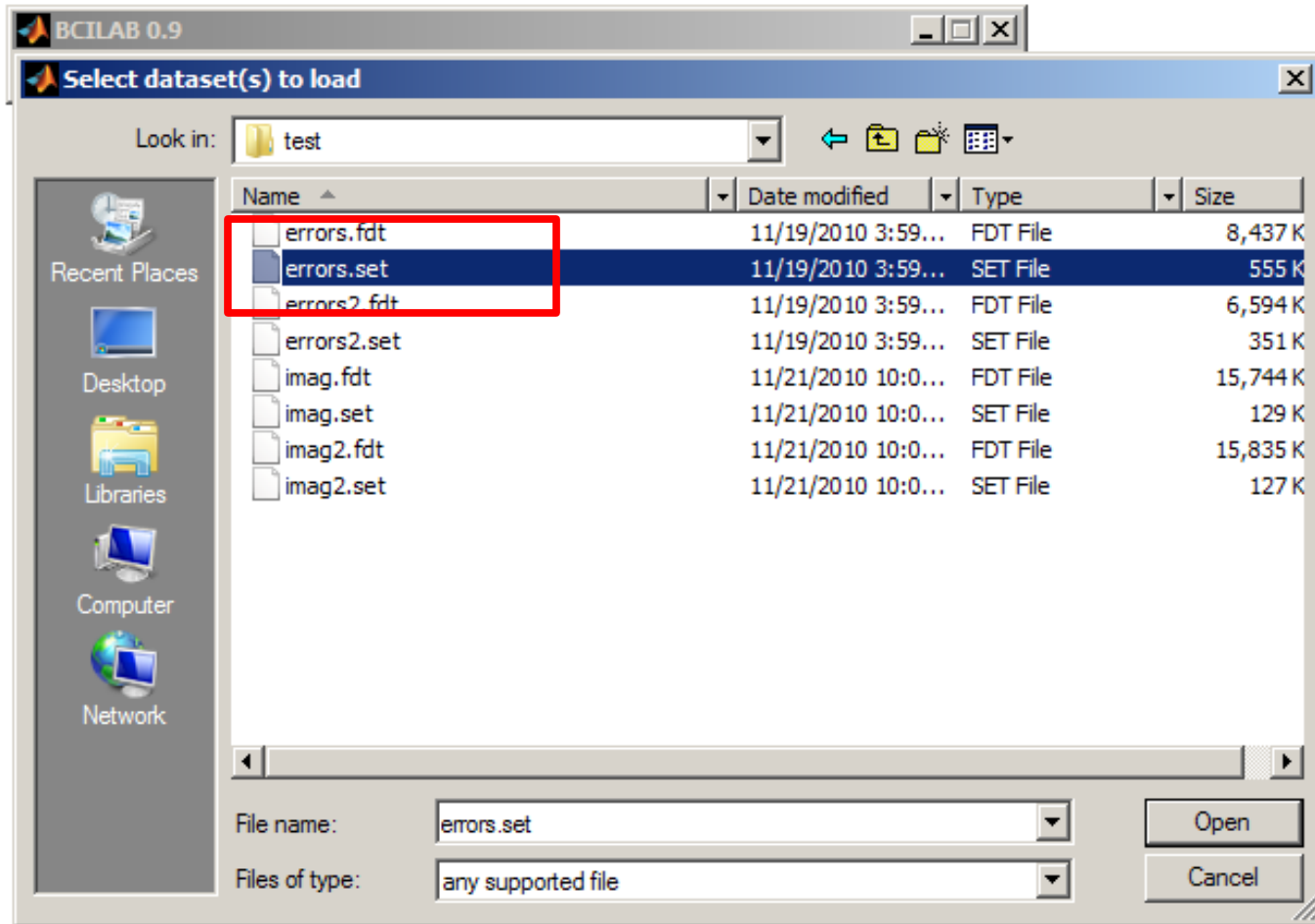
- Experimental task: ~100 randomized trials, 3 types of stimuli:
 - expected/correct event: type 'S 11'
 - unexpected event A: type 'S 12'
 - unexpected event B: type 'S 13'
- Sample:



