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

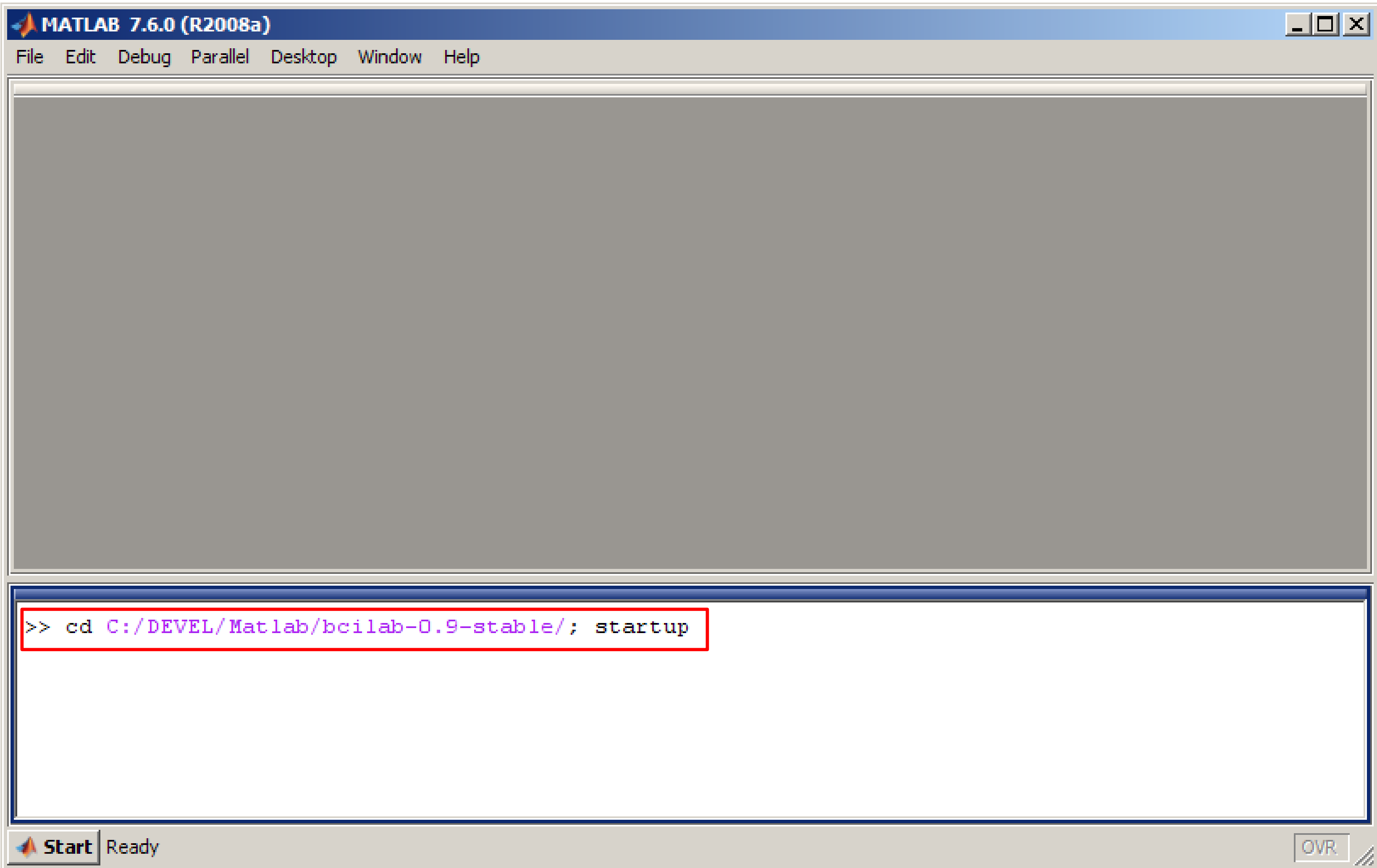
- MATLAB 2008a+ (scripts will run on 7.1+ (2005), but not this version)
- 1GB+ RAM (better: 2GB+)
- Windows, Linux, or Mac
- For smooth workshop: **No** toolboxes in MATLAB path other than Mathworks toolboxes (or EEGLAB)
- To use certain additional features (not covered today): Signal Processing Toolbox,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

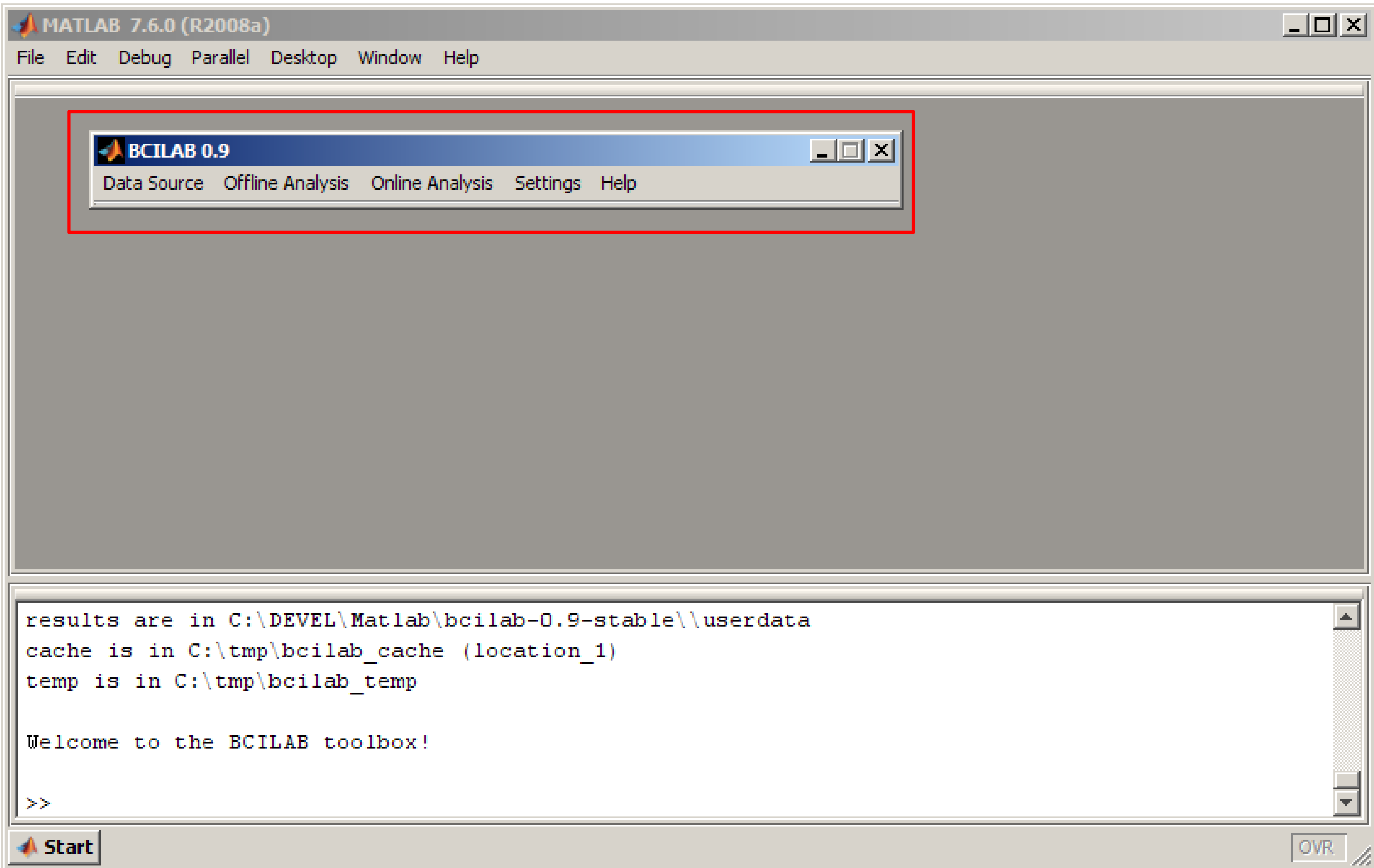
```
cd /your/directory/bcilab-0.9-stable
startup
```

Startup

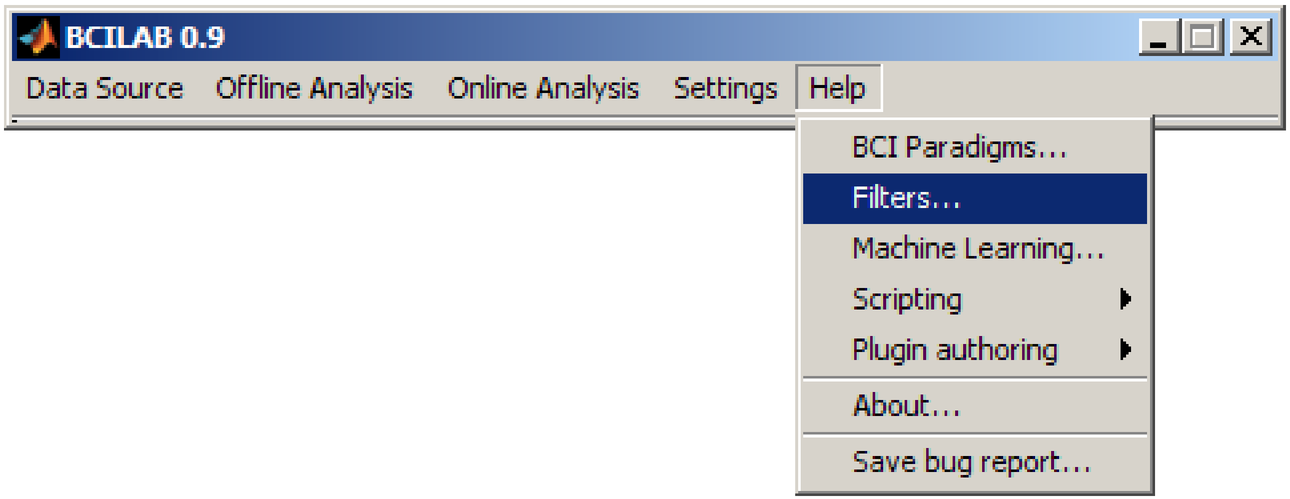


MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help
>> cd C:/DEVEL/Matlab/bcilab-0.9-stable/; startup
Start Ready
OVR

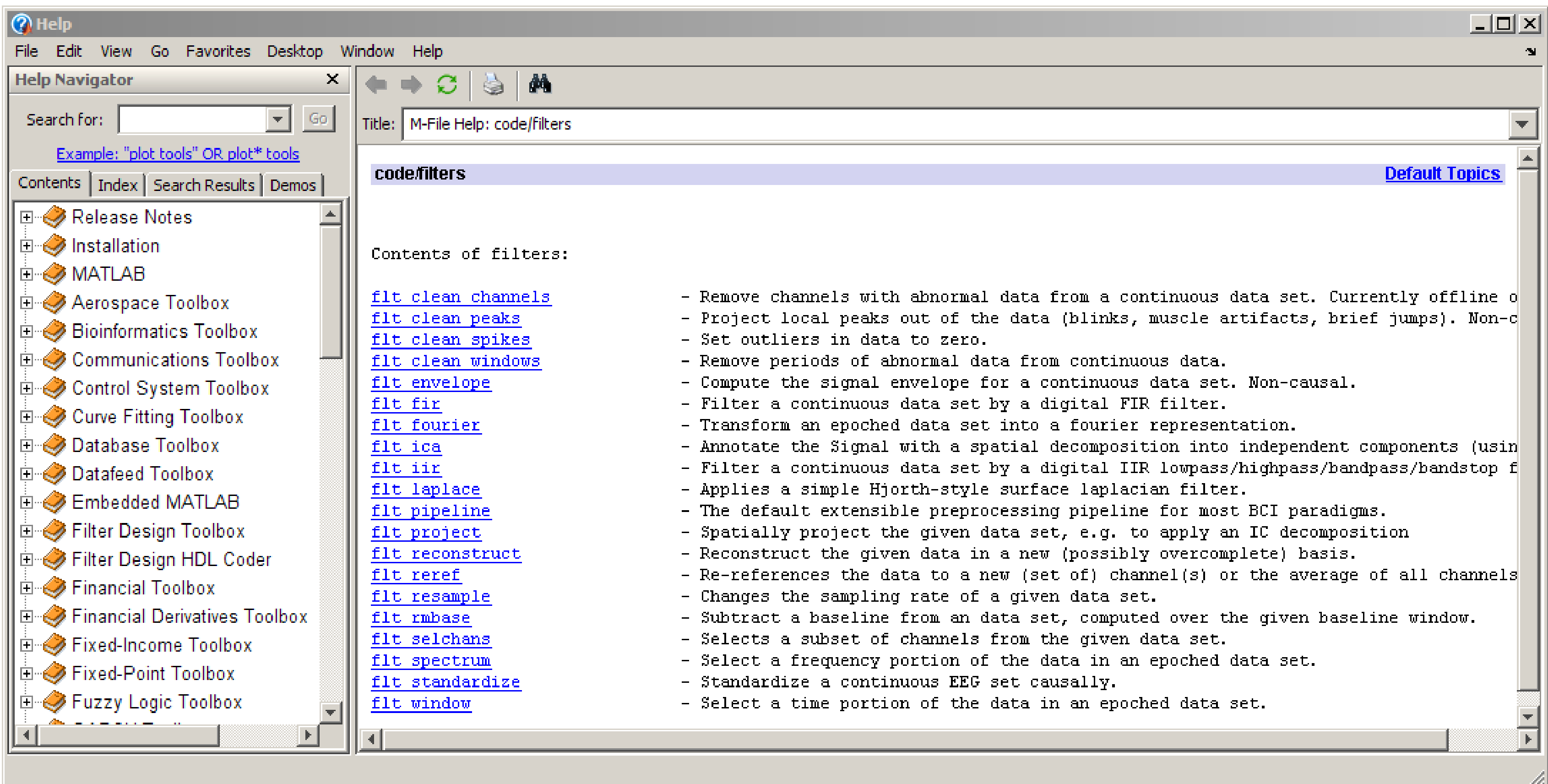
Toolbox GUI


MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help
BCILAB 0.9
Data Source Offline Analysis Online Analysis Settings Help
results are in C:\DEVEL\Matlab\bcilab-0.9-stable\\userdata
cache is in C:\tmp\bcilab_cache (location_1)
temp is in C:\tmp\bcilab_temp
Welcome to the BCILAB toolbox!
>>
Start
OVR

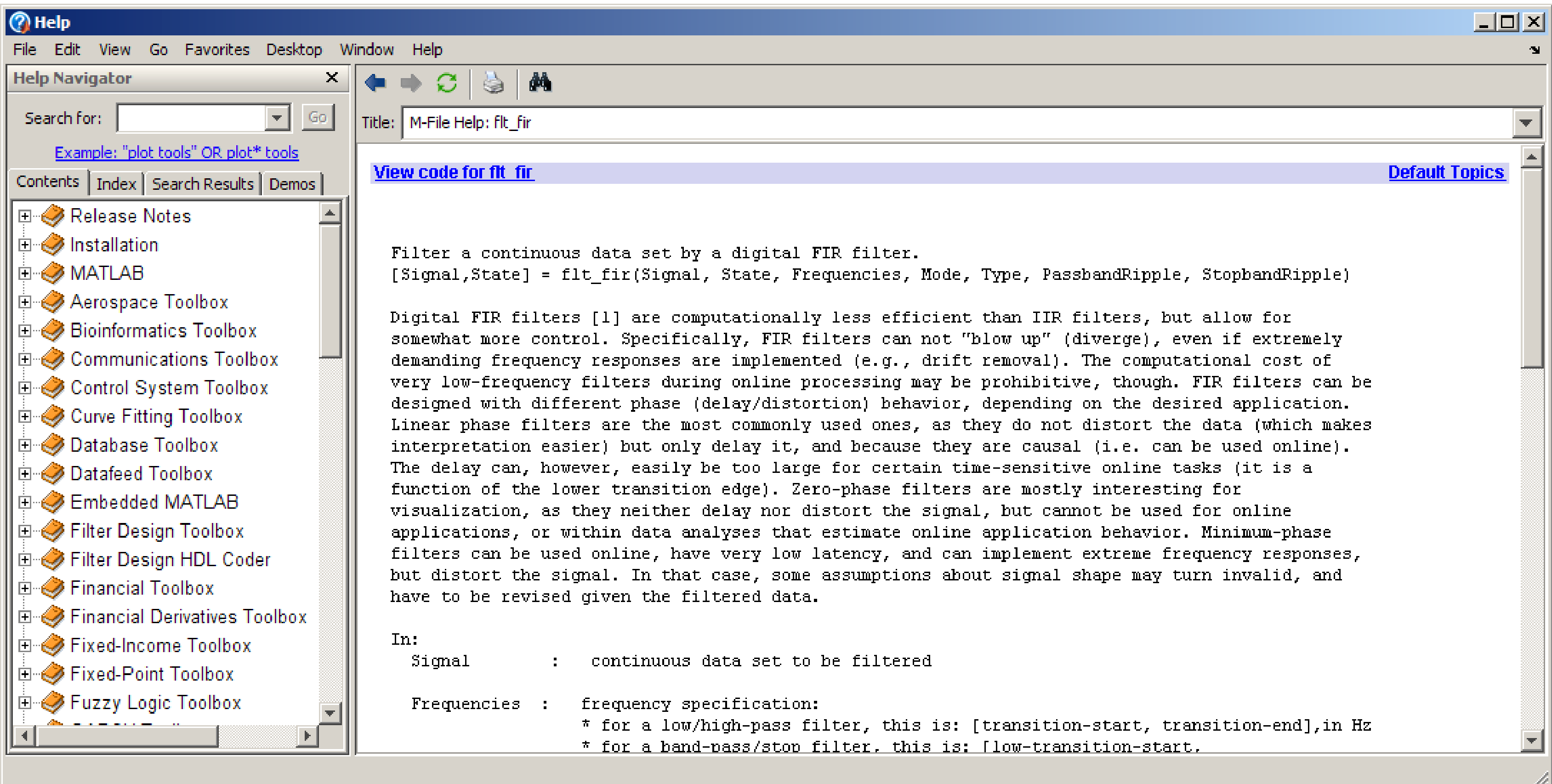
Getting help (if needed)



Getting help (if needed)



Getting help (if needed)



Use case 1

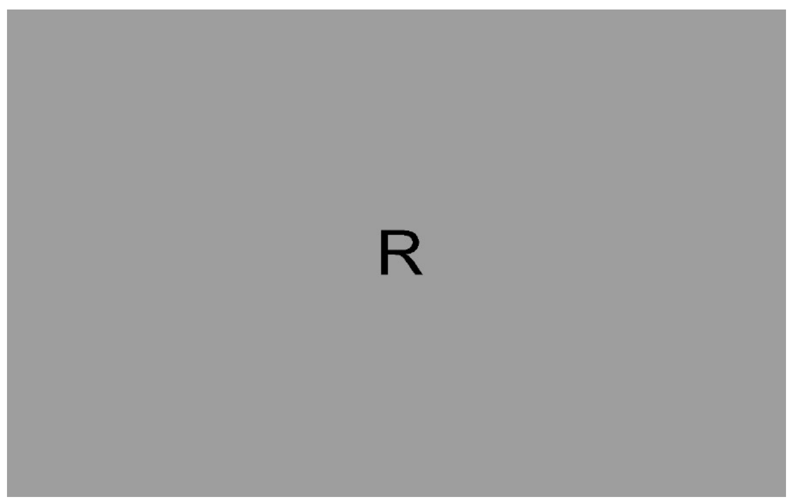
- You just recorded pilot data for some new study
- The idea is to try to estimate a certain aspecty of cognitive state
- The question is what method works best, and what accuracies can be achieved

Use case 1

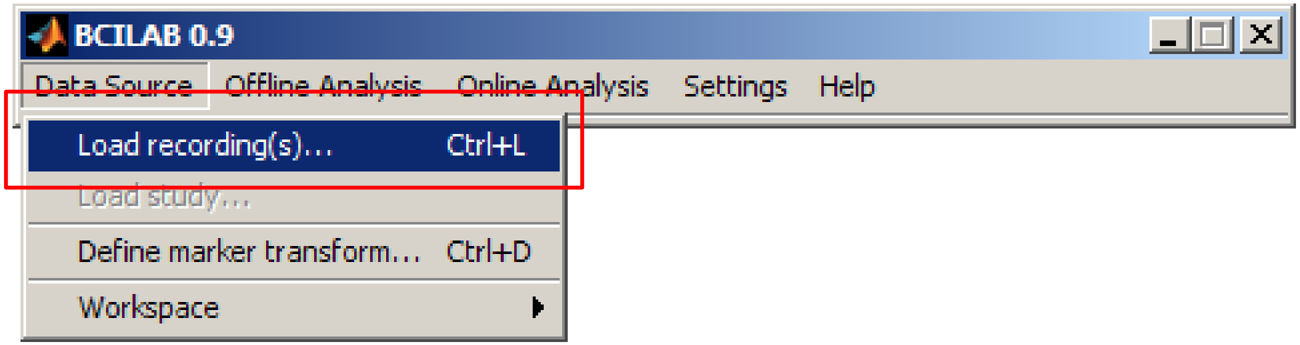
- Scenario: Subject is instructed to imagine a hand movement, either **left** hand or **right** hand (standard BCI case)
- Task: Estimate, from raw data, which hand movement was imagined
- Experimental data: EEG, 32 channels, 2 sessions (each ~30 min.), 2 sub-blocks per session with intermittent pause

