**Demo on how to use BCILAB**

1. Download BCILAB from the internet
   1. Two ways
      1. FTP
      2. BCILAB from github
         1. Careful to make sure you grab the stable ‘master’ branch
   2. Notes… I encountered 3 errors when trying to get it running
      1. ‘java.net.UnknownHostException: [LOCAL\_CMPTR\_NAME]’
         1. Solved with <https://coderwall.com/p/ktbkea/play-framework-2-error-nodename-nor-servname-provided-or-not-known>
      2. Error using supergui….  
         Error in eeglab>eeg\_mainfig…  
         Error in eeglab….  
         ….  
         ….
         1. Solved downloaded MATLAB R2014a
         2. This is a problem with the push button they BCILAB is using…
      3. Java.lang.NullPointerException error when you try to launch R2014a.
         1. Solved with patch: <http://www.mathworks.com/support/bugreports/1098655>
2. Go ahead and open the main folder and take a look at all the directories
   1. [Video Link](https://youtu.be/Sup2SaJyFQE?t=1m6s)
   2. Code
      1. Contains… all the code
   3. Dependencies
      1. All the externally maintained dependencies
   4. Userdata
      1. Which contains sample data and such
      2. What we will be working with for this demo
   5. Userscripts
      1. Contains a few tutorial scripts
      2. This is where you would put your scripts though
   6. bcilab.m
      1. The one main ‘m’ file that is ran
      2. The start up file
   7. Build
      1. Contains compiled versions of BCILAB that you can run without matlab, yay!
   8. ‘RELEASE NOTES.TXT’
      1. Just read them… it takes like five minutes
      2. Do it… I really did, release notes are not like terms of service ha, they are there to help you!
3. Getting BCILAB Started
   1. IF YOU HIT ERRORS SEE IF THEY ARE THE ONES I HIT AND SOLVED
   2. Switch into the directory where you extracted the BCILAB into, if you cloned with git, then the directory where it was cloned   
      cd /Users/myAwesomeName/[FOLDER\_BCILAB\_IS\_IN]; bcilab
      1. This is where errors come ☺
   3. A little GUI pops up if successful
4. Working in the GUI
   1. [Video Link](https://youtu.be/Sup2SaJyFQE?t=2m23s)
   2. This GUI allows you to do everything described in Module 6
   3. How do I load recorded data into BCILAB? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=3m8s)
      1. Data Source -> Load Recording(s)…
      2. In the tutorial select the 2nd file ending in ‘.vhdr’
         1. You can prune the data here before loading if you don’t have enough memory
      3. Tap ‘Ok’
      4. Test to make sure you loaded the data properly by running the command whos
         1. This will display everything in the workspace and provide you the name of the data that was just loaded in.
         2. Type in the Name returned, by default is ‘lastdata’
   4. How do I create a new approach with the GUI?
      1. [Jump to place in youtube video](https://youtu.be/Sup2SaJyFQE?t=10m54s)
      2. Office Analysis -> New Approach…
      3. Pick which ever ‘BCI Paradigm’ you want to use
         1. BCI Paradigms are covered in Lecture 6 as well!
      4. Click ‘OK’
      5. There are always defaults…
         1. These defaults come from some existing paper where the method originated from
         2. Not normally perfect for your task and will most likely have to be modified
      6. Put a general range of interest in for the EPOCH
      7. Windowed mean approach is used and comes in the form of   
         [0.1 0.2;0.2 0.3;….]
         1. Don’t want to use too many windows, start with 5 or 6
      8. Select which ever classifier/Machine Learning (ML) approach you want to use
      9. After you hit ok
         1. You can edit the documentation if you have something inventive here
         2. You can also save the approach to disk so you can use it from memory in the future
         3. By default approach is called ‘lastapproach’ in the workspace and you can change the name here as well
         4. Press ‘OK’ when finished
   5. How do I train a model to an approach? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=12m31s)
      1. Offline Analysis -> Train new model…
      2. Selected approach
         1. This will be any approach that is in the workspace
      3. Calibration data source
         1. The data with which you want to calibrate
      4. Target Markers
         1. Type in how the string markers in the data set relate to the string markers it would predict
            1. Do I want it to predict ‘-1’ & ‘1’
         2. These are the names of the target markers found in the data source, these are cell arrays and you need to make sure your spelling and capitalization is accurate.
         3. If you are unsure (should have it documented in the data) you can click the button “Inspect data…” which will bring up a snap shot of the raw EEG data with markers
         4. Each group of markers represents a class of output
         5. Subgroups are created with nested sub arrays
         6. Example  
            {‘marker1’,’marker2’,{‘marker3a’,’marker3b’}}
      5. Performance Search
         1. Loss/Performance Metric
            1. Determines how you want this to be determined, you could for example say “Mean Square Error (MSE)”
            2. Can stick to the default of “Automatically Choose”
      6. Performance Estimates
         1. Can determine how you want the data evaluated
         2. By default it is a block wise validation not randomized
            1. Because it comes from a time series
      7. Select ‘OK’ when complete and satisfied, you will see a print out in the command window   
         ‘beginning new computation…’
         1. It will take a few seconds to computer
         2. Memory could be an issue here, try to use a computer with at least 4 GBs of RAM
   6. How do I use the data source inspector? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=13m36s)
      1. You can see the markers here labeled at the top
      2. You may press the left (<) and right (>) or far left (<<) and far right (>>) to move in time through the data
   7. What does all the numbers on “Review Results” mean? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=17m43s)
      1. You can always pull up the last results by  
         Offline Analysis -> Review Results…
      2. Error Rate
         1. This is the amount your ML will be wrong like 5% of the time or something like that
      3. True Positive
         1. Correct Result
         2. ML predicts true and result is true
      4. False Positive
         1. Incorrect Result
         2. ML predicts true but really false
      5. True Negative
         1. Correct Result
         2. ML predicts false and result should be false
      6. False Negative
         1. Incorrect Result
         2. ML predicts false and result should be true
   8. How do I visualize the model [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=18m20s) [for writing a paper or so have you] that I just trained?
      1. Offline Analysis -> Visualize Model…
      2. This will populate a figure with the different epochs you selected. It