Experimental task

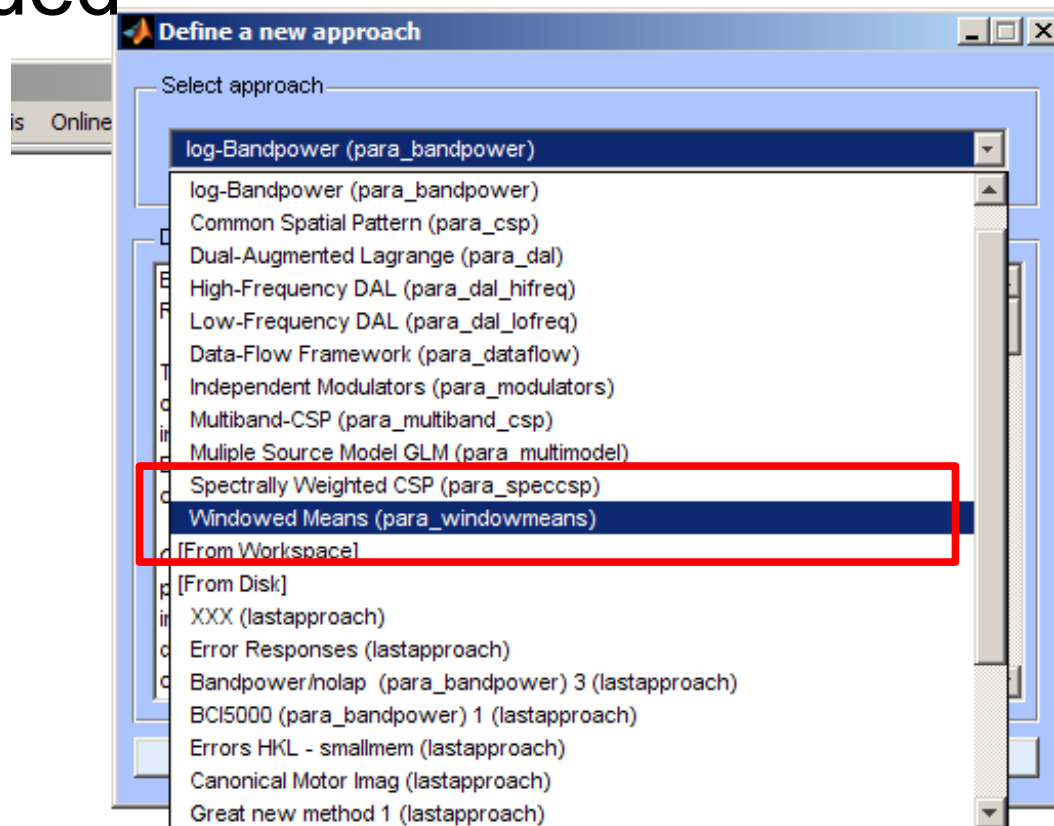
- The colored letter either rotates as expected (in response to a user key command), or differently