Experimental task

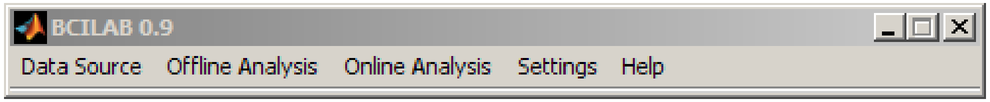
- 160 trials
- Randomized Instruction: L or R
- Displayed for 3s, followed by blank screen for 3.5s
- Sample:

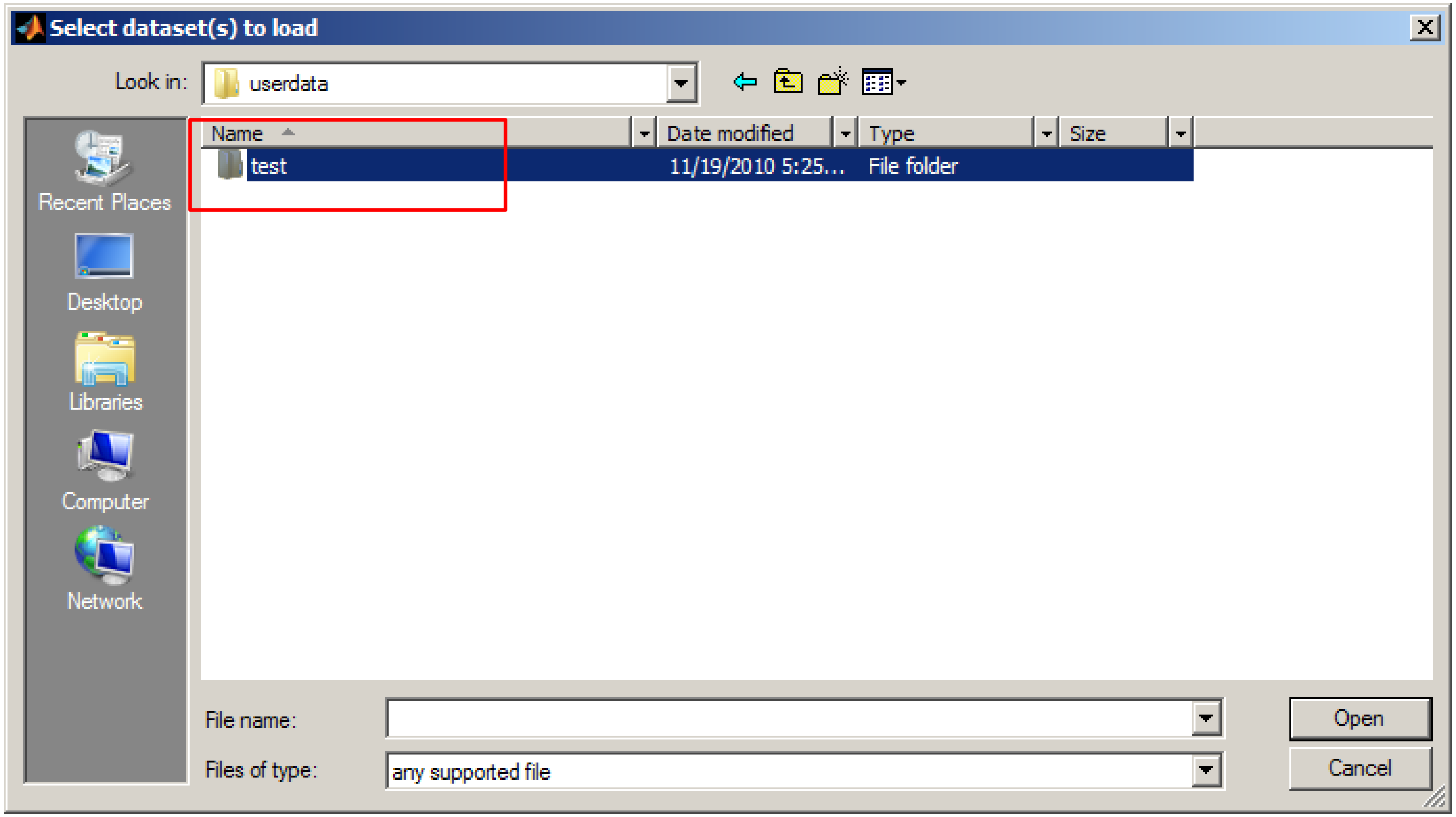


Load the data

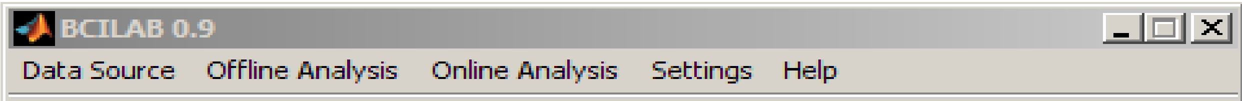


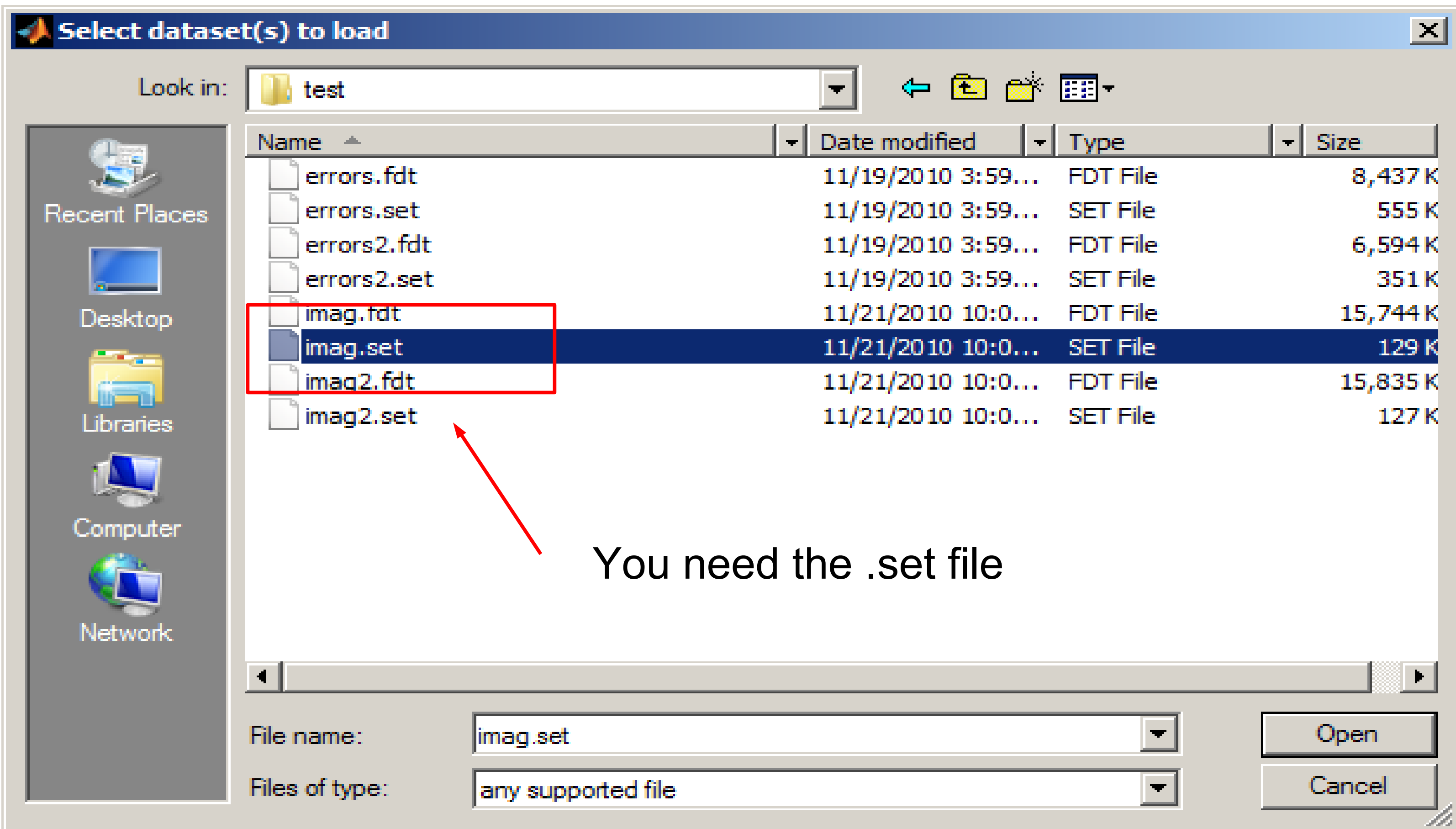
Load the data



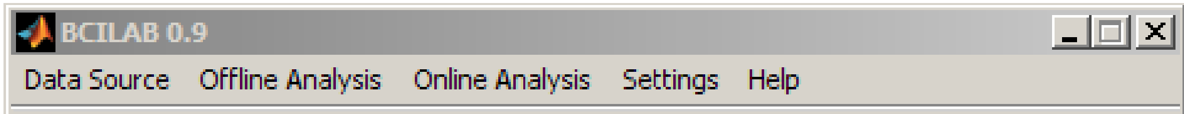


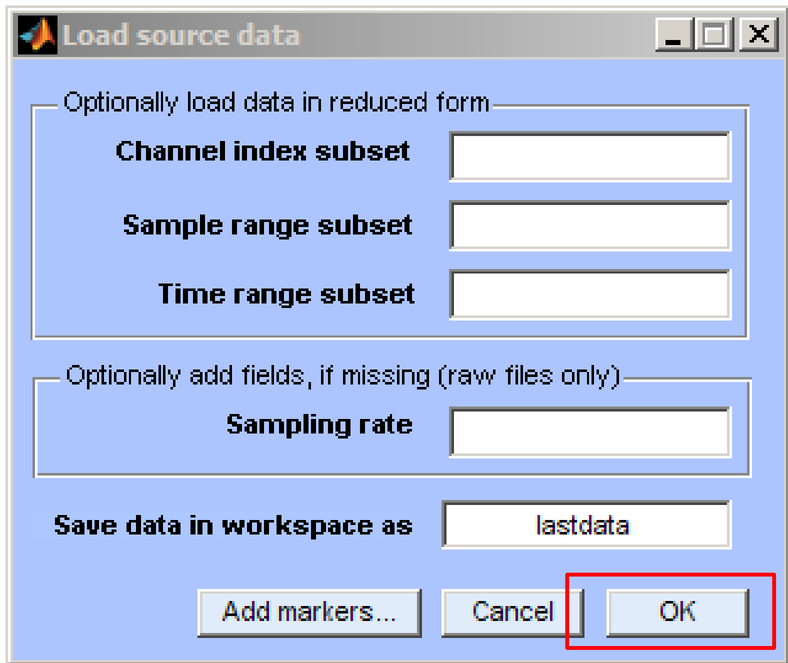
Load the data





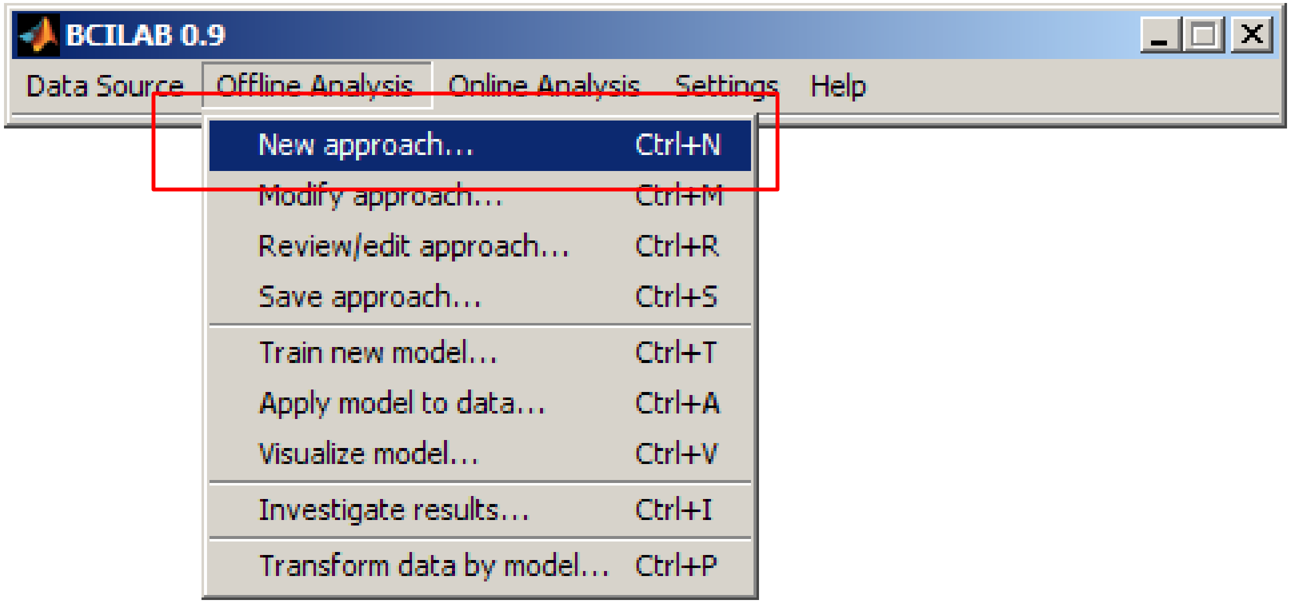
Load the data





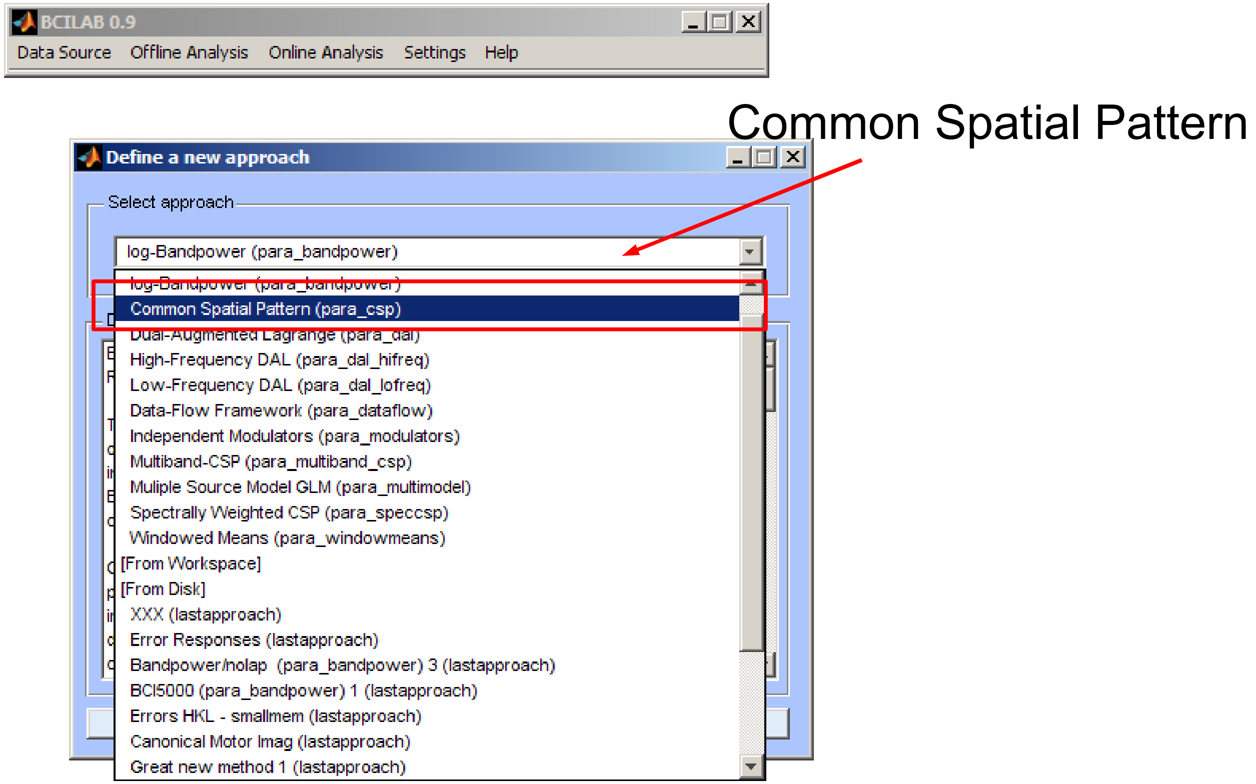
Define an approach

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



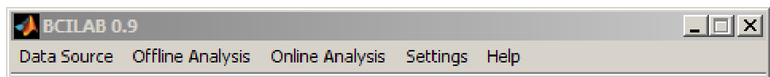
Define an approach

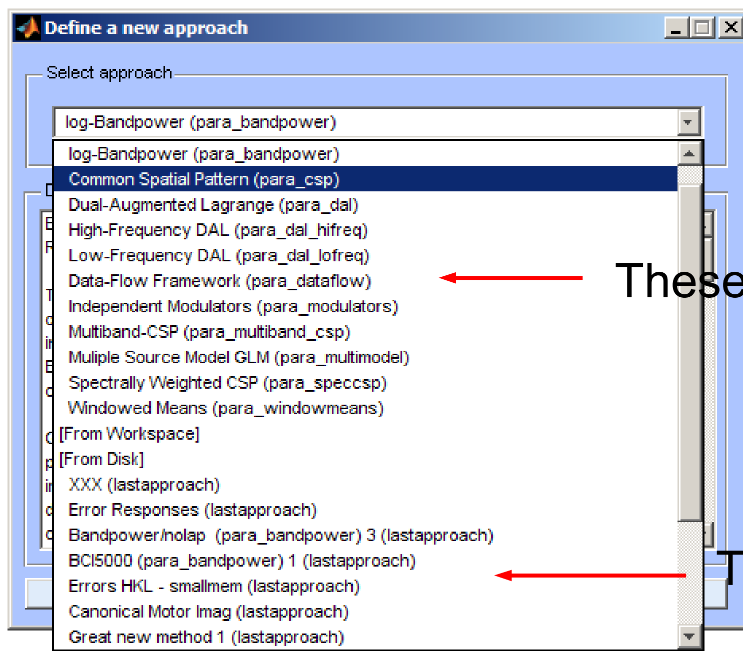
- You never start completely from scratch, but on the basis of what is known to work



Define an approach

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

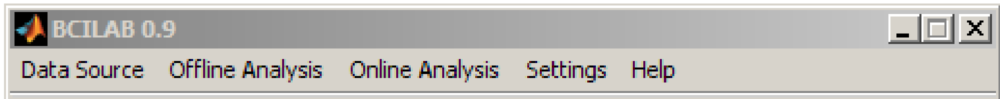


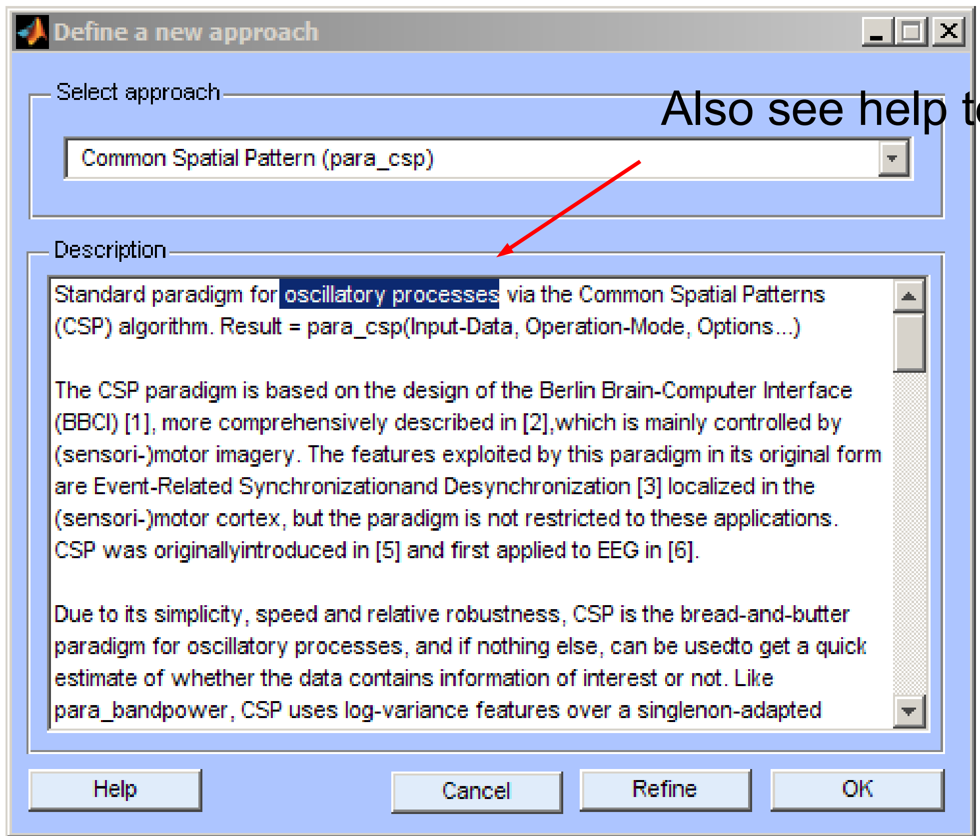


These are pre-defined templates

These are user-defined

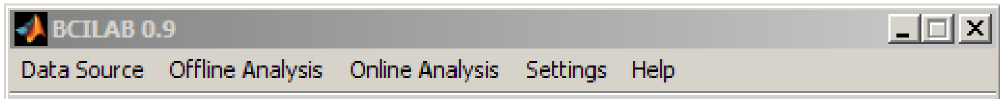
Define an approach

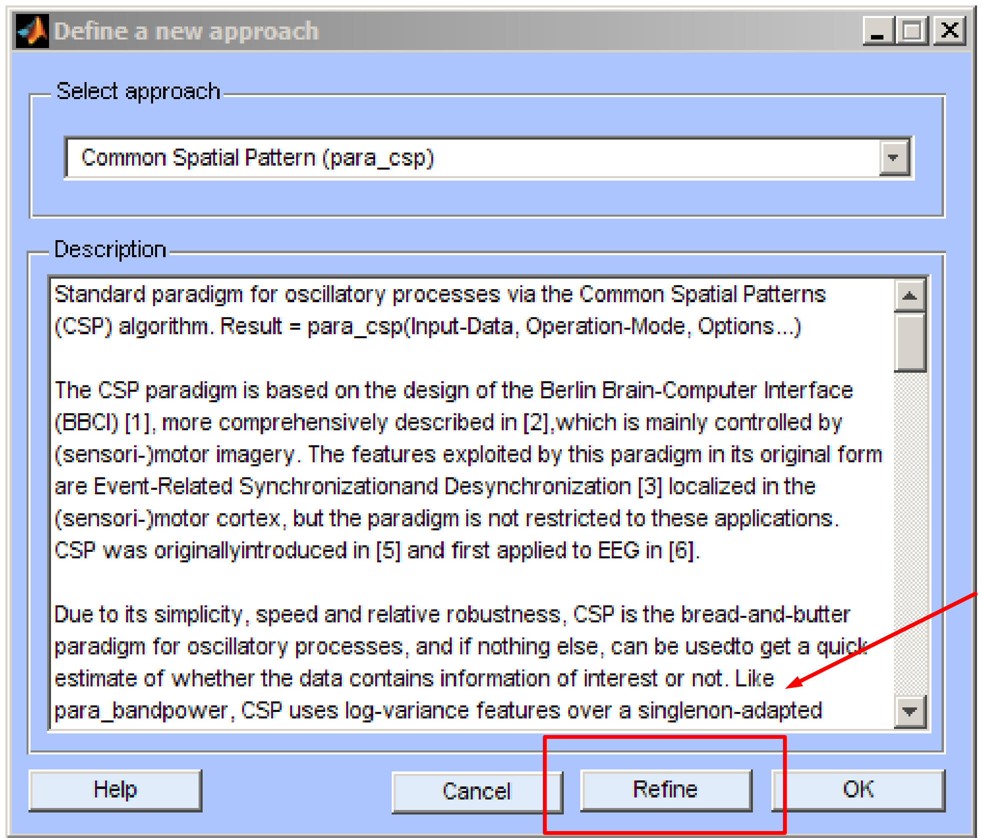




Also see help text

Define an approach

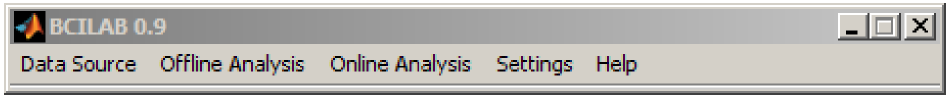


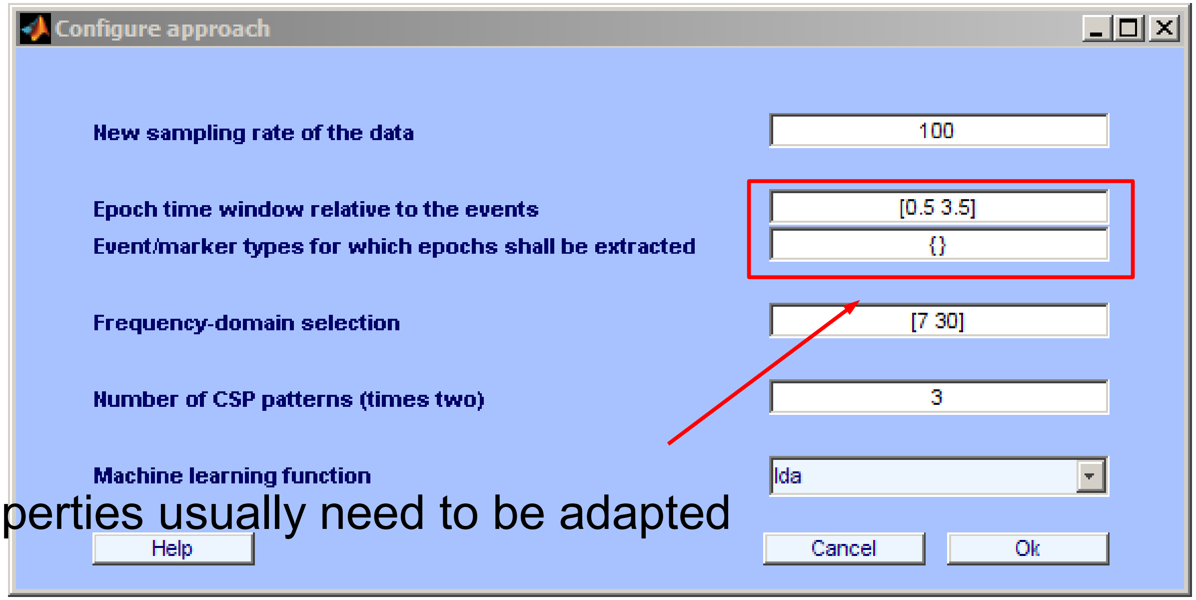


Adapt the template to your experiment

Configure approach

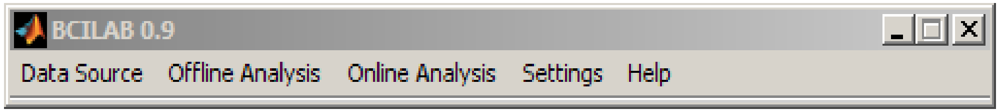
- Key properties can be configured in this dialog

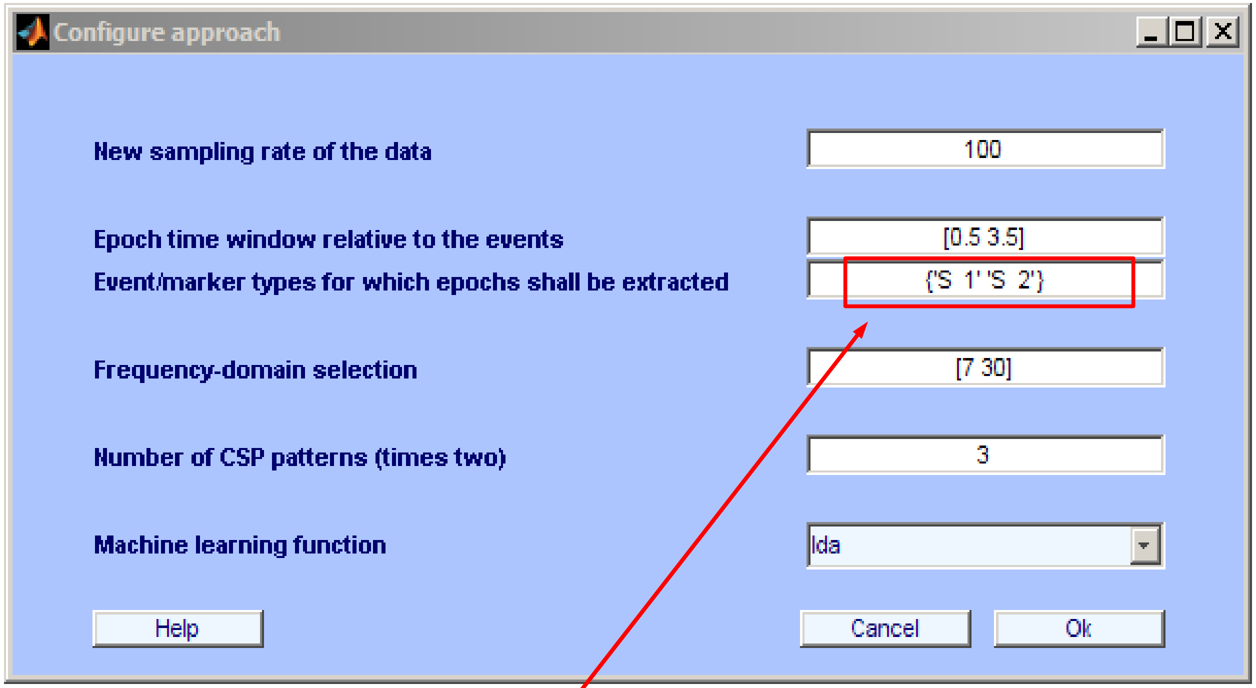




Trial epoch properties usually need to be adapted

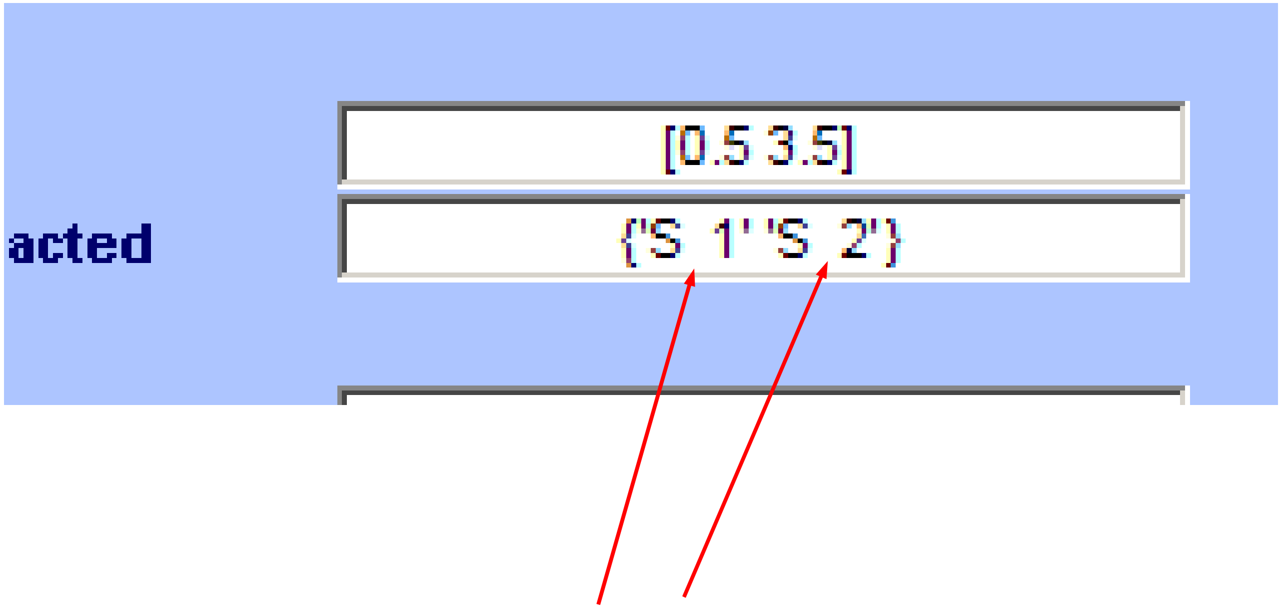
Configure approach





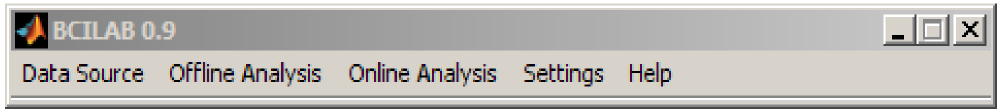
Fill in the 2 event types for this dataset; Stimulus 1 & 2, called 'S 1' and 'S 2' (Brain Products names)

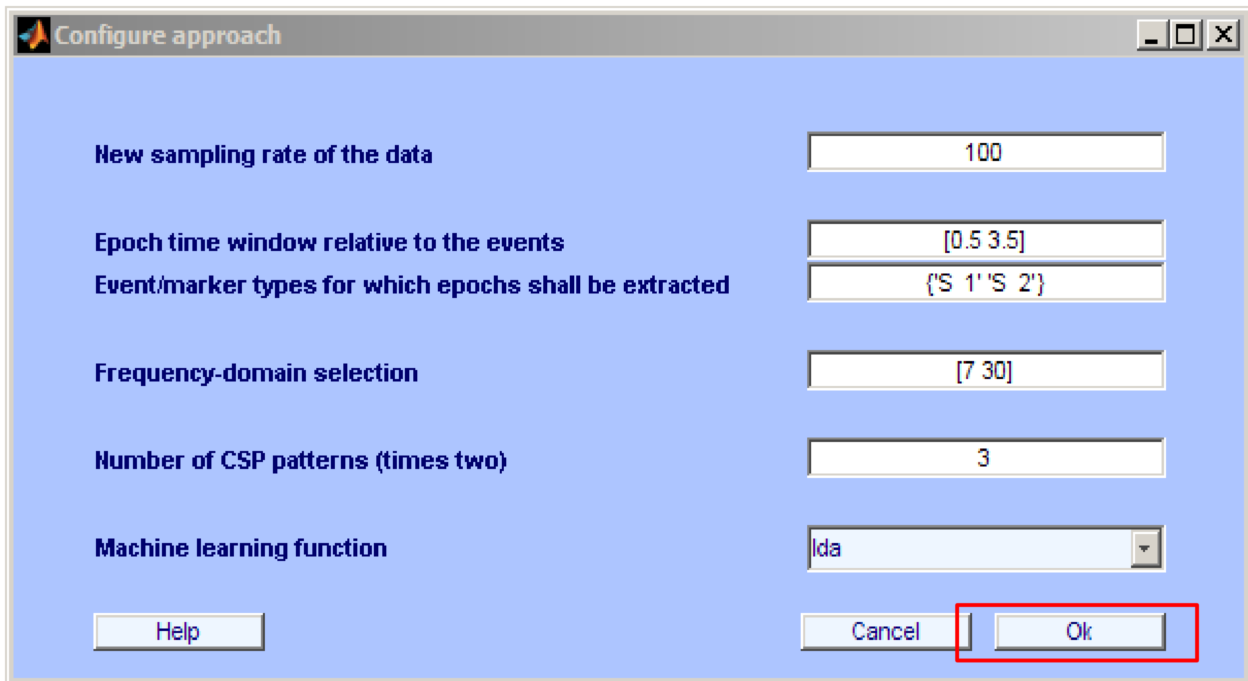
Configure approach



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

Configure approach

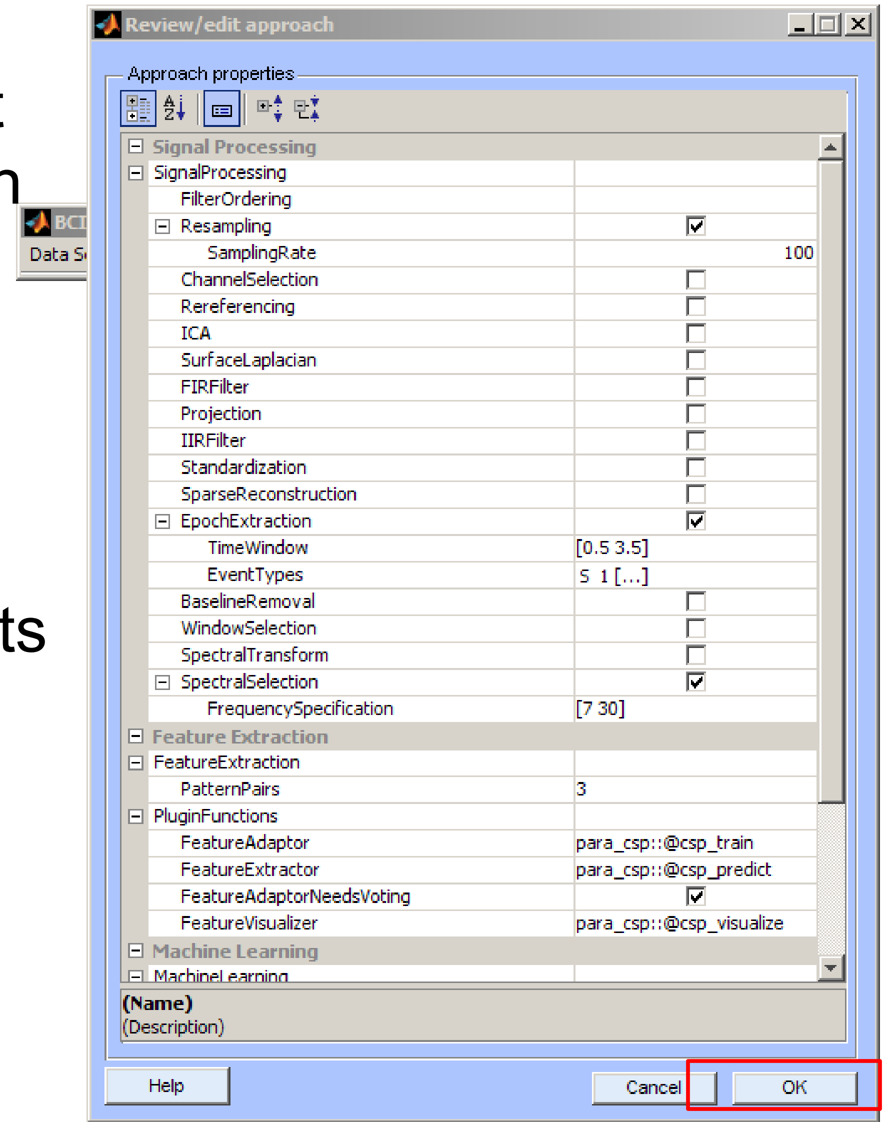




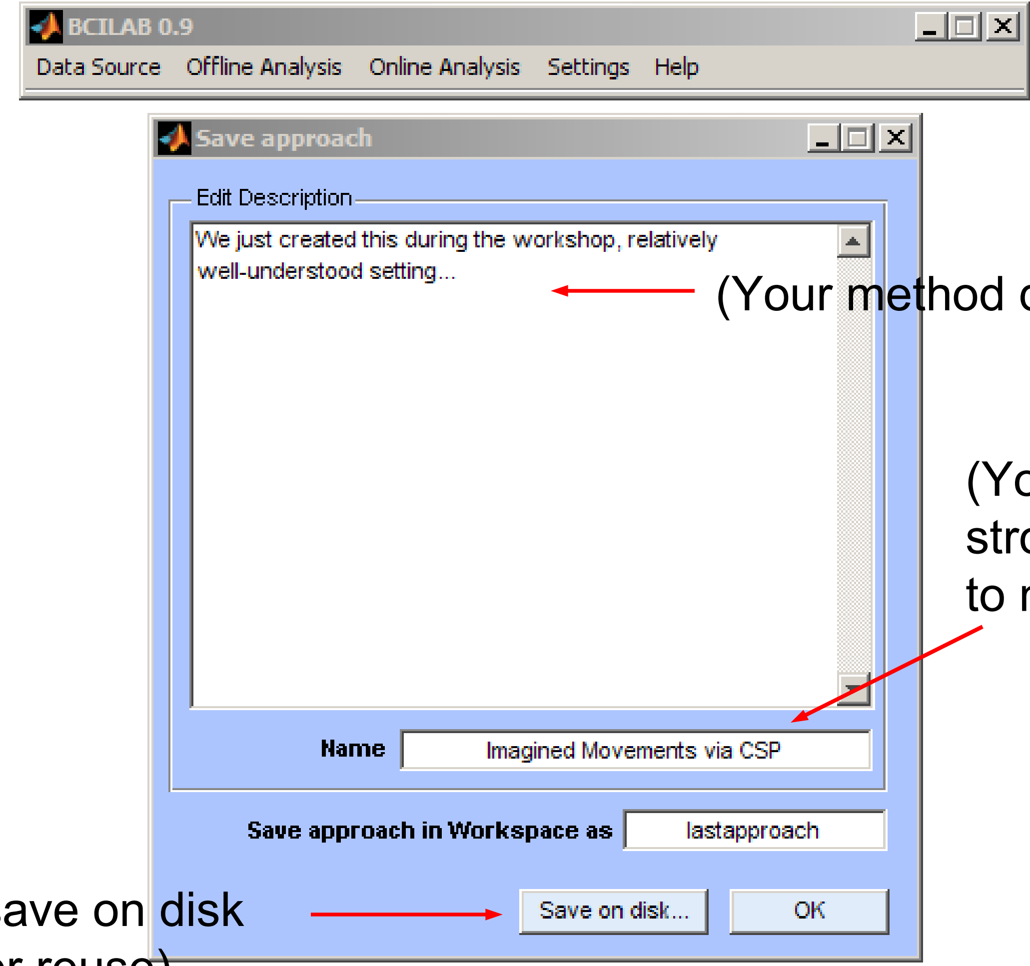
(Takes a while after clicking OK)