Load data

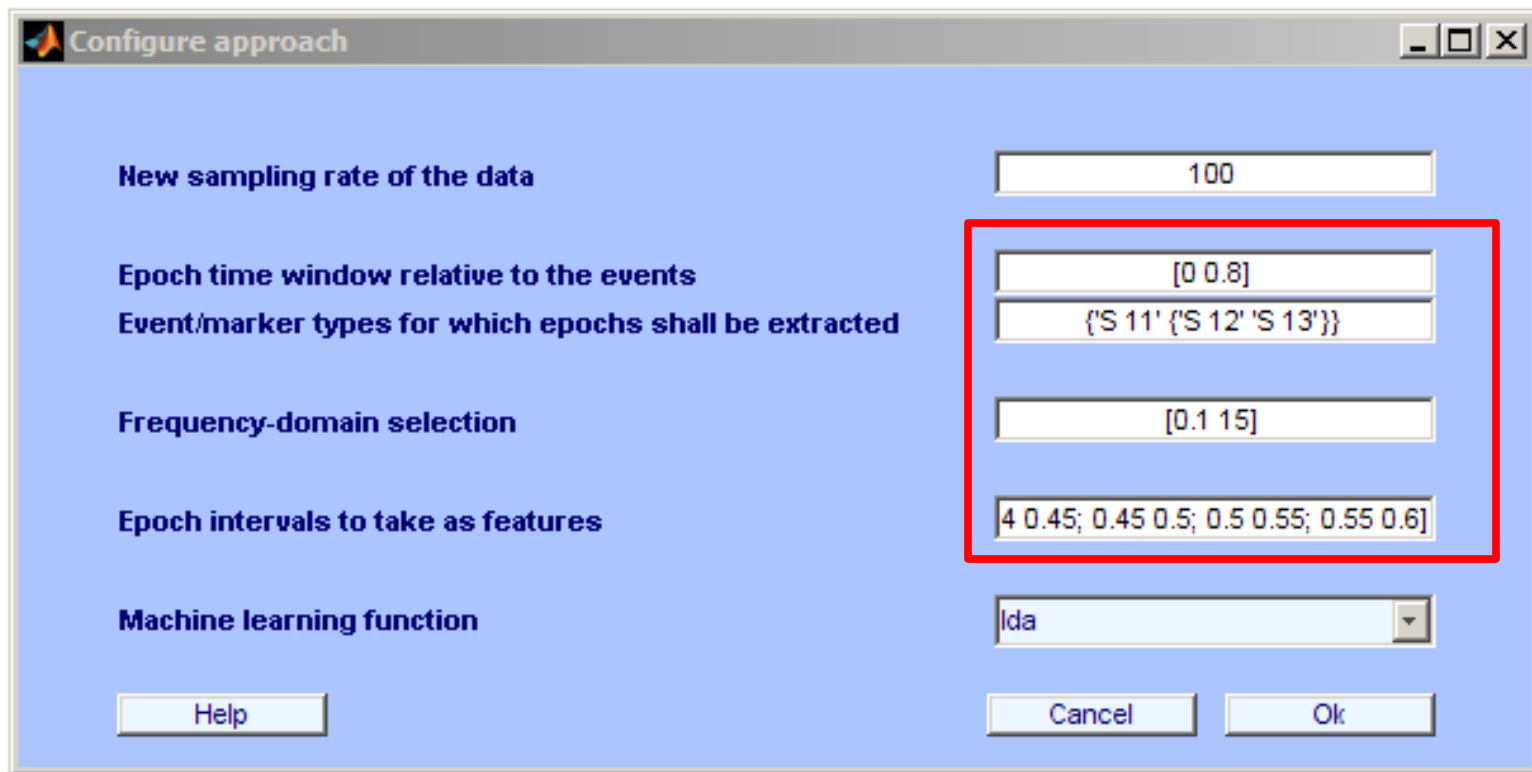
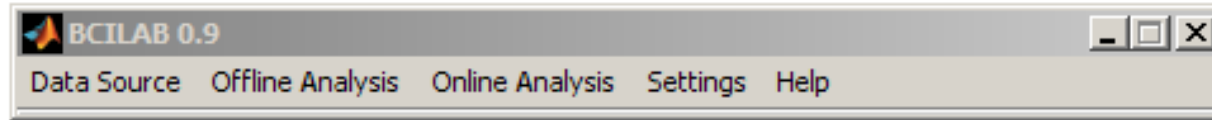


Define approach

- This time, an ERP-specific approach is needed



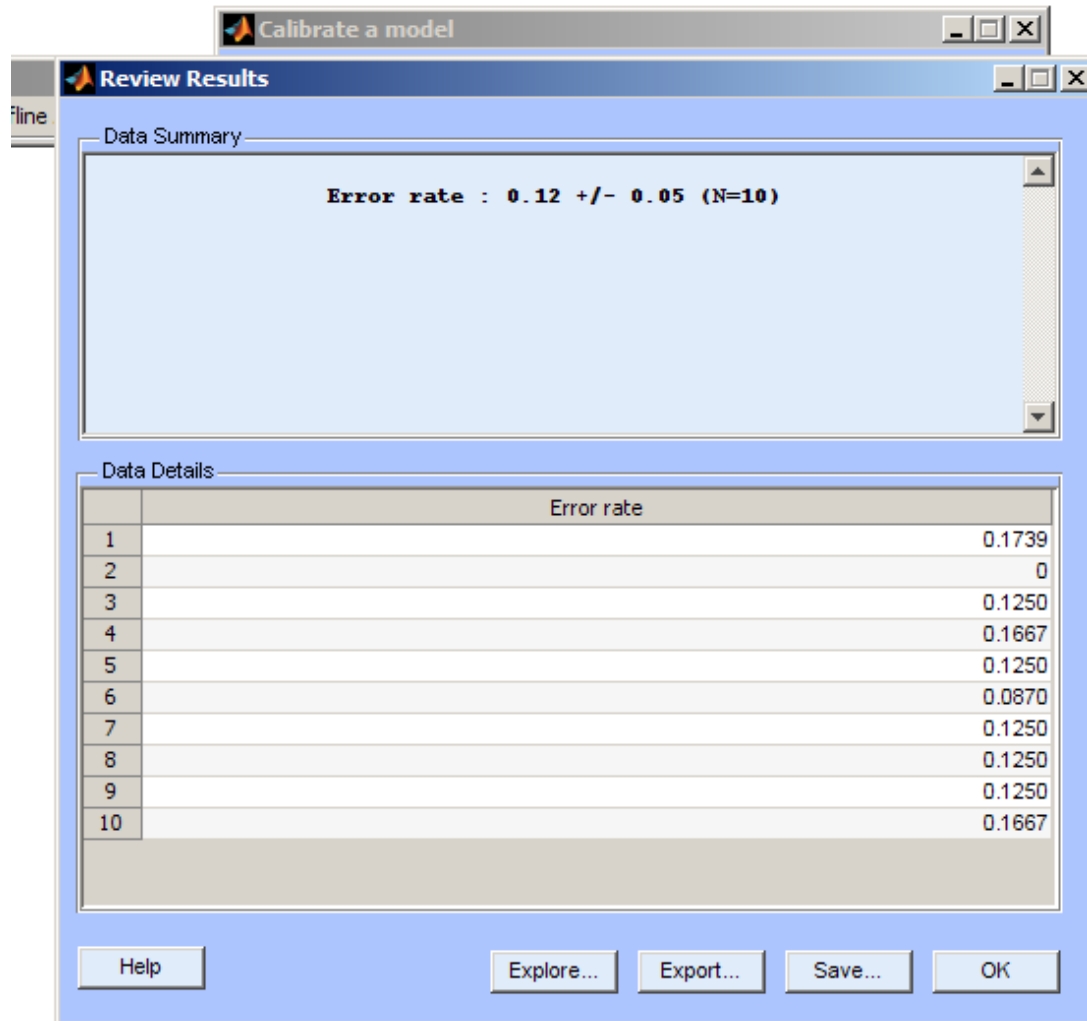
Major customizations



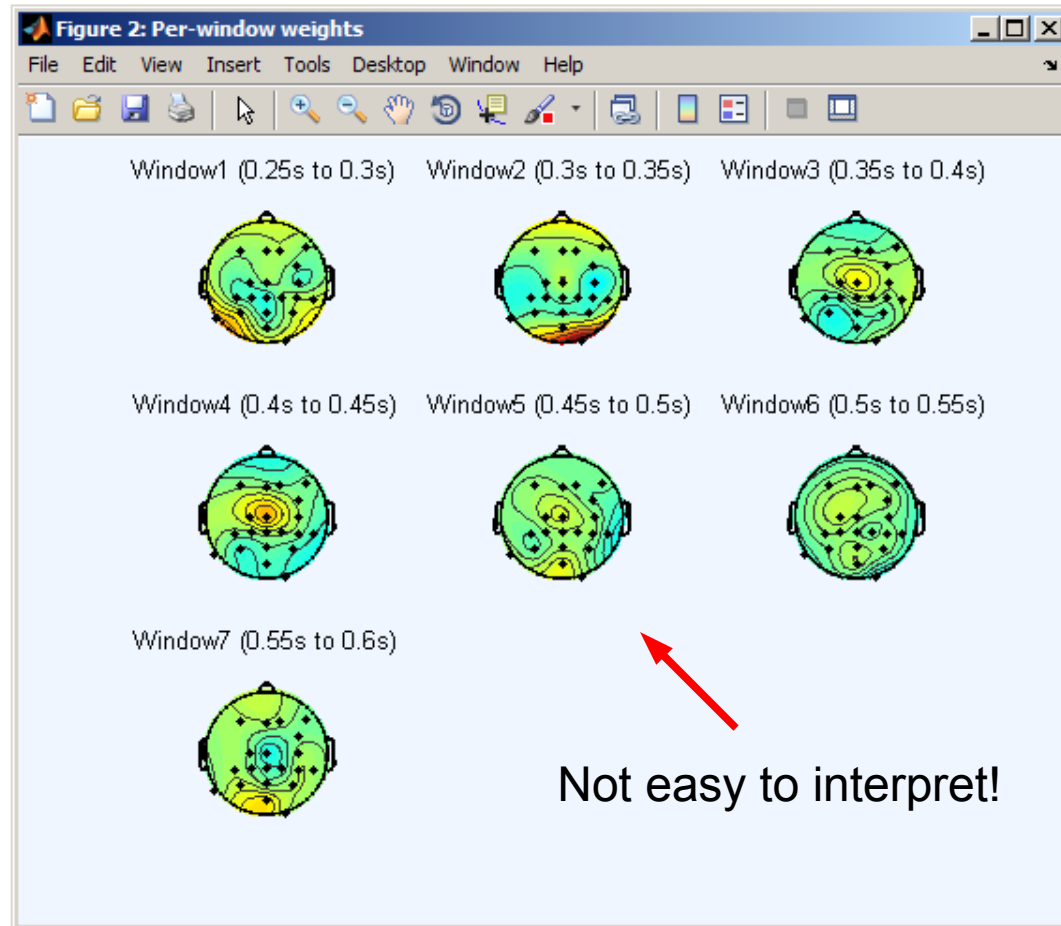
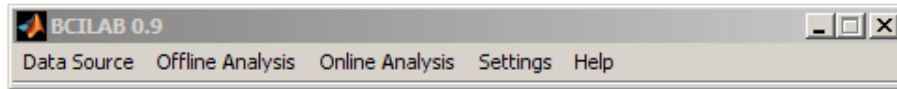
Event markers: {'S 11' {'S 12' 'S 13'}}