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

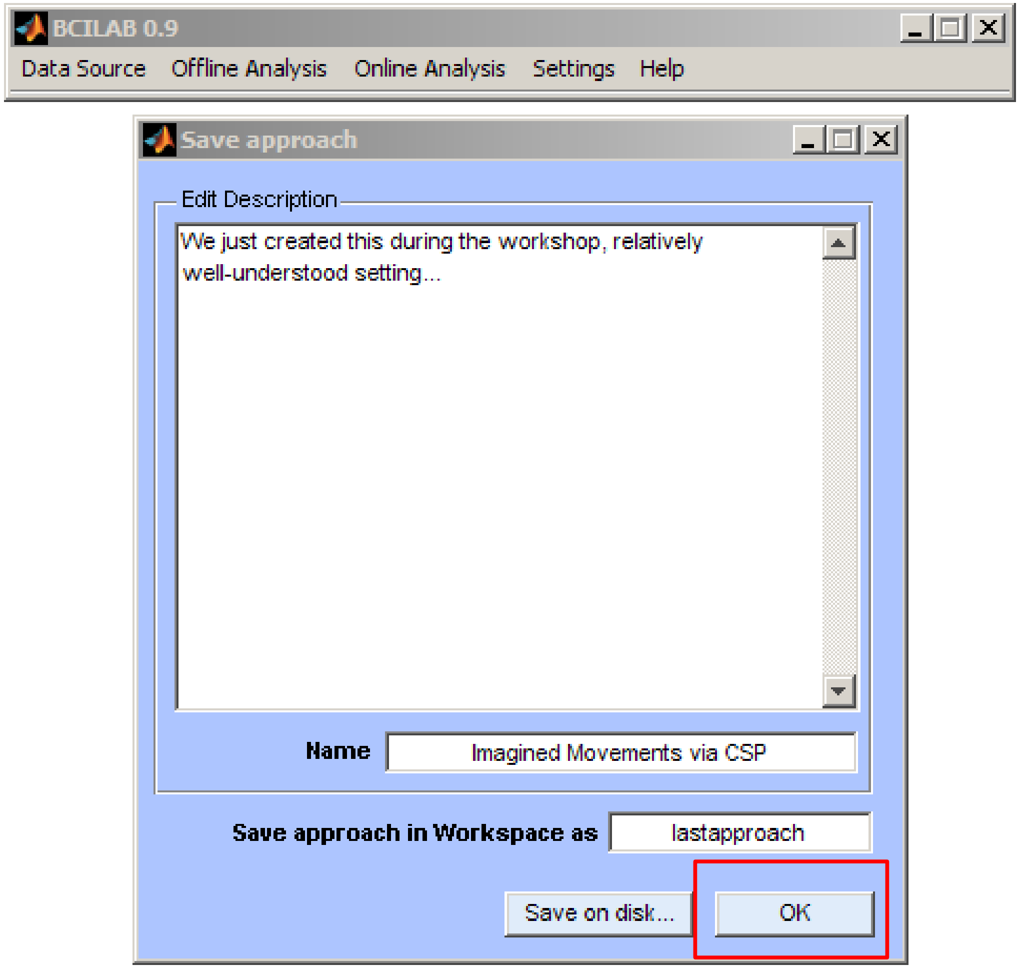


(Your method description)

(Your executive summary
strongly recommended
to name this properly)

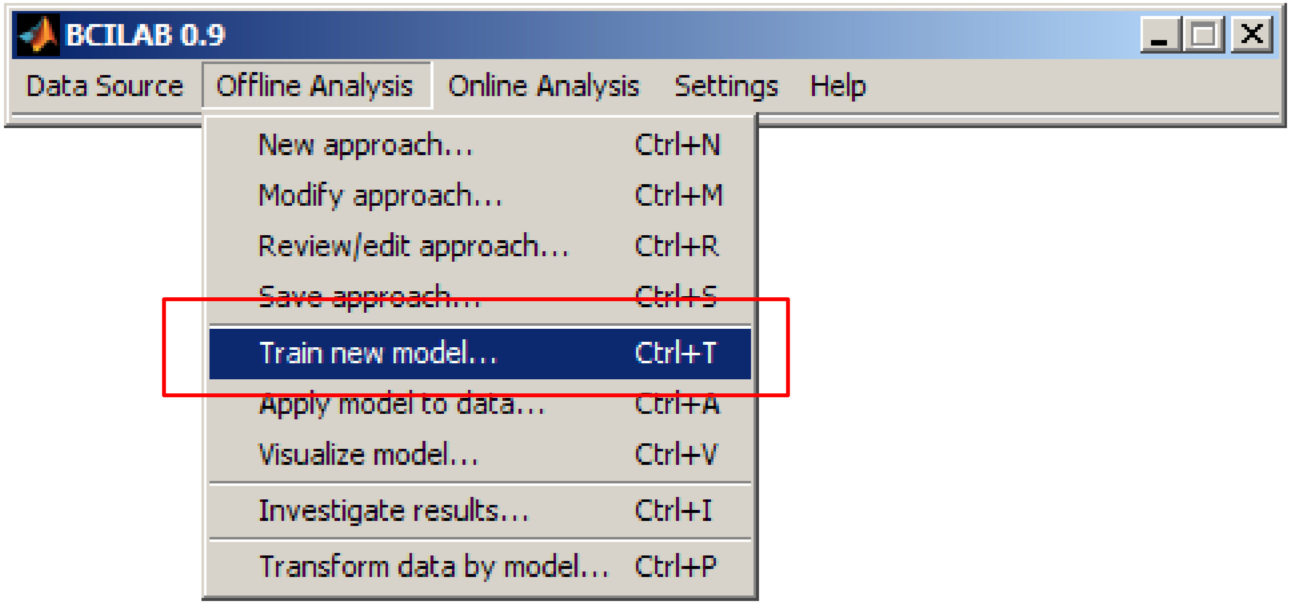
(You can save on disk
for later reuse)

Save approach

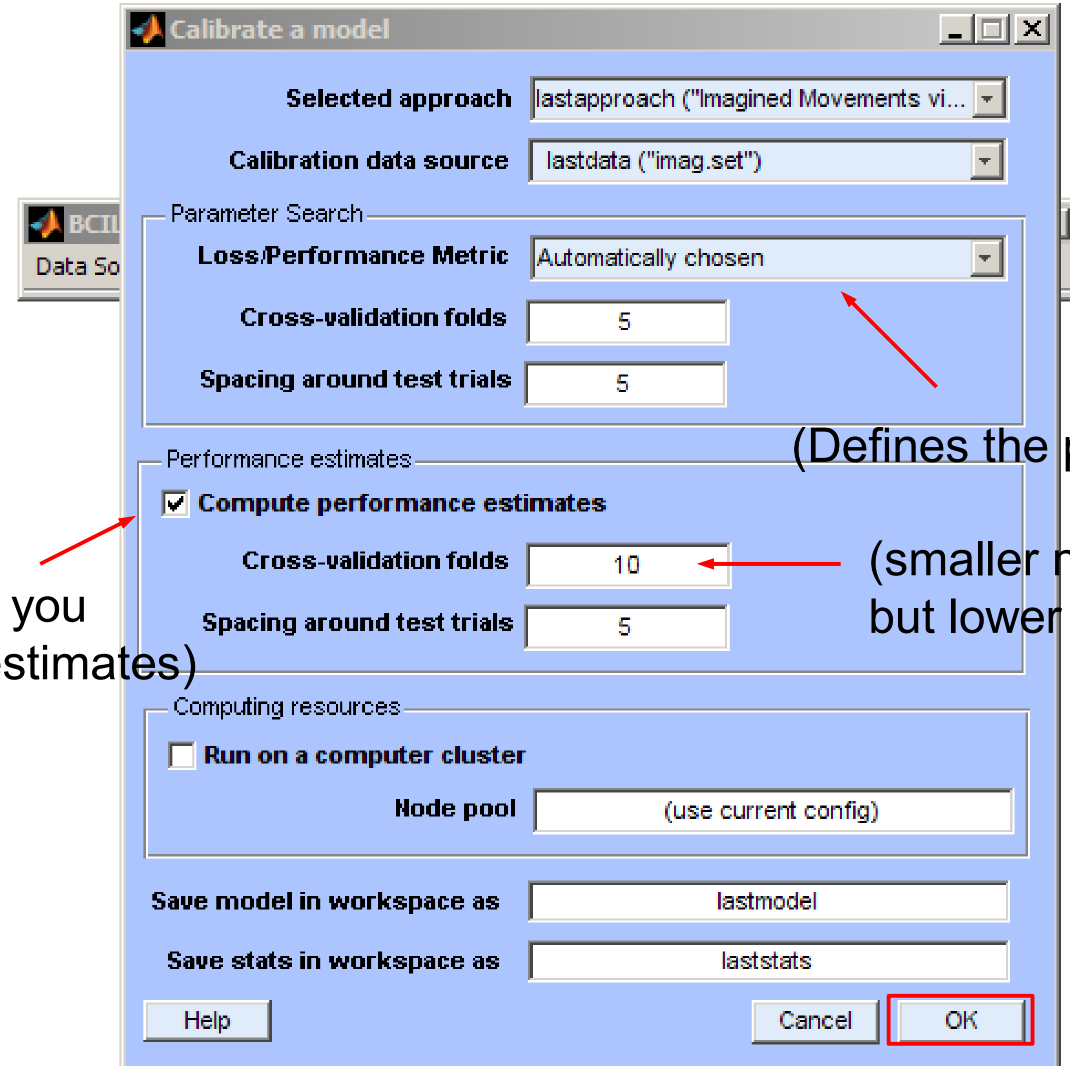


Learn a predictive model

- Put the method to the test...



Learn a predictive model

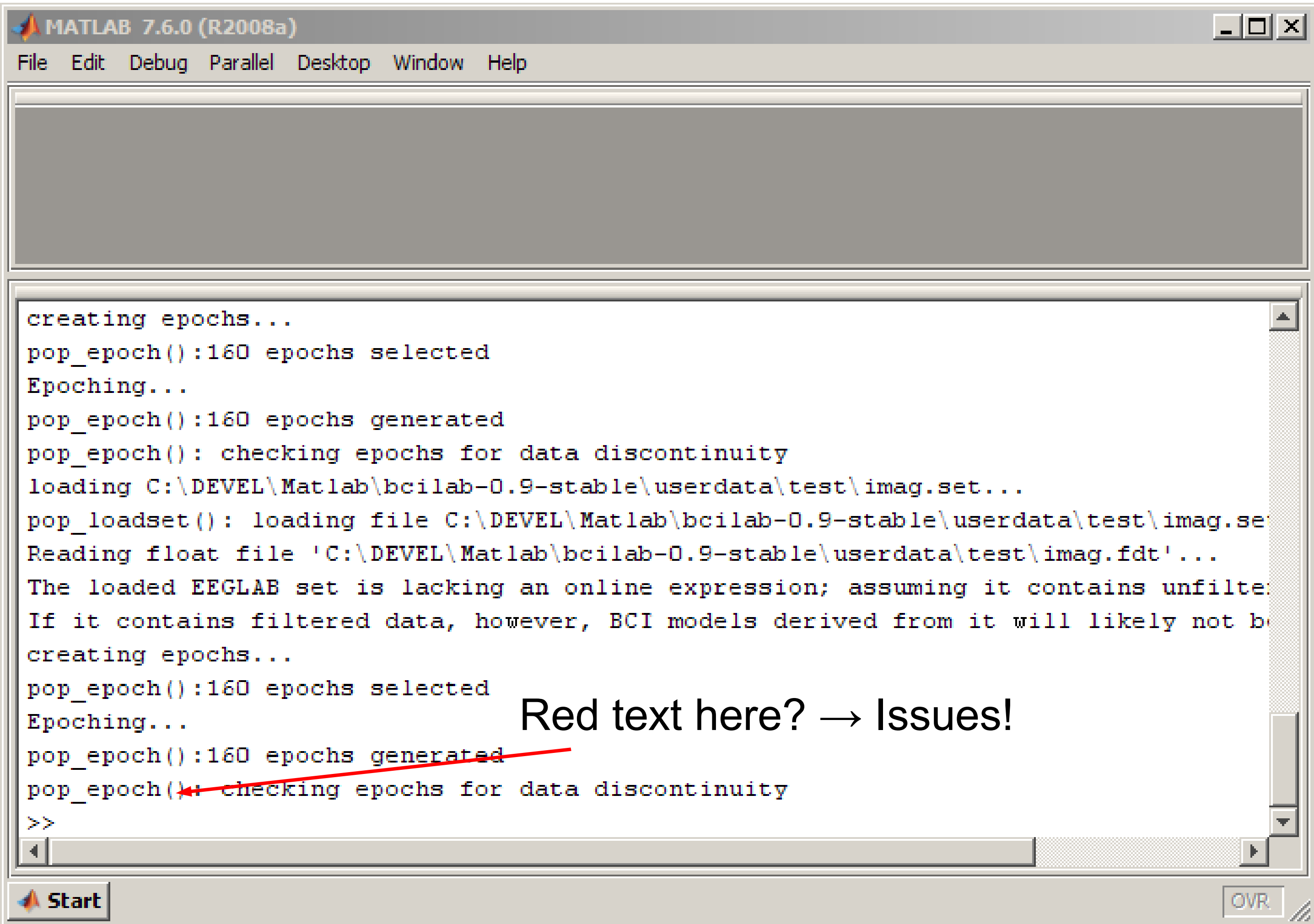


(Defines the performance metric)

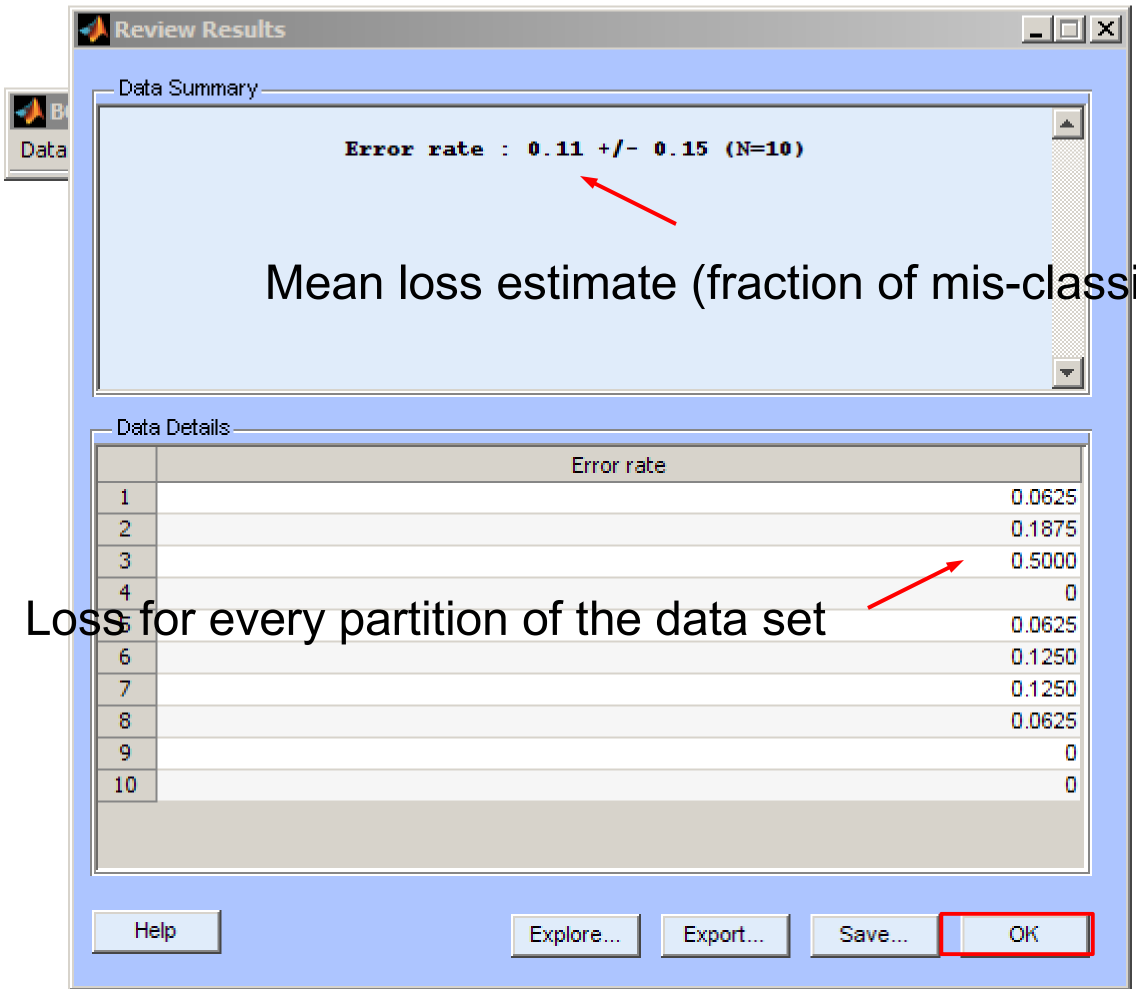
(If this is checked, you get performance estimates)

(smaller number: faster, but lower quality estimates)

Wait for results



Review results



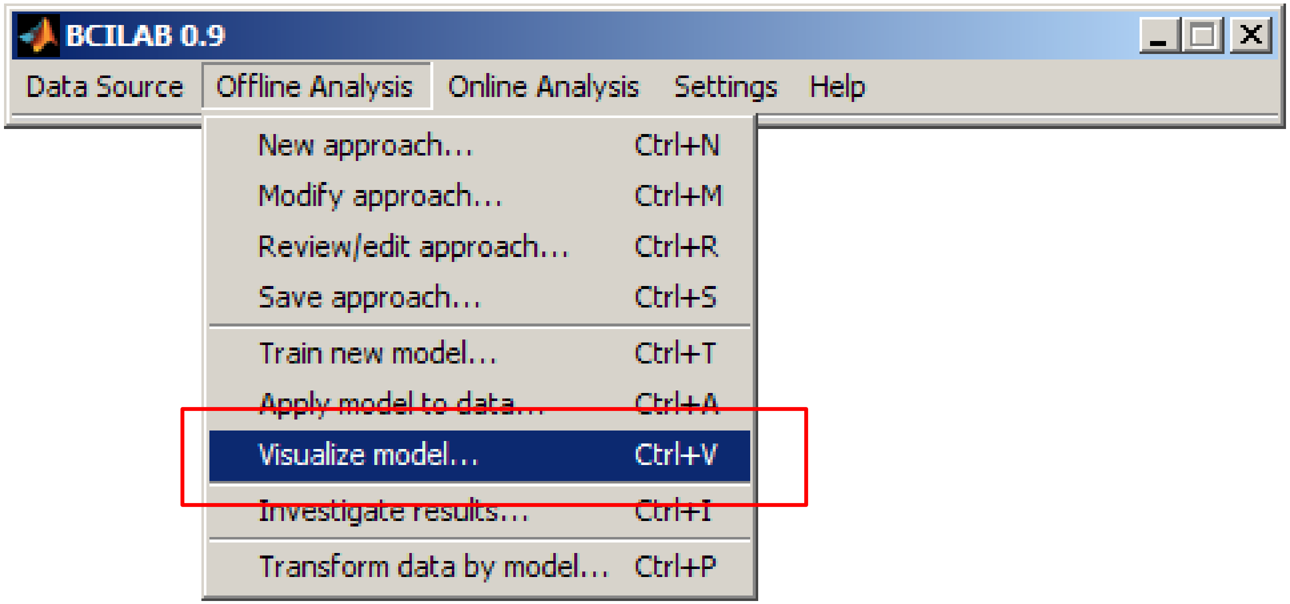
Mean loss estimate (fraction of mis-classified trials)

Loss for every partition of the data set

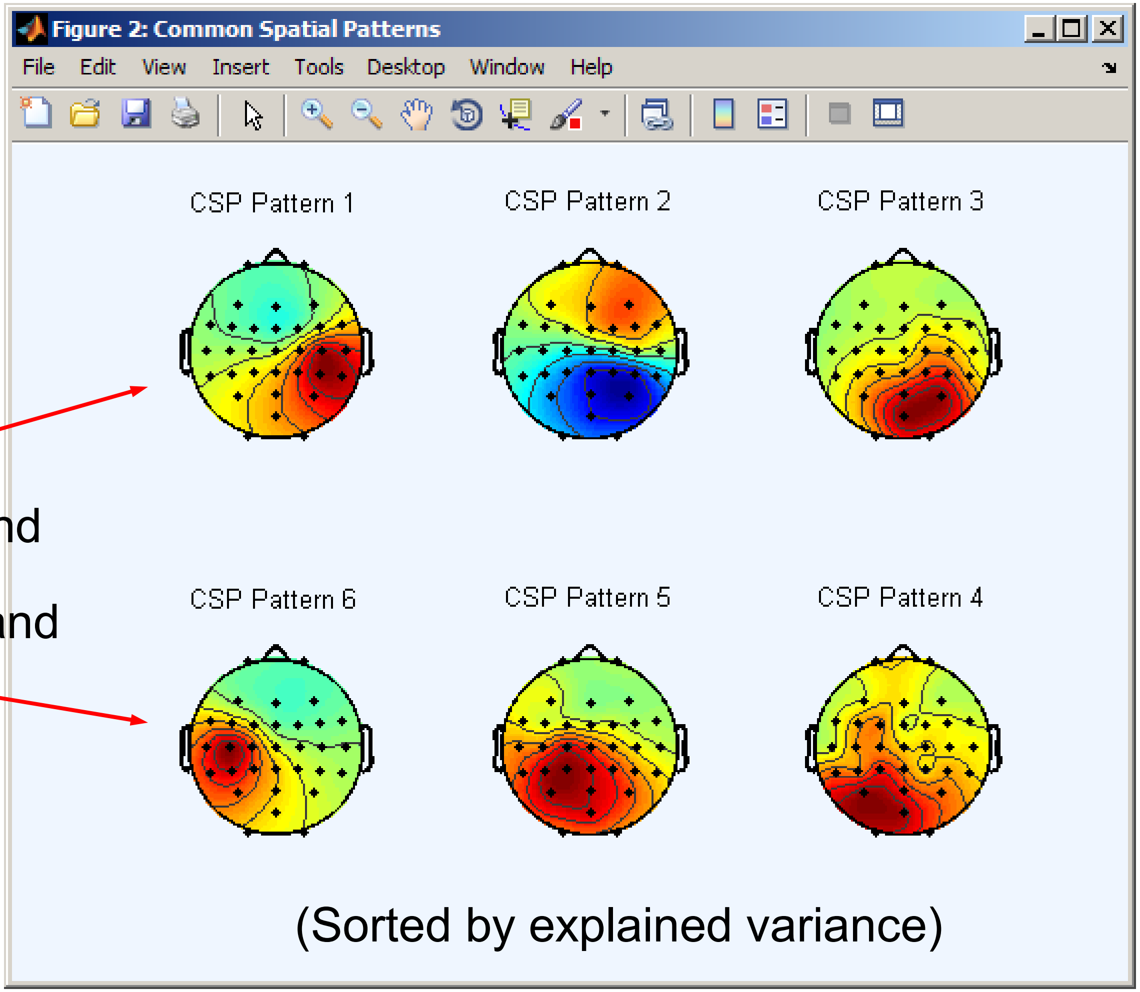
Review results

- 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

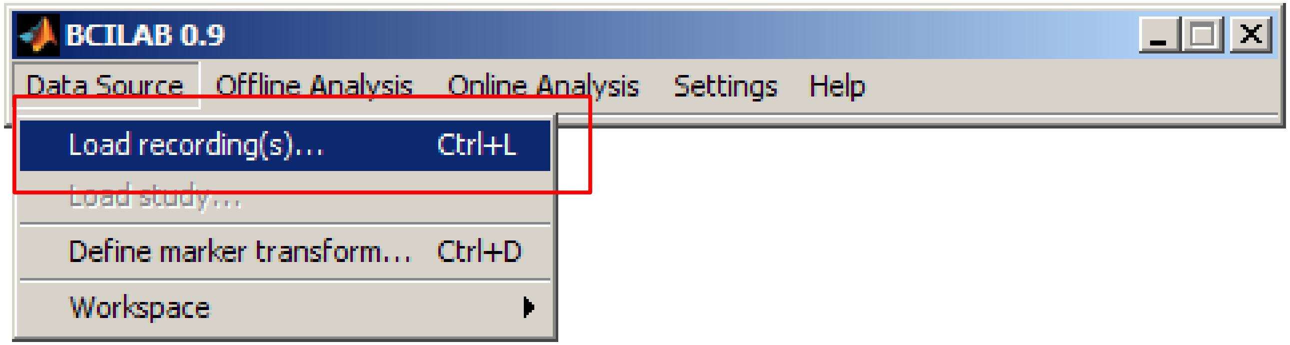


Motor cortex dipoles, left hand

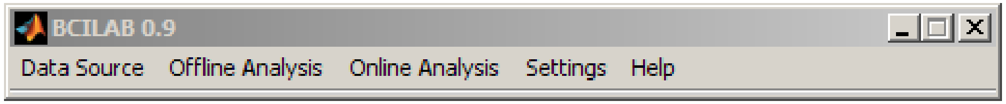
Motor cortex dipoles, right hand

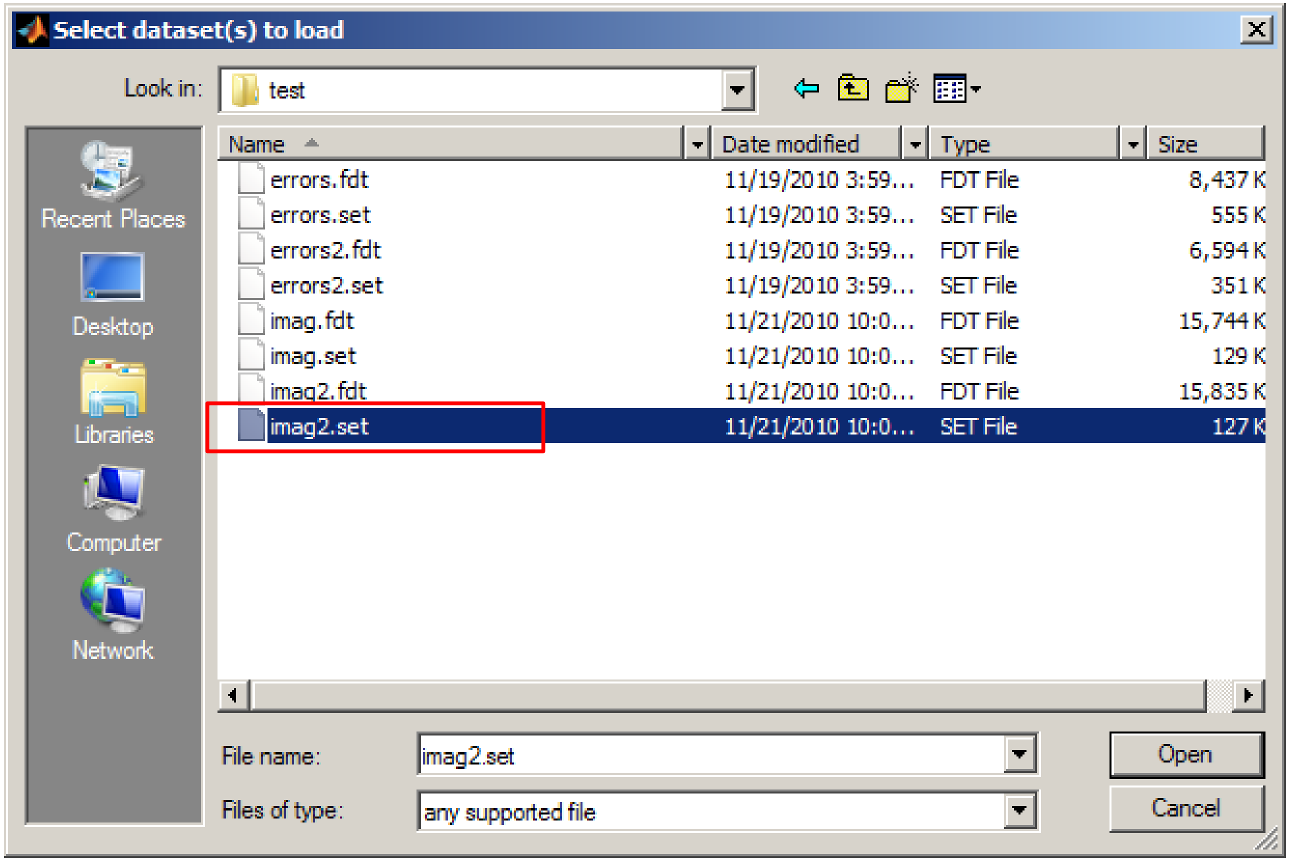
(Sorted by explained variance)

Apply model to 2nd session

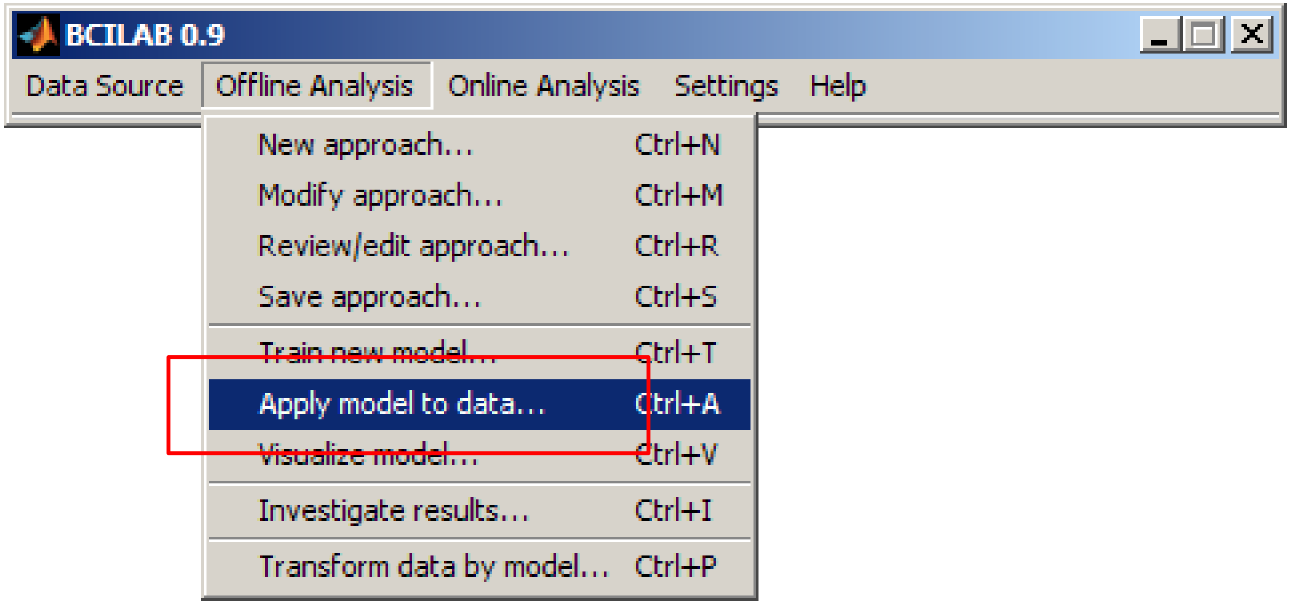


Apply model to 2nd session

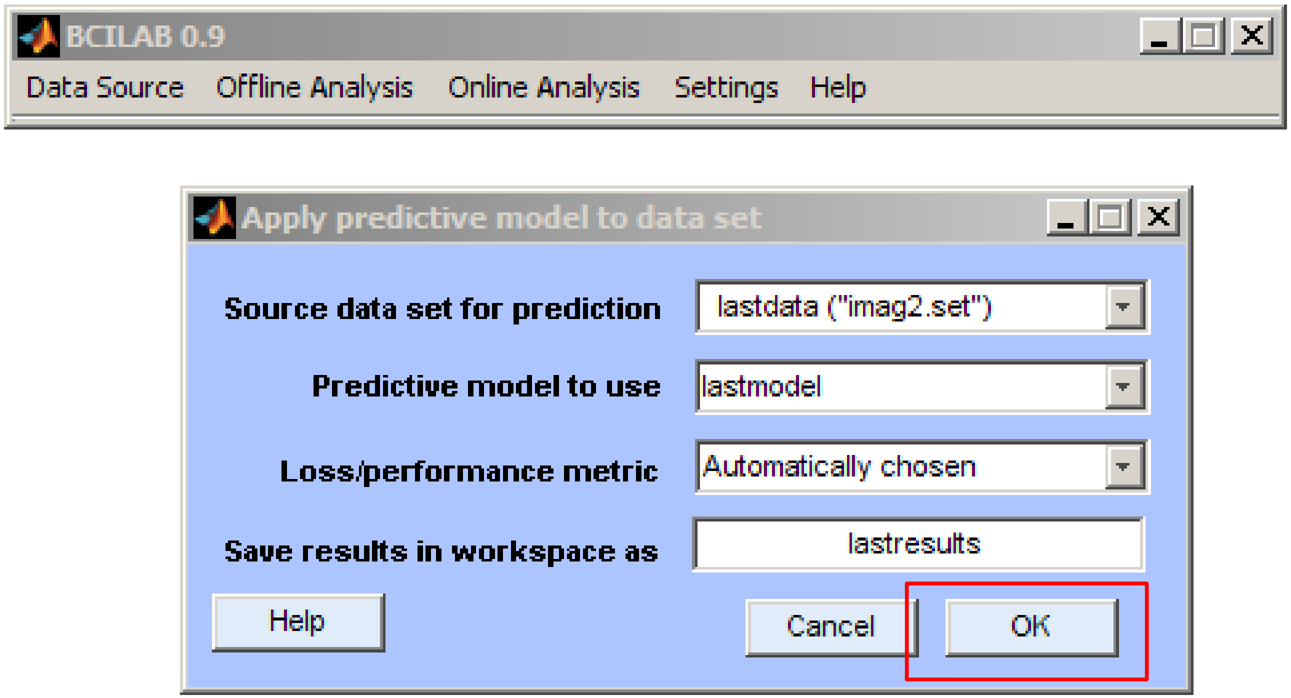




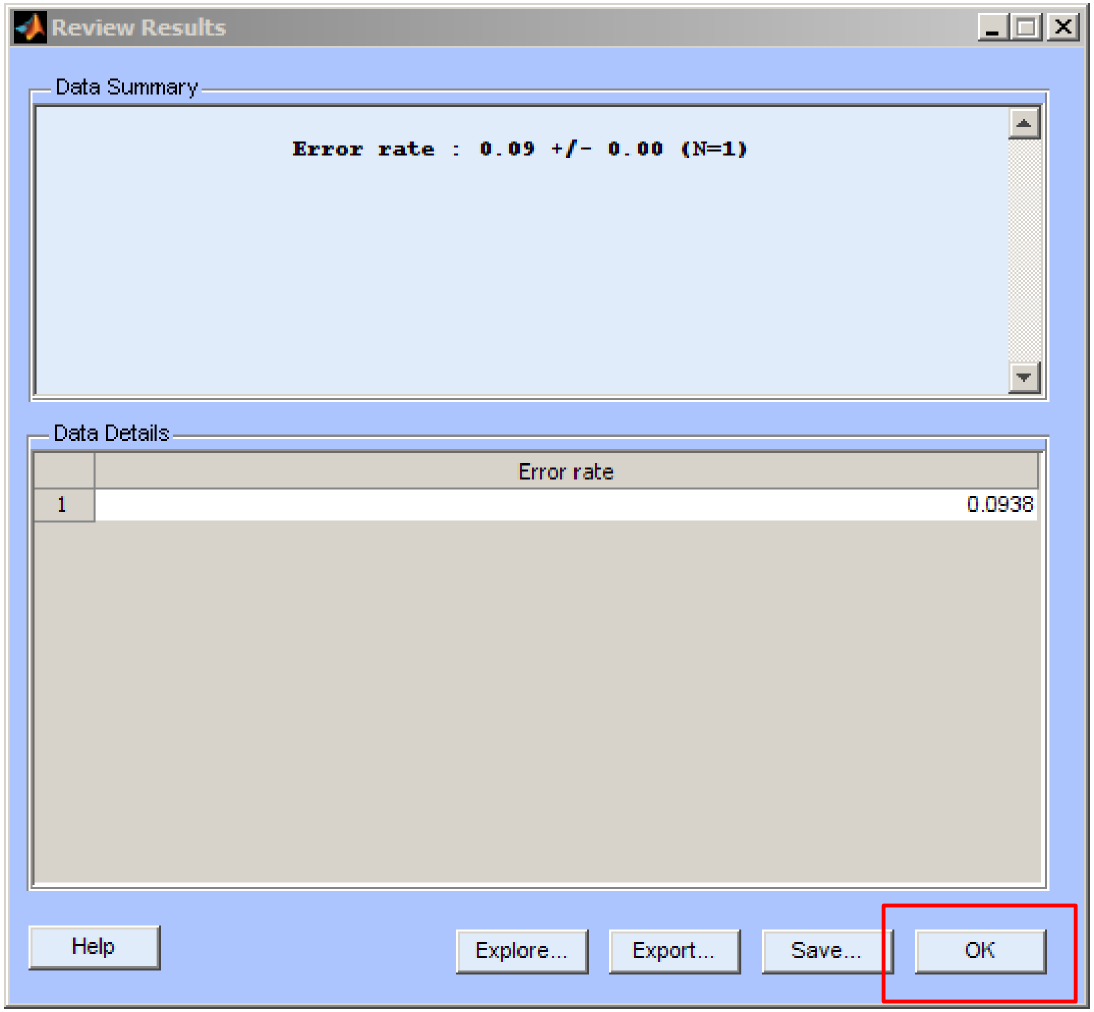
Apply model to 2nd session



Apply model to 2nd session

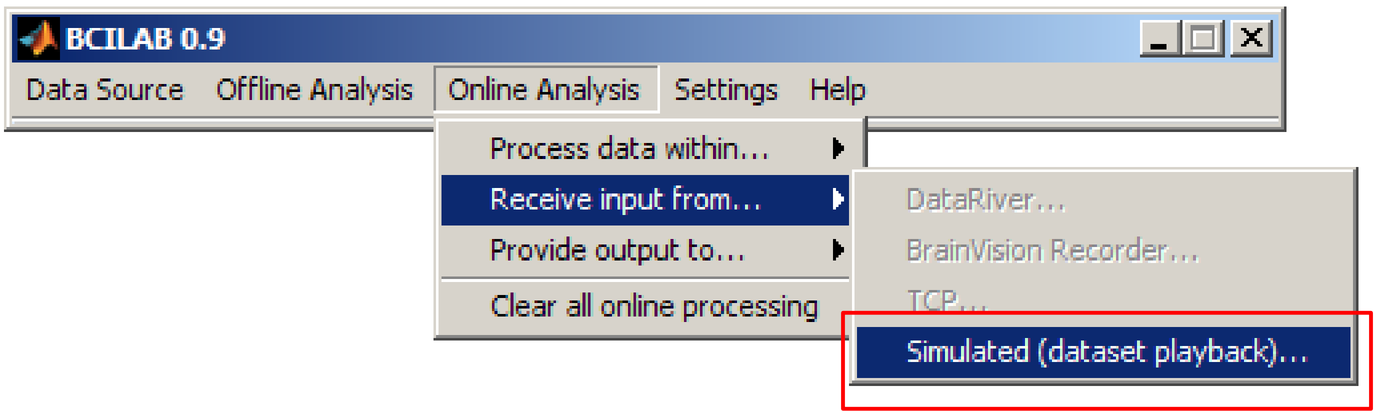


Review results



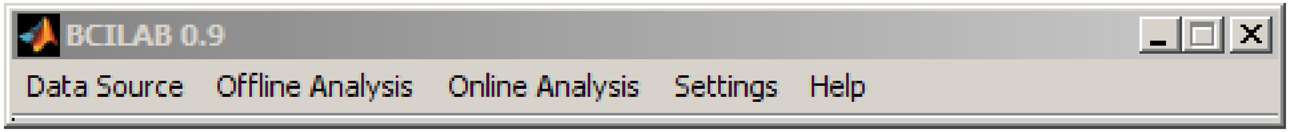
OR: Apply model online

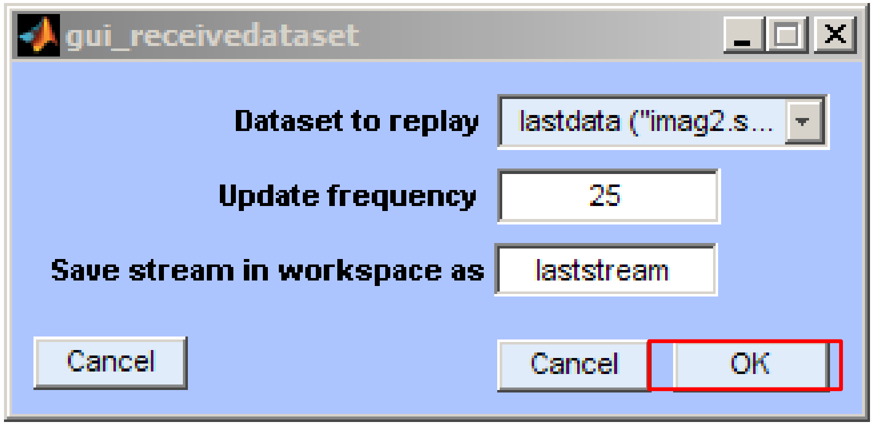
- (if you have a subject sitting next to you)
- Today: use a simulated data source (playing back the 2nd session)