Epoch intervals: [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]

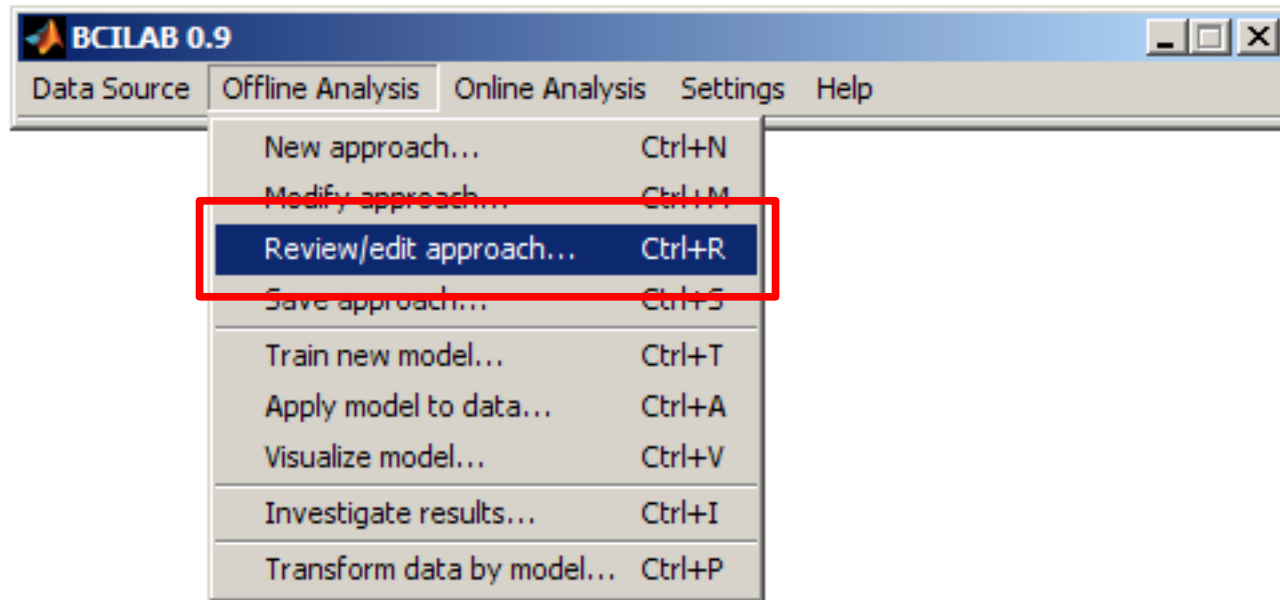
Train model, visualize



Train model, visualize



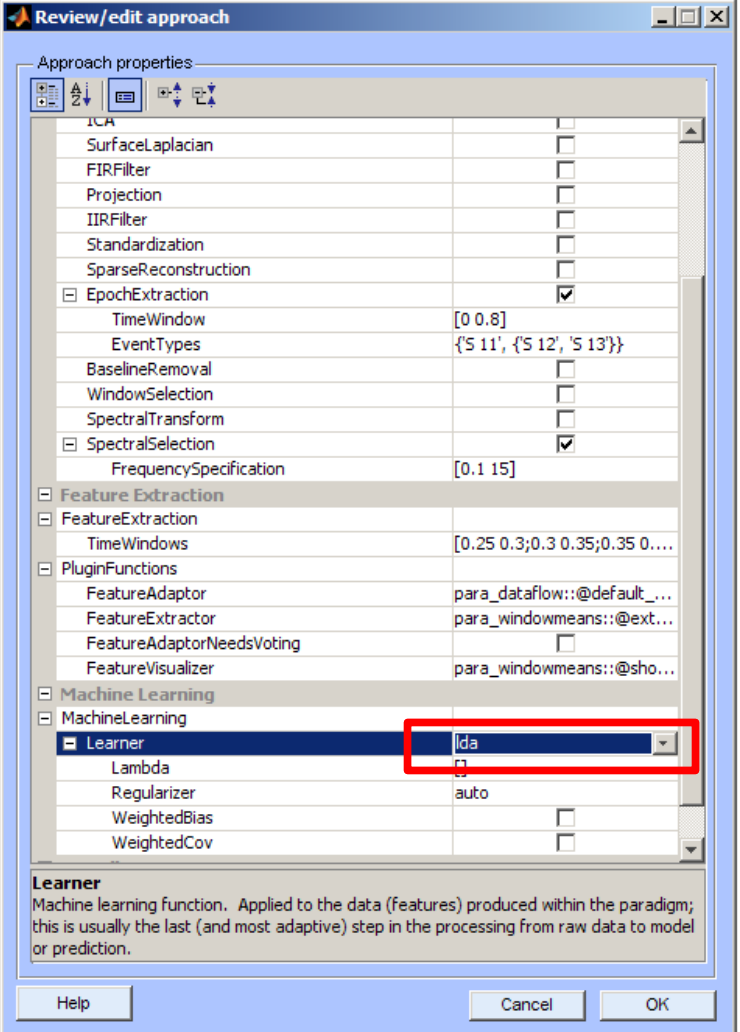
Using a sparse classifier



Using a sparse classifier

- Scroll down to Learner
- Select logreg instead of lda (also see help pane)
- do not close yet!

BCILAB 0.9
Data Source Offline



Review/edit approach

Approach properties

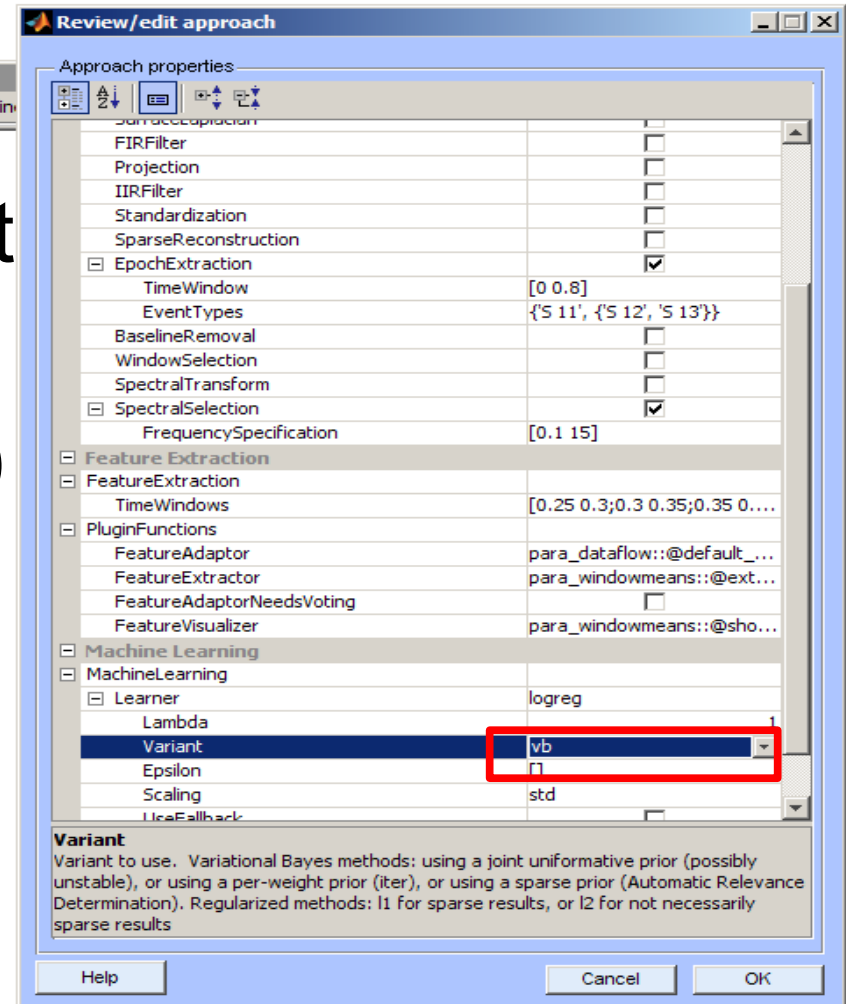
ICA		<input type="checkbox"/>
SurfaceLaplacian		<input type="checkbox"/>
FIRFilter		<input type="checkbox"/>
Projection		<input type="checkbox"/>
IIRFilter		<input type="checkbox"/>
Standardization		<input type="checkbox"/>
SparseReconstruction		<input type="checkbox"/>
EpochExtraction		<input checked="" type="checkbox"/>
TimeWindow	[0 0.8]	
EventTypes	{'S 11', {'S 12', 'S 13}}	
BaselineRemoval		<input type="checkbox"/>
WindowSelection		<input type="checkbox"/>
SpectralTransform		<input type="checkbox"/>
SpectralSelection		<input checked="" type="checkbox"/>
FrequencySpecification	[0.1 15]	
Feature Extraction		
FeatureExtraction		
TimeWindows	[0.25 0.3;0.3 0.35;0.35 0....	
PluginFunctions		
FeatureAdaptor	para_dataflow::@default_...	
FeatureExtractor	para_windowmeans::@ext...	
FeatureAdaptorNeedsVoting		<input type="checkbox"/>
FeatureVisualizer	para_windowmeans::@sho...	
Machine Learning		
MachineLearning		
Learner	lda	
Lambda		<input type="checkbox"/>
Regularizer	auto	
WeightedBias		<input type="checkbox"/>
WeightedCov		<input type="checkbox"/>