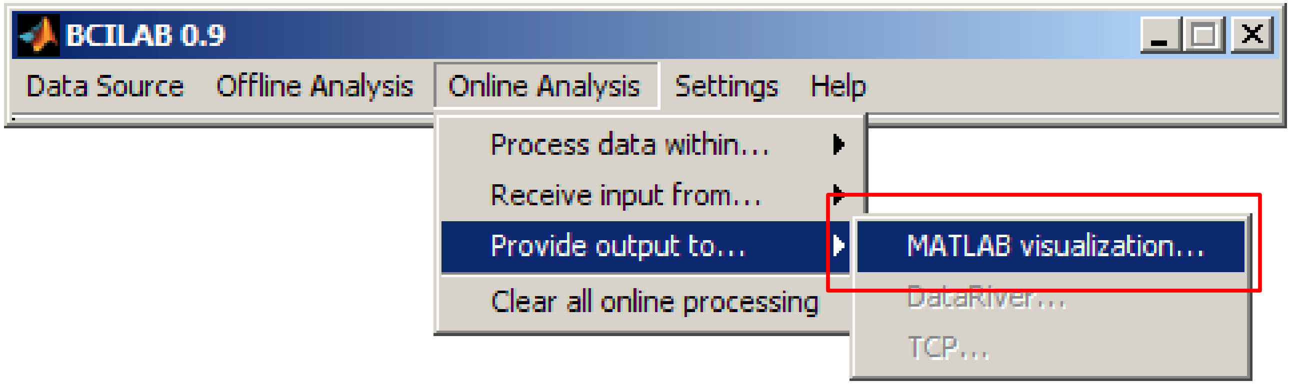
Apply model online

- This adds a data feed process in the background



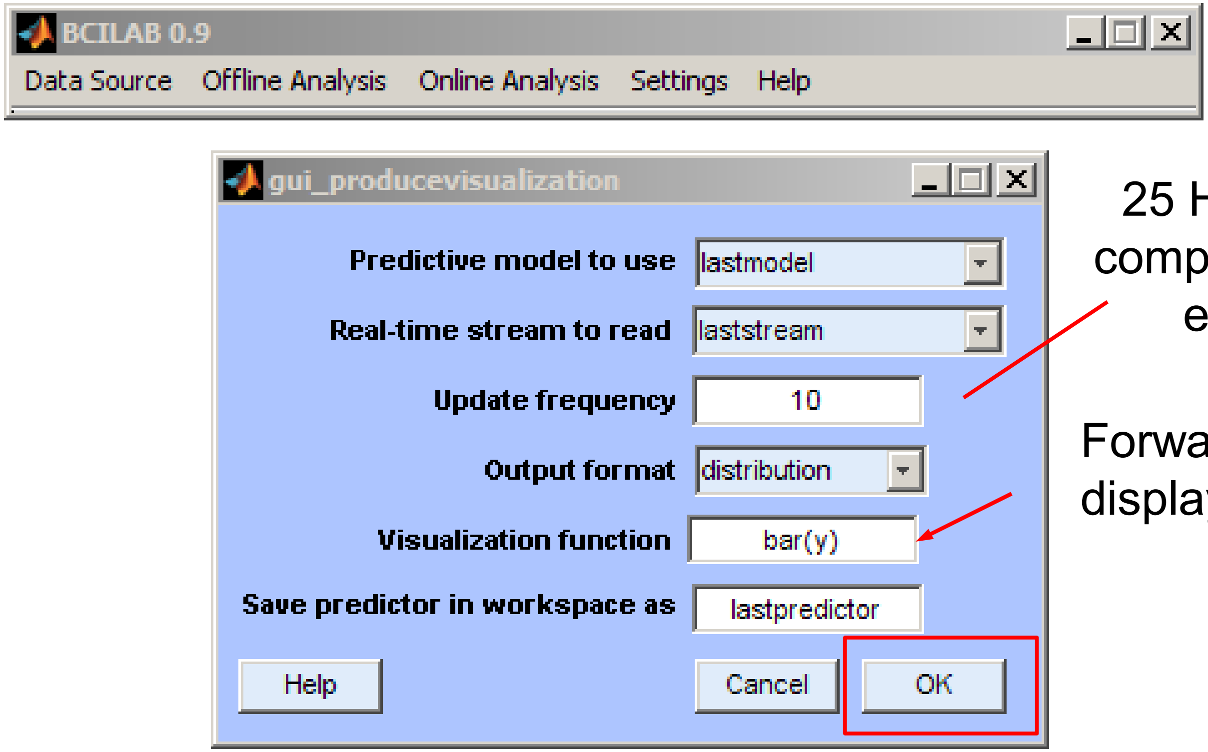


Apply model online



Apply model online

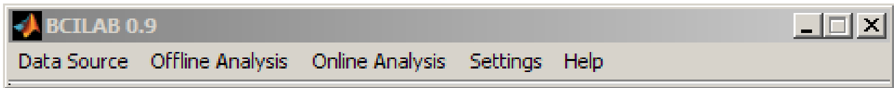
- This adds a real-time inference process in the background

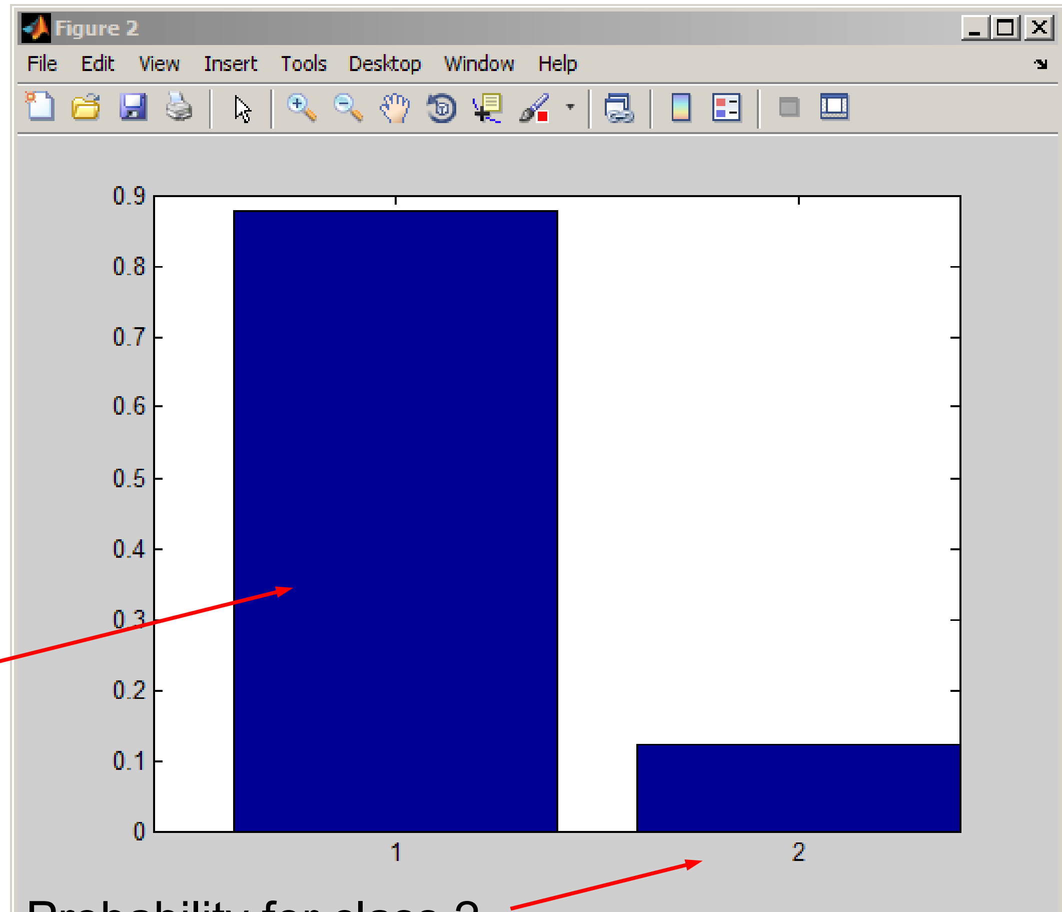


25 Hz if your computer is fast enough

Forwarded to this display command

Real-time output





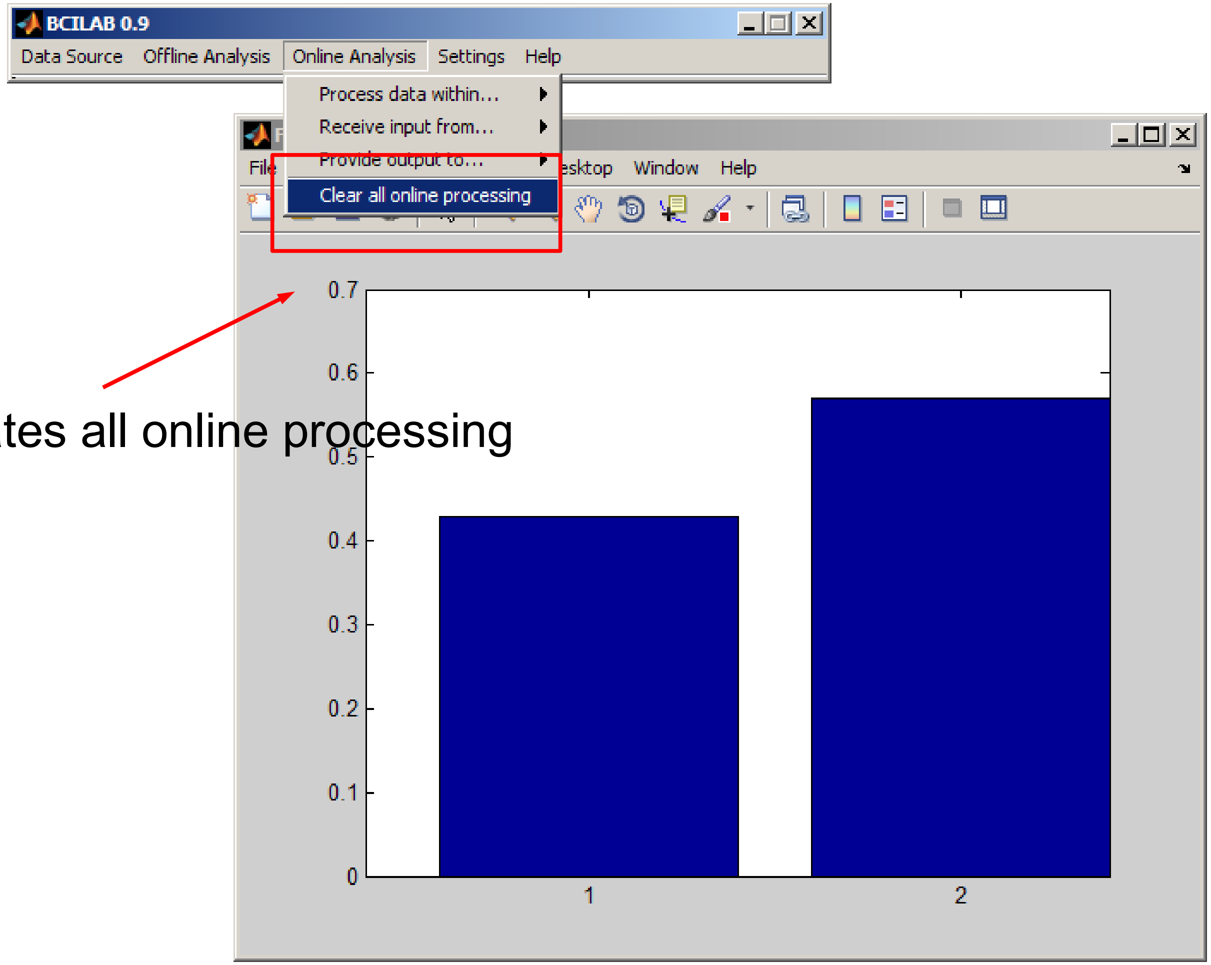
Probability for class 1

Probability for class 2

Real-time output

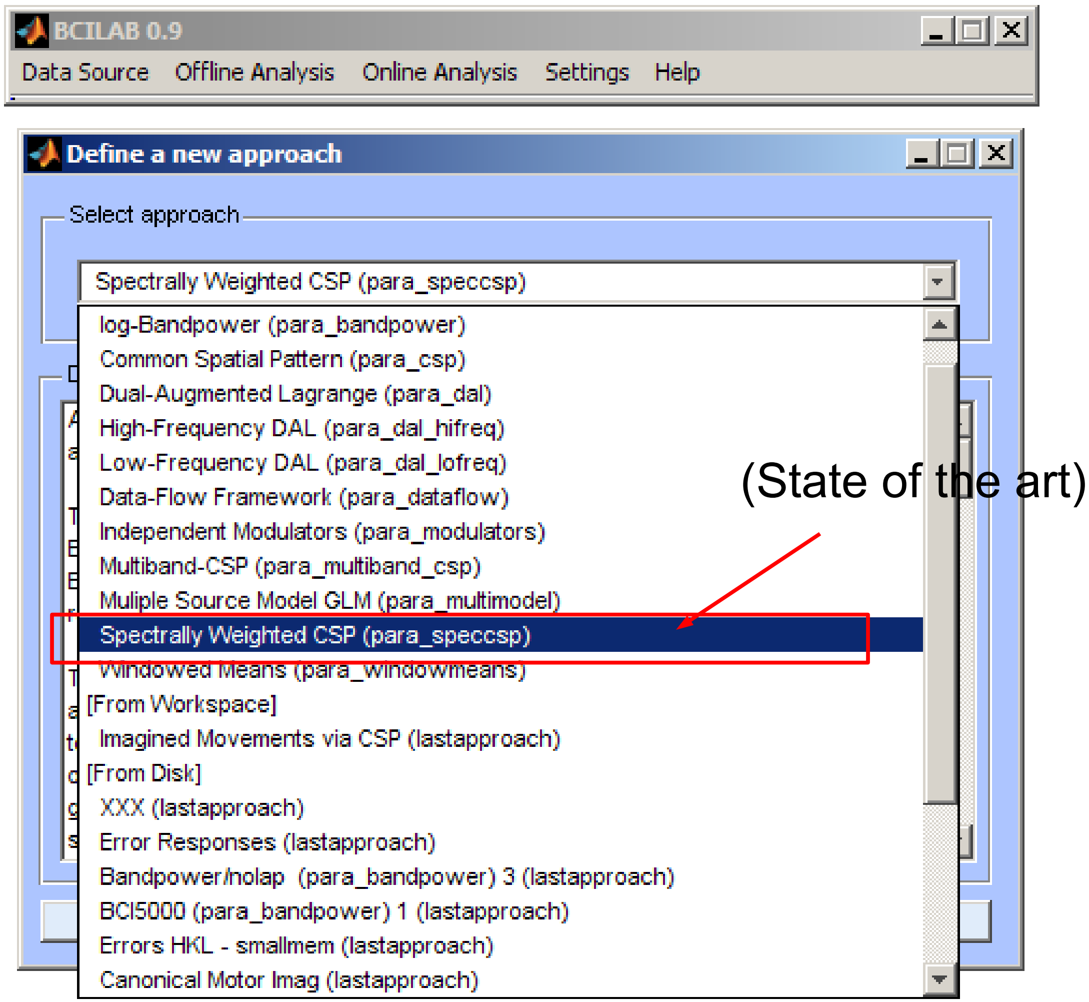
- If you have more classes, you get more bars
- You can also remap to other parameters (e.g. expected value)
- Note: the simple graphics command always renders into the current window

Real-time output

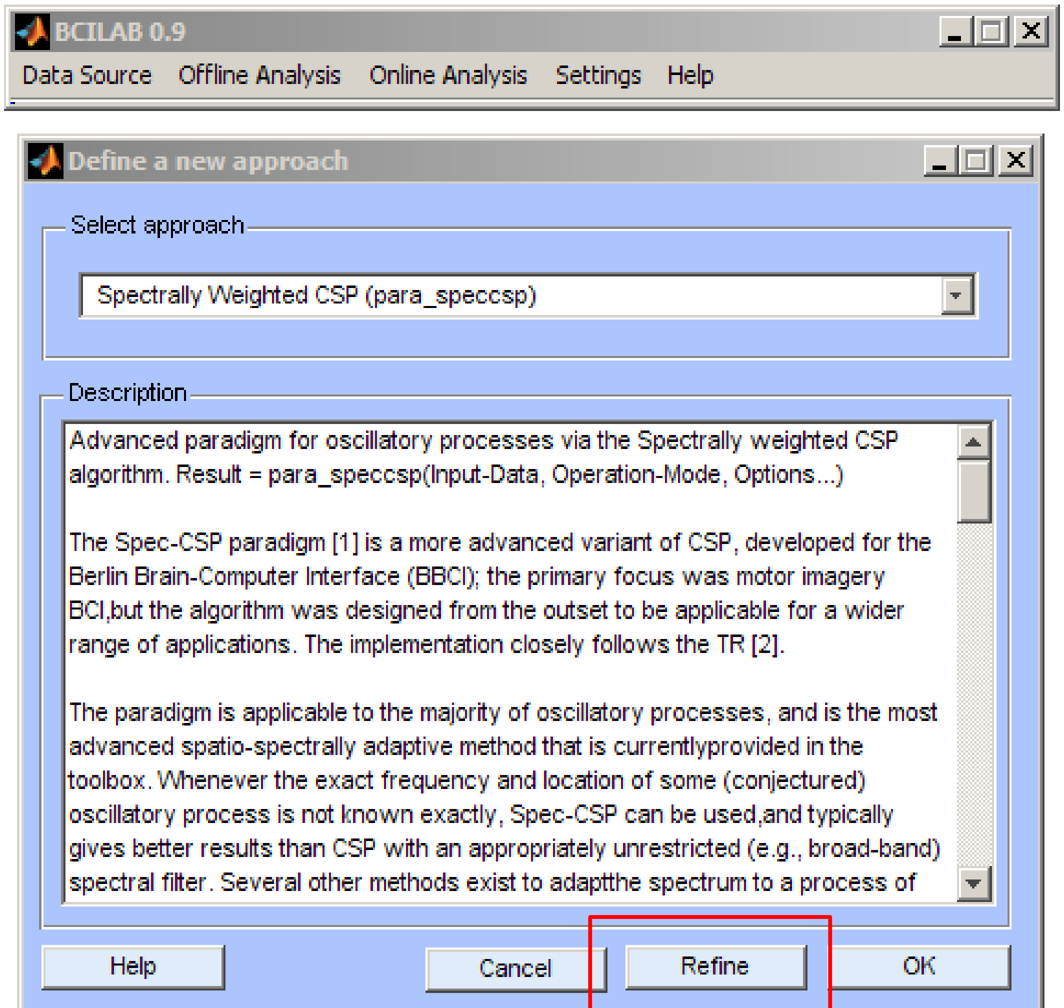


This terminates all online processing

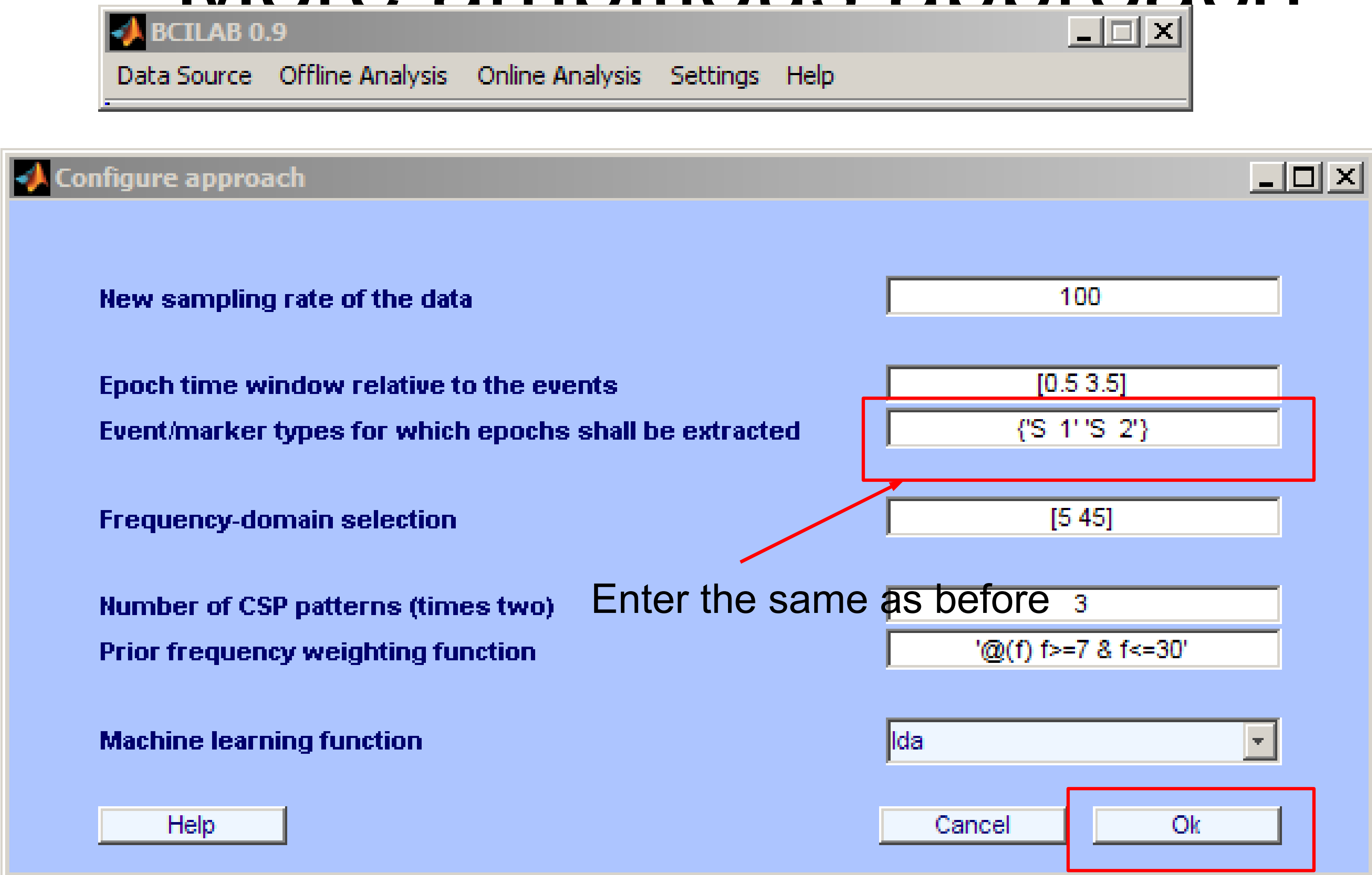
More ambitious approach?



More ambitious approach?



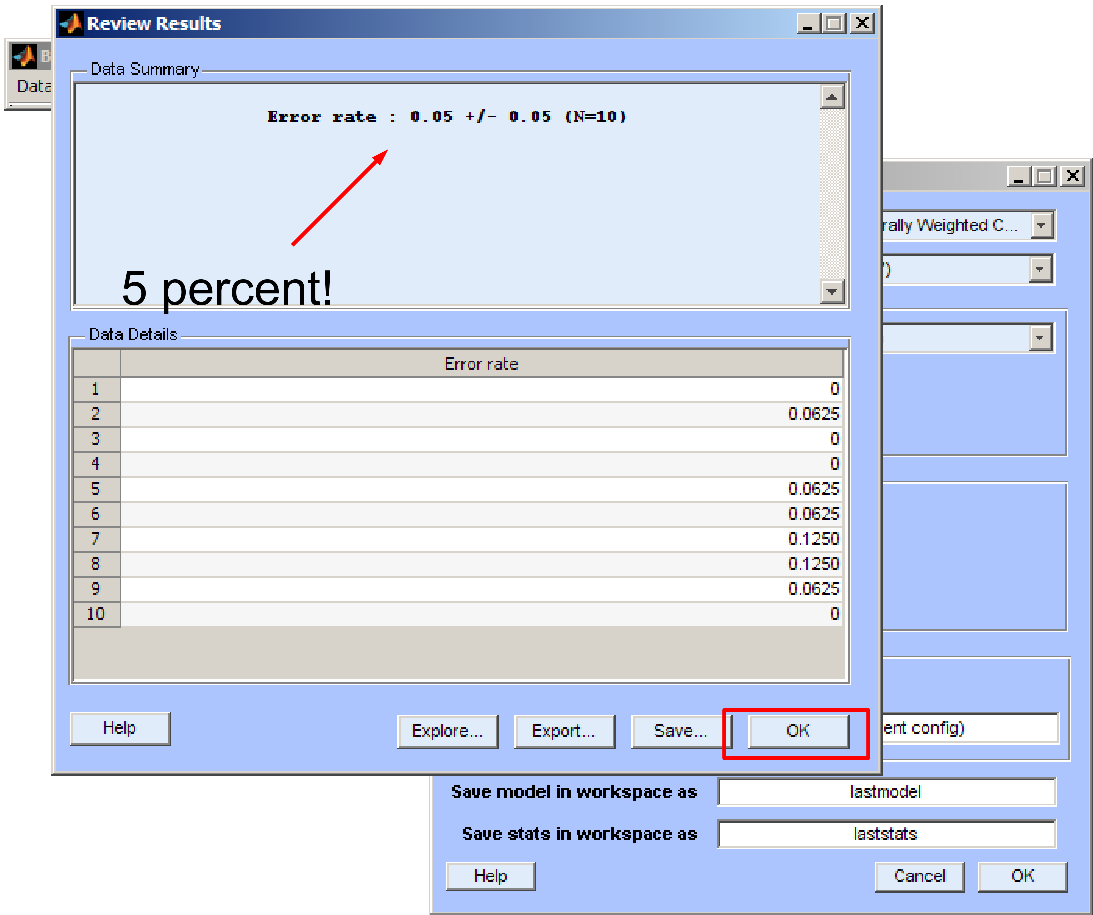
More ambitious approach



Enter the same as before

Train model, review results

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



Next steps

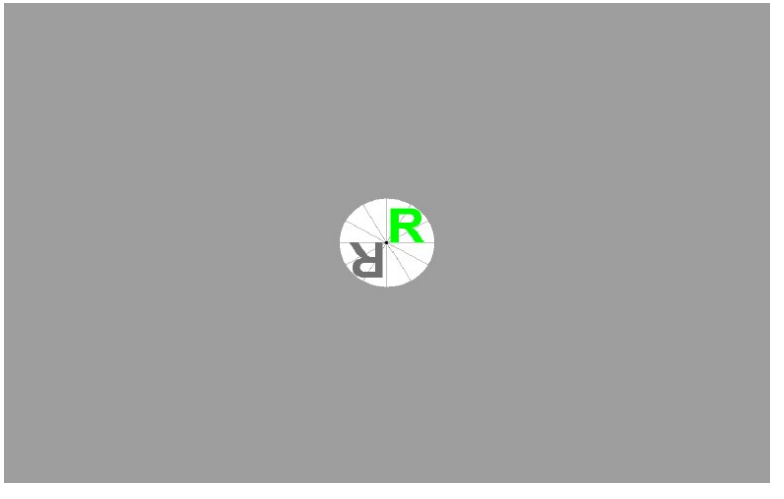
- Run online, apply to dataset, edit parameters, try to improve results, ...

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

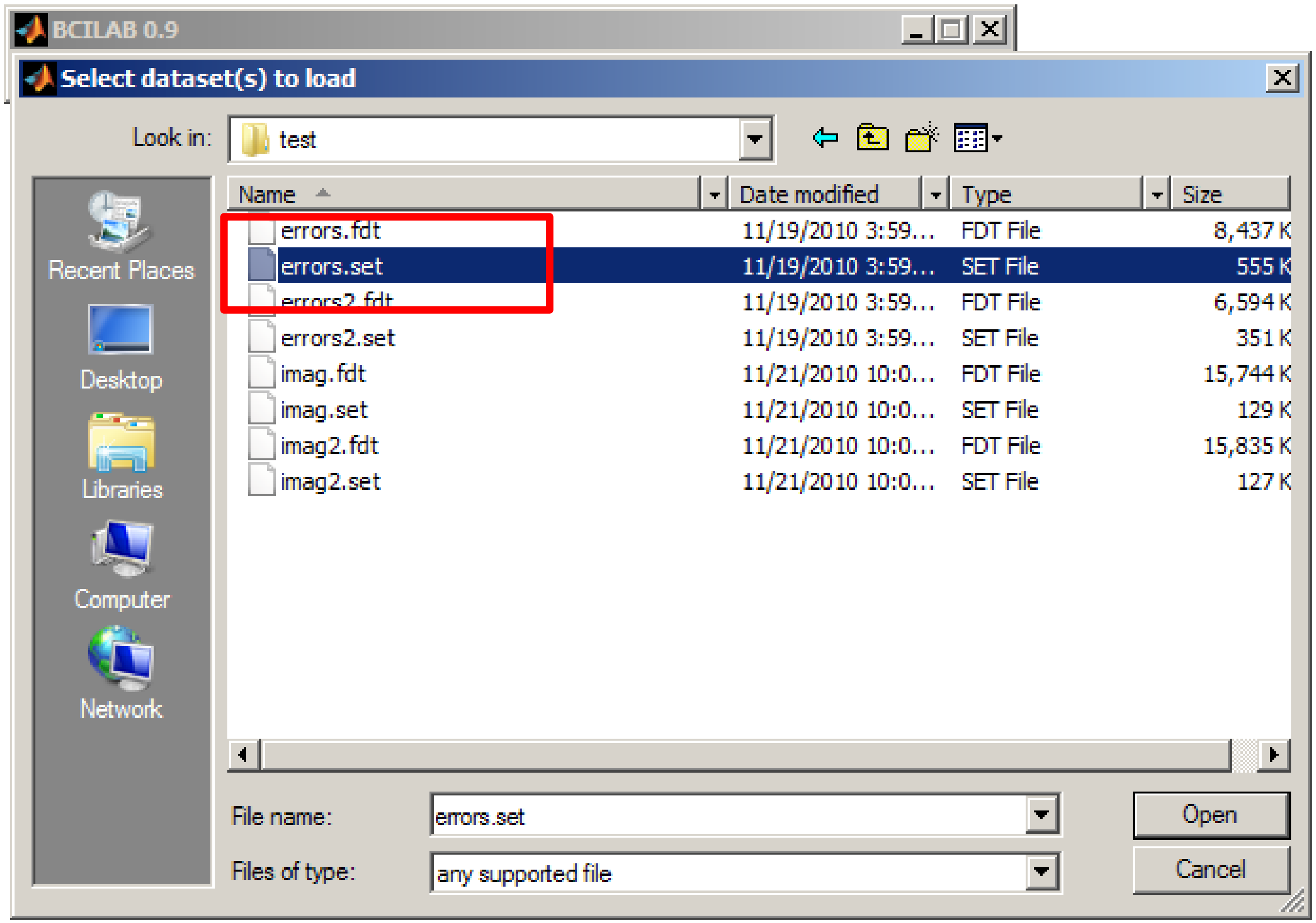
- Experimental task: ~100 randomized trials, 3 types of stimuli:
  - expected/correct event: type 'S 11'
  - unexpected event A: type 'S 12'
  - unexpected event B: type 'S 13'
- Sample:



Experimental task

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

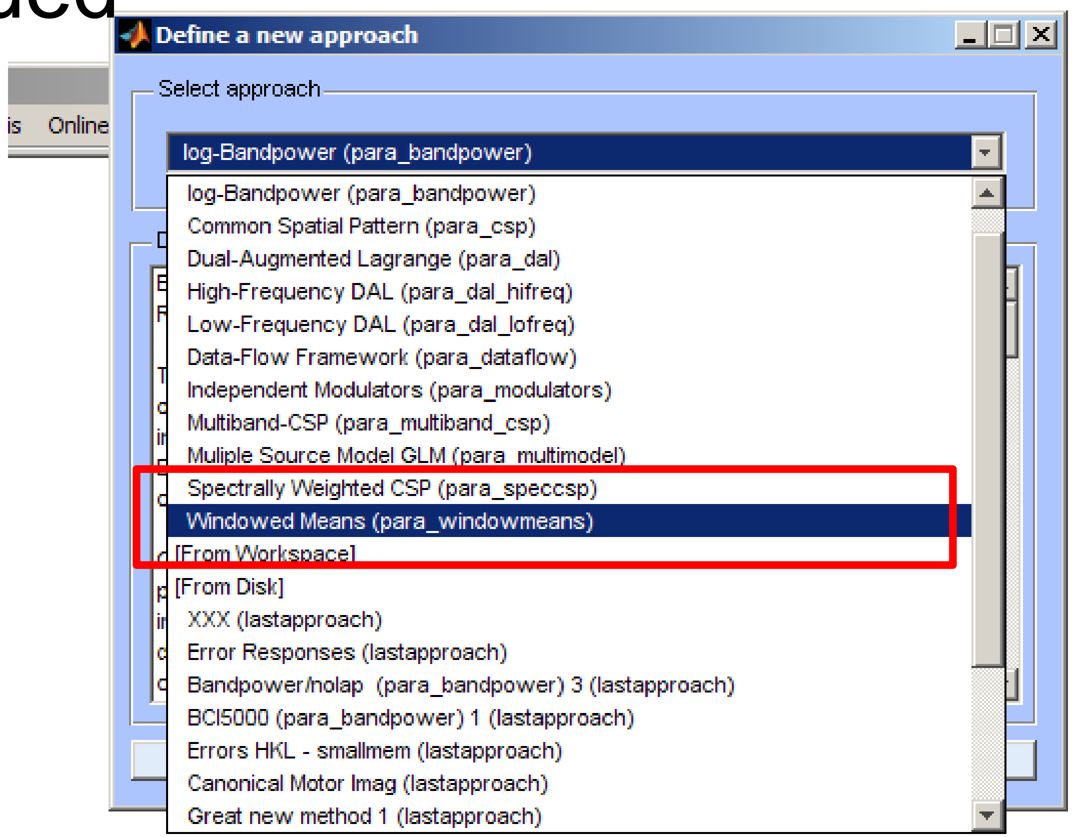
Load data



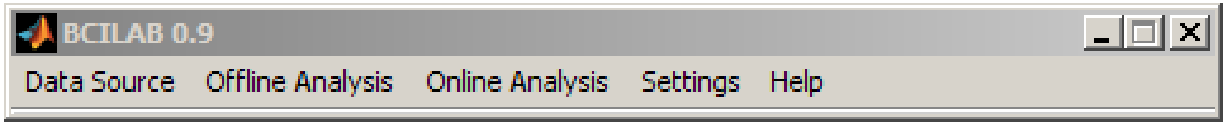
BCILAB 0.9
Select dataset(s) to load
Look in:
test
Recent Places
Desktop
Libraries
Computer
Network
Name
Date modified
Type
Size
errors.fdt 11/19/2010 3:59... FDT File 8,437 K
errors.set 11/19/2010 3:59... SET File 555 K
errors2.fdt 11/19/2010 3:59... FDT File 6,594 K
errors2.set 11/19/2010 3:59... SET File 351 K
imag.fdt 11/21/2010 10:0... FDT File 15,744 K
imag.set 11/21/2010 10:0... SET File 129 K
imag2.fdt 11/21/2010 10:0... FDT File 15,835 K
imag2.set 11/21/2010 10:0... SET File 127 K
File name:
errors.set
Files of type:
any supported file
Open
Cancel