Learner
Machine learning function. Applied to the data (features) produced within the paradigm; this is usually the last (and most adaptive) step in the processing from raw data to model or prediction.

Help Cancel OK

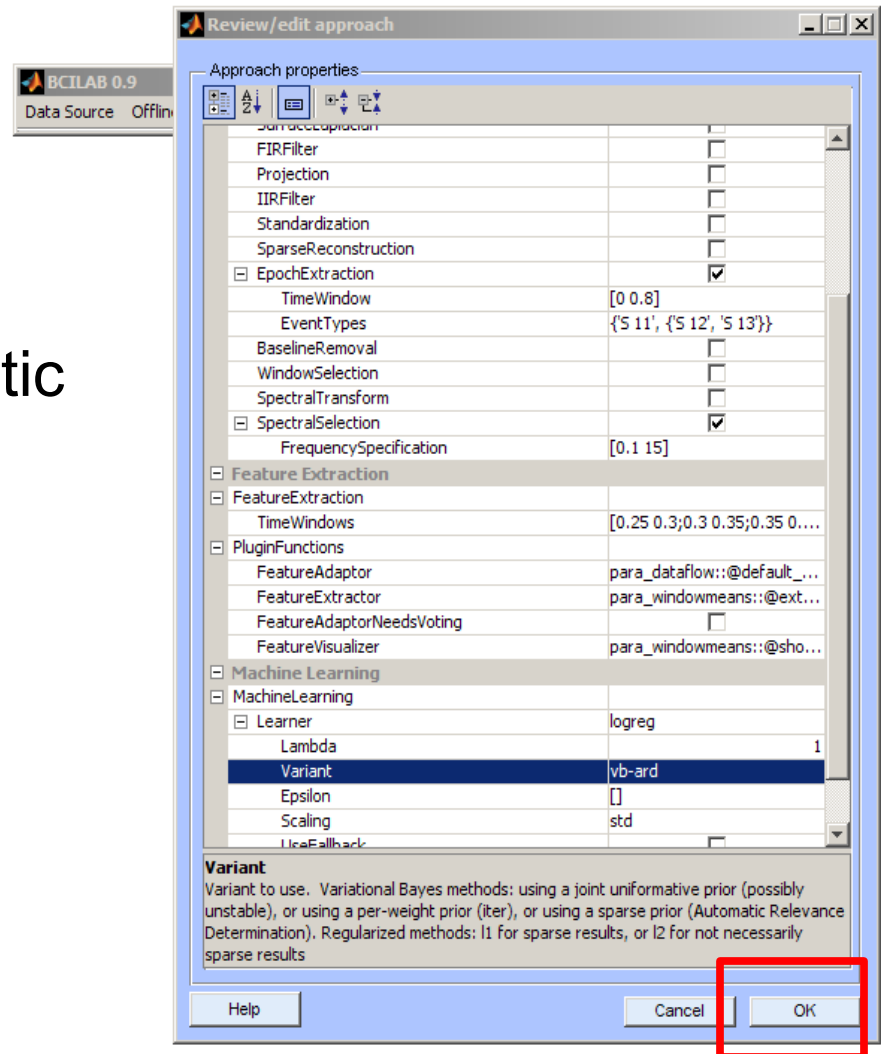
Using a sparse classifier

- Scroll down to Variant
- Select vb-ard instead of vb (again see help)

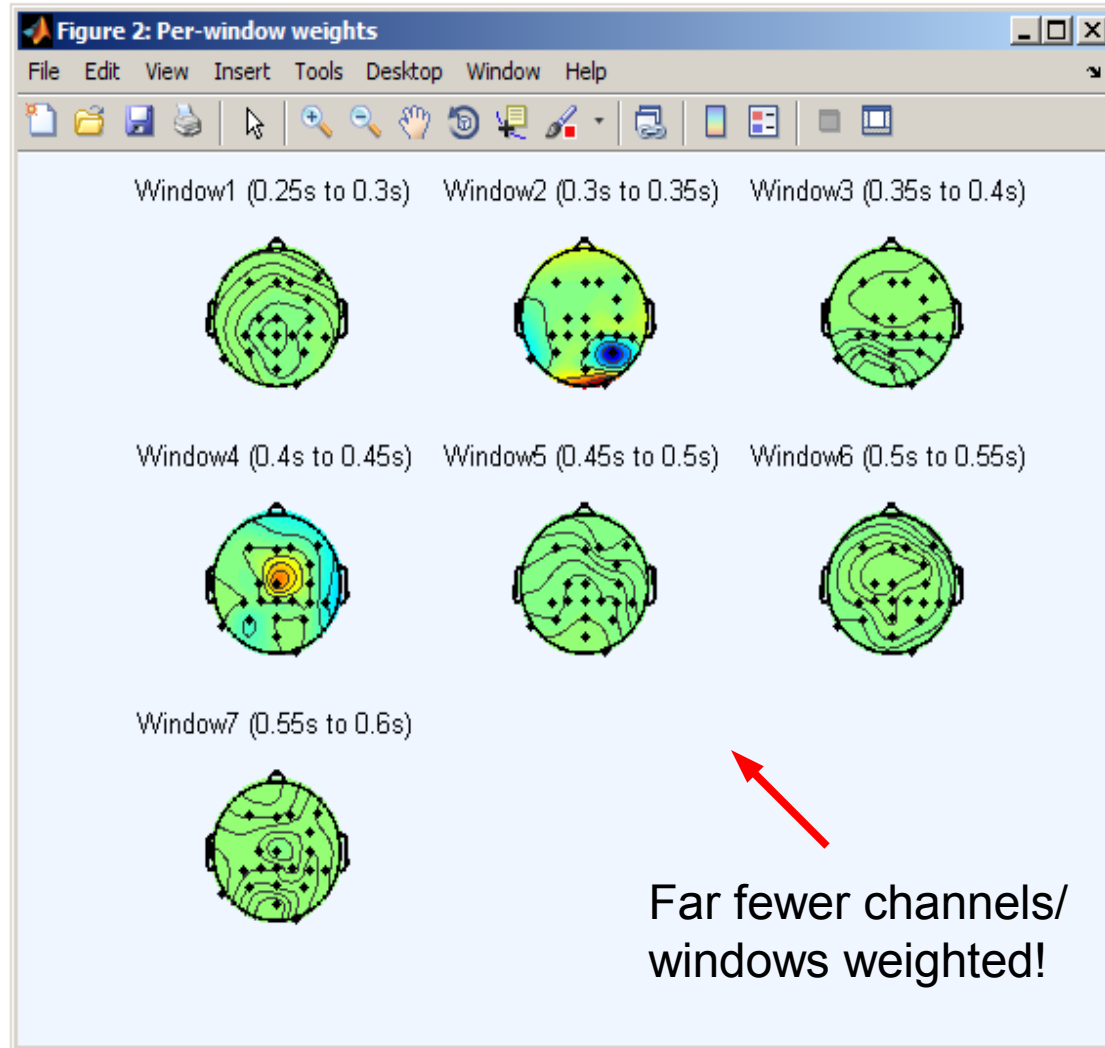


Using a sparse classifier

- Done!
- You selected:
“Variational Bayesian Logistic Regression with Automatic Relevance Determination”
as classifier



Train model, visualize



Train model, visualize

- Sparse classifiers can give you more robust models (fewer channels / sources of errors used), and more interpretable models (only the most relevant features retained)

Scripting

- For analogous scripts, see `userscripts/workshop_script.m`
 - Also contains a 3-class case at the end
- For plugins, see
 - `code/filters/*`
 - `code/machine_learning/*`
 - `code/paradigms/*`
 - `code/online_plugins/*`

Thanks!

Questions?