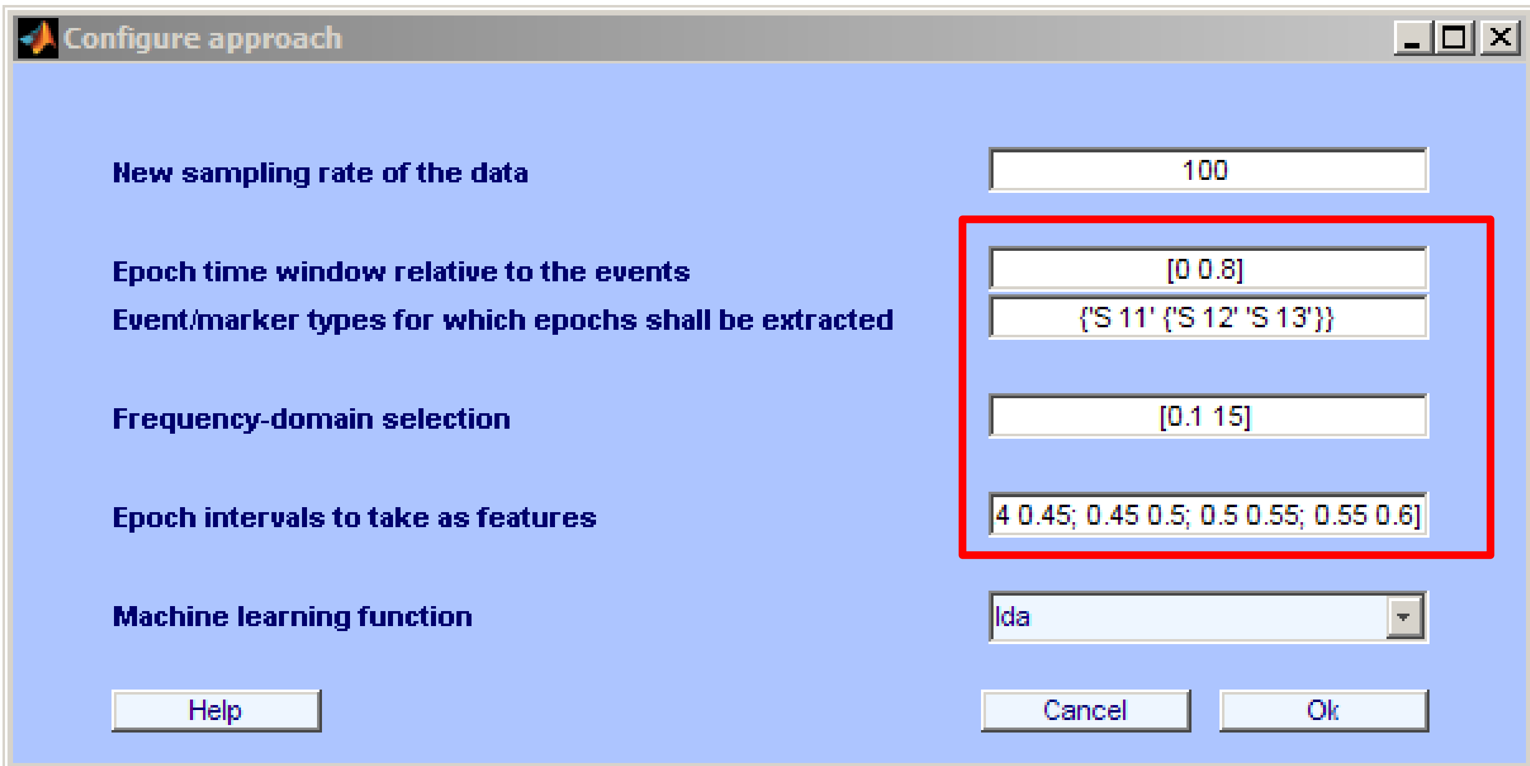
Define approach

- This time, an ERP-specific approach is needed



Major customizations

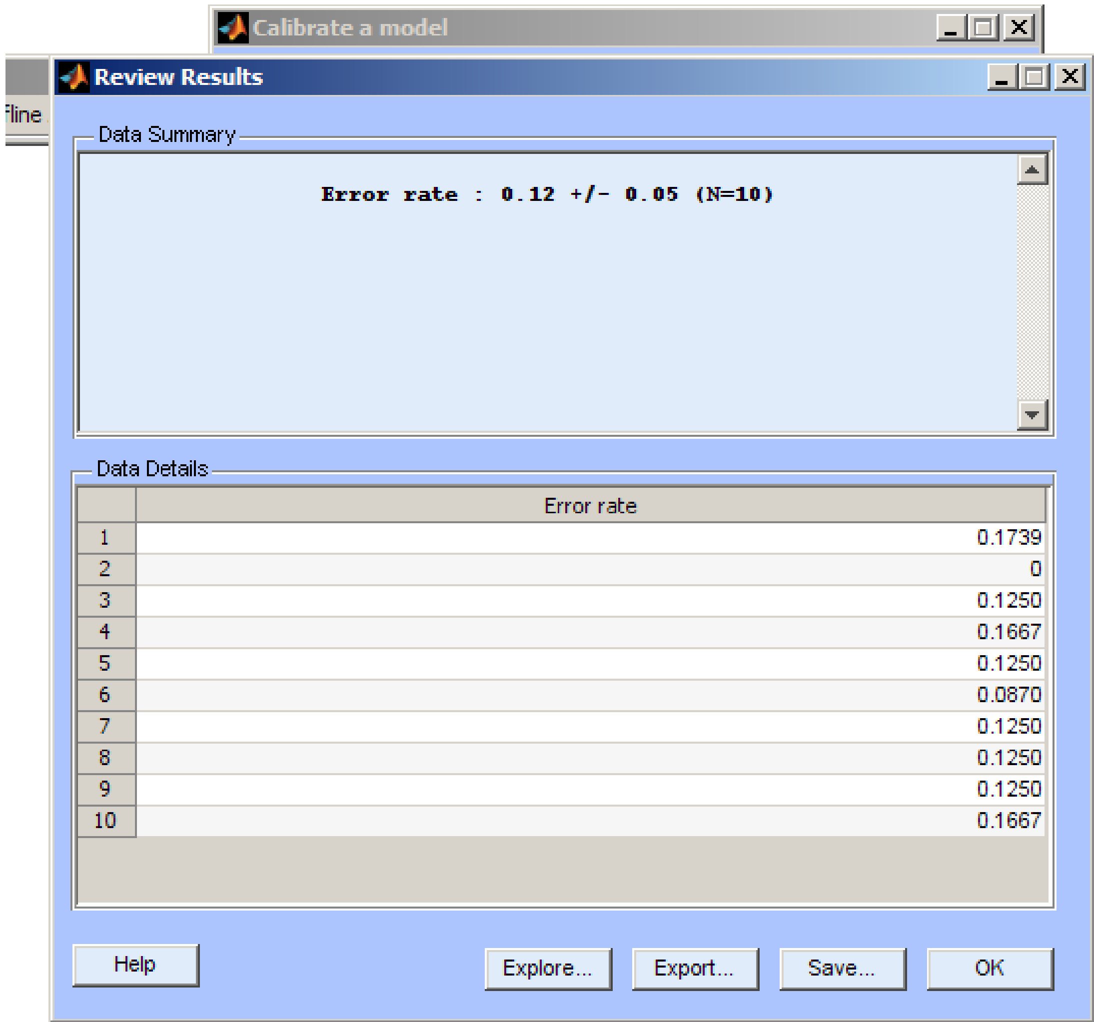




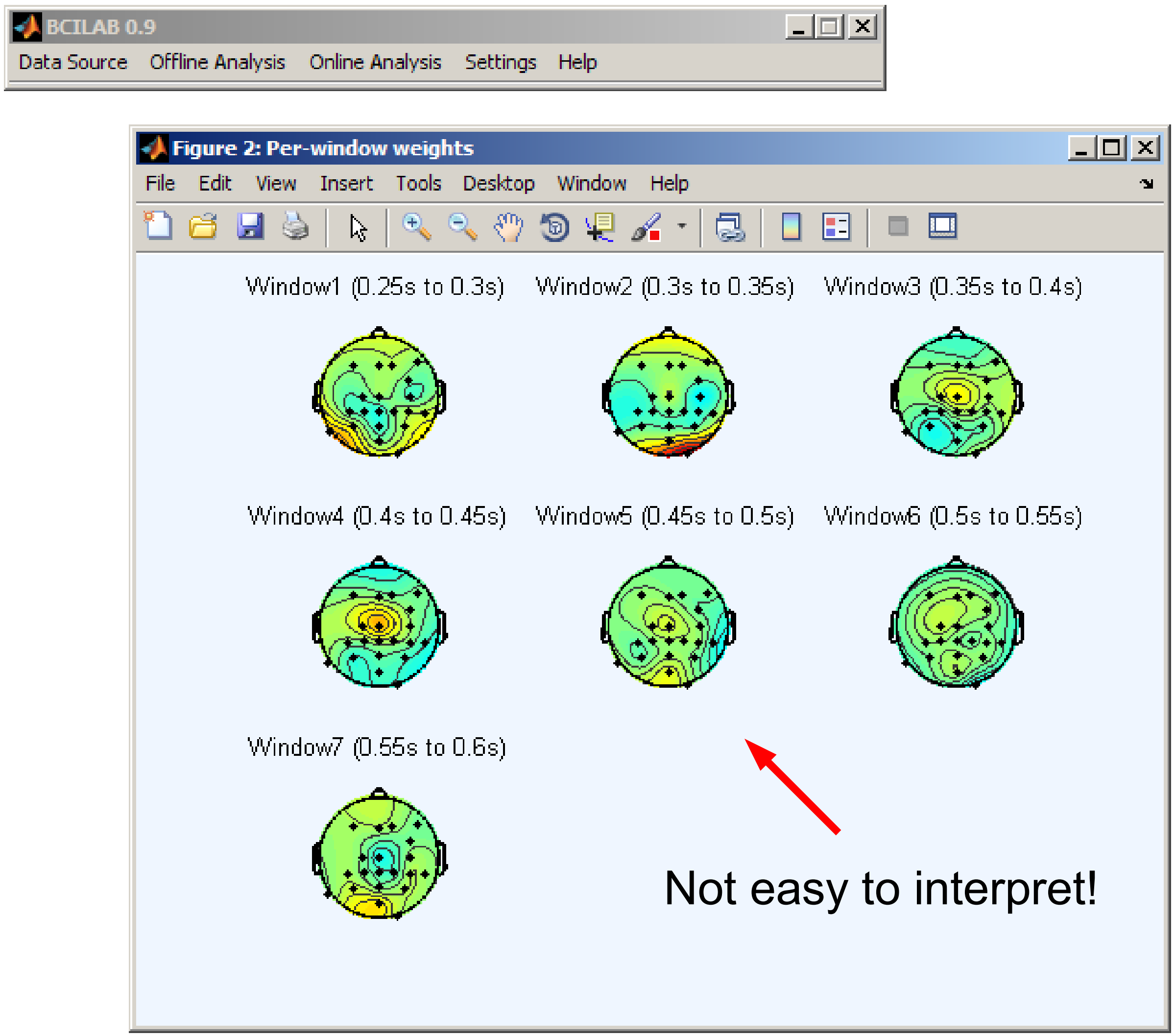
Event markers: {'S 11' {'S 12' 'S 13'}}

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]

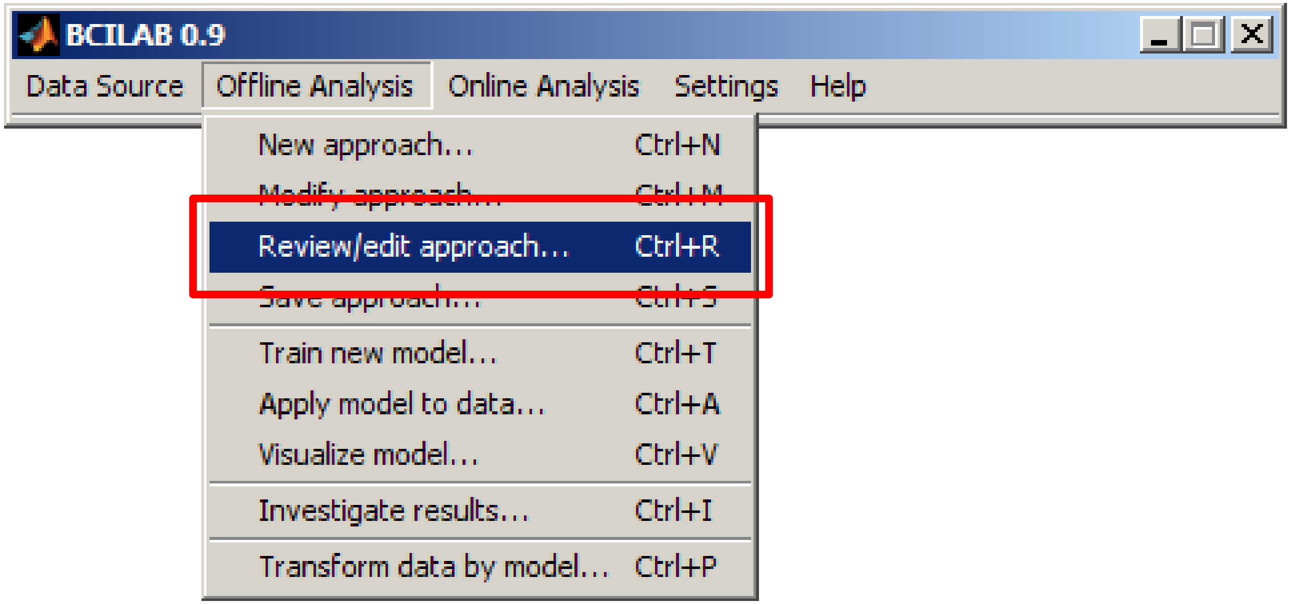
Train model, visualize



Train model, visualize

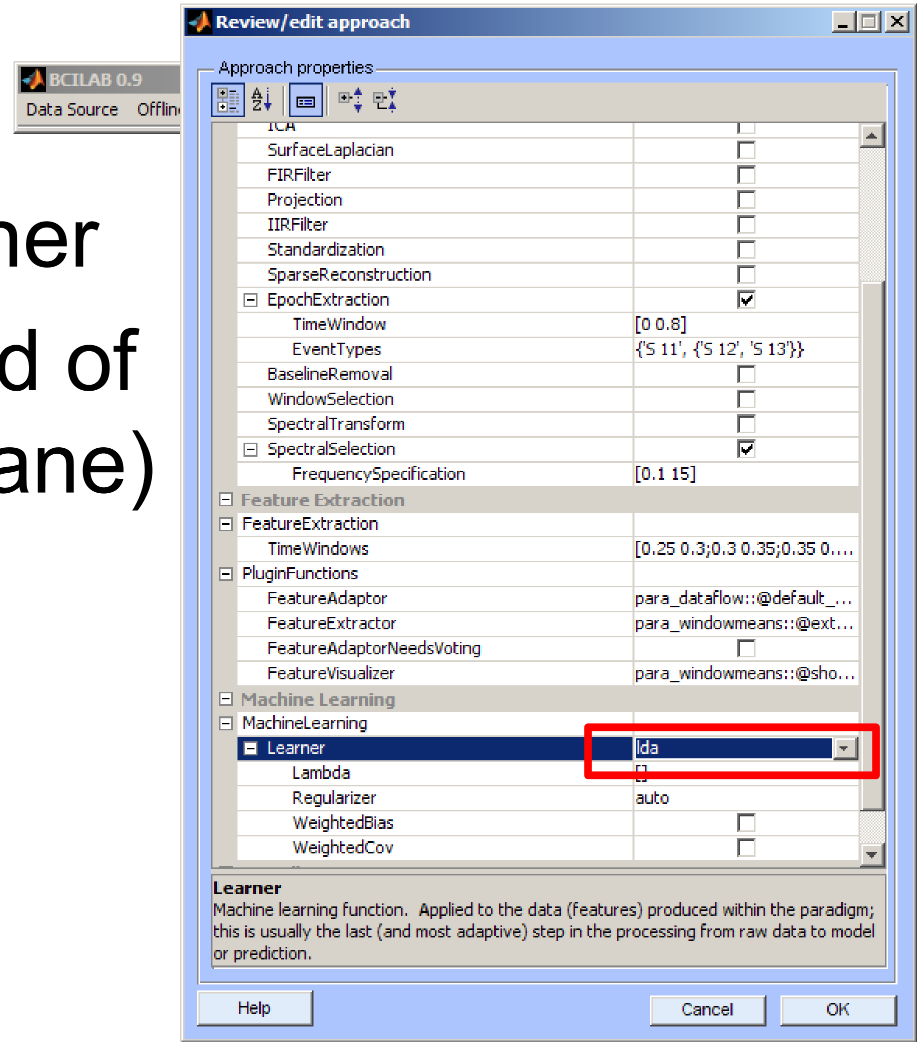


Using a sparse classifier



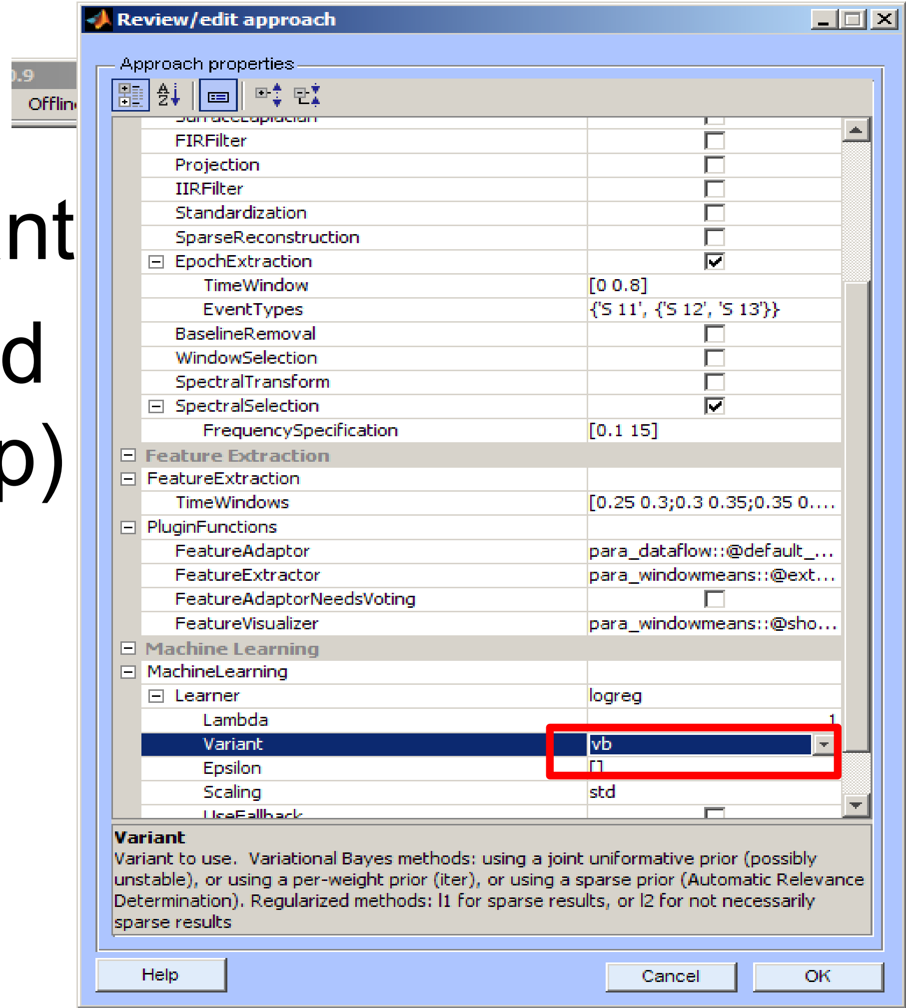
Using a sparse classifier

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



Using a sparse classifier

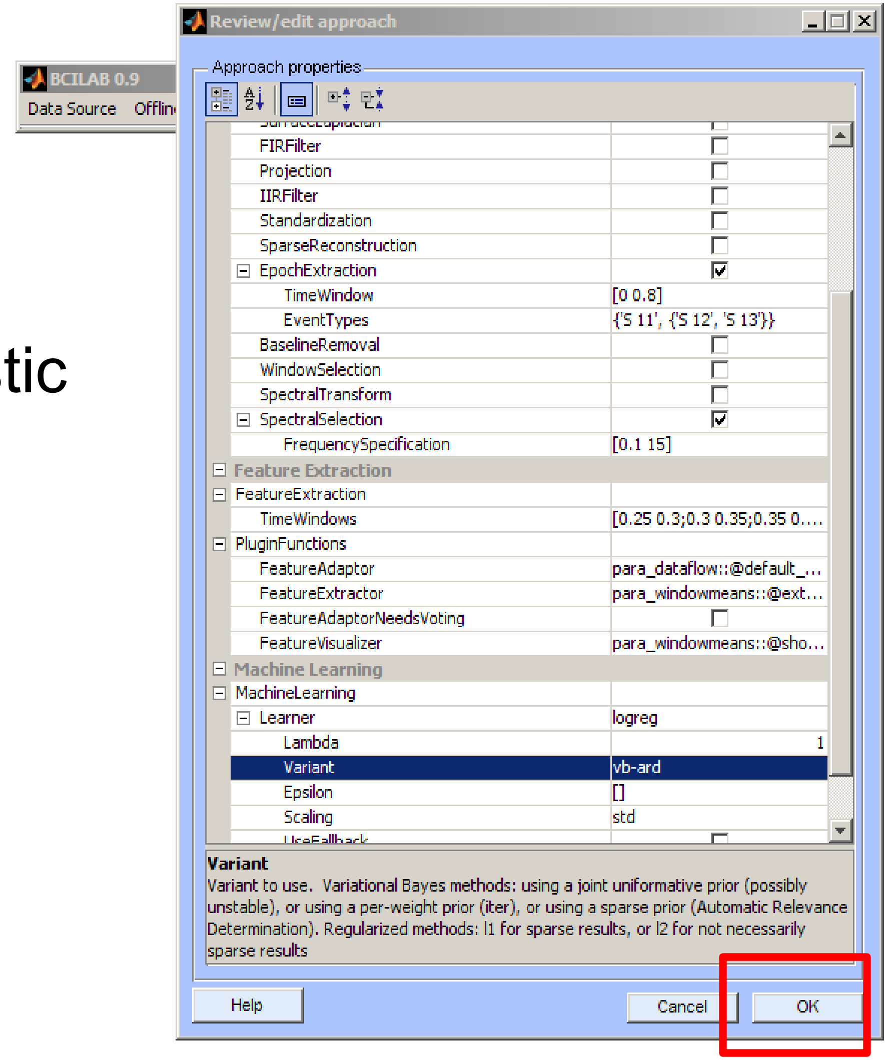
- Scroll down to Variant
- Select vb-ard instead of vb (again see help)



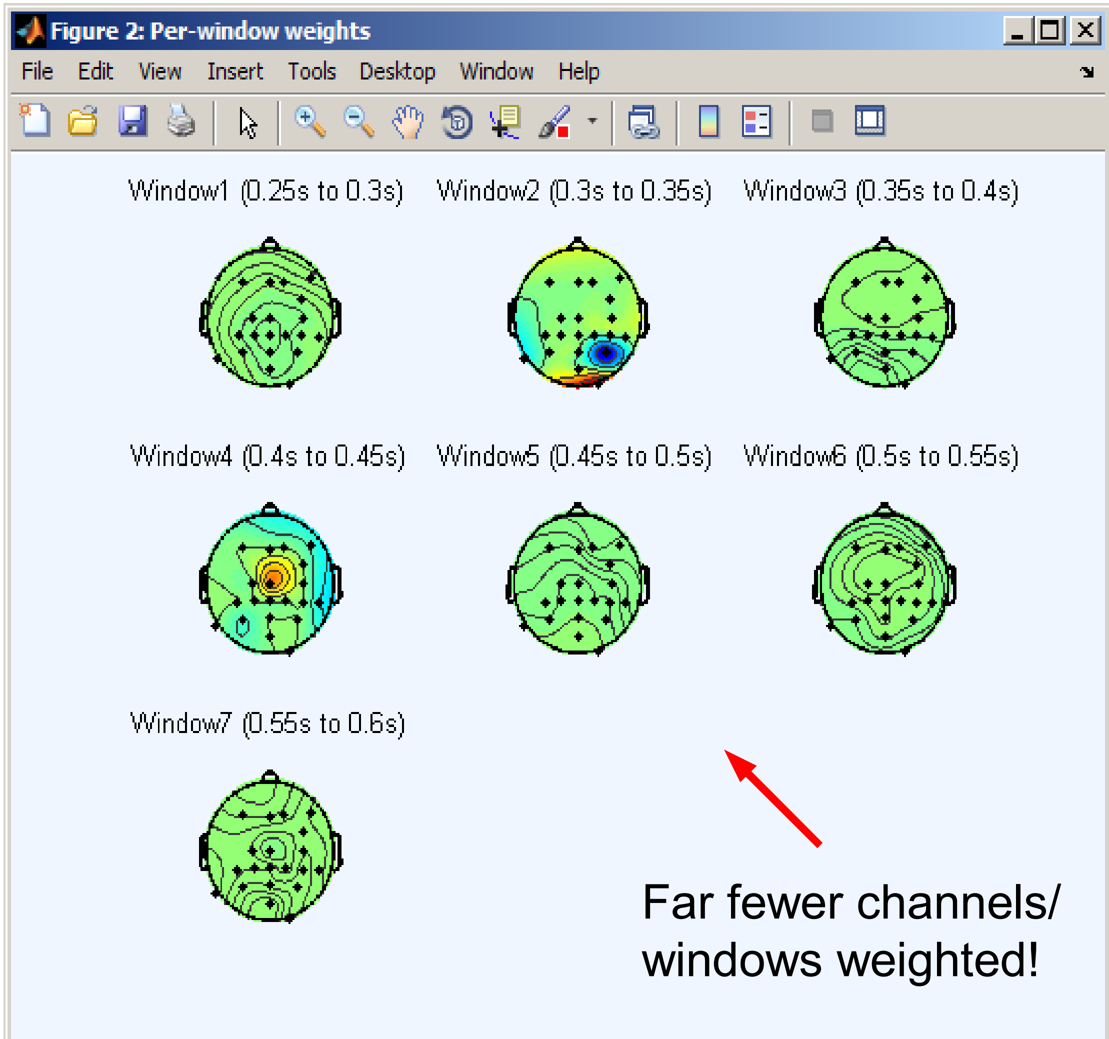
Using a sparse classifier

- Done!
- You selected: "Variational Bayesian Logistic Regression with Automatic Relevance Determinaton"

  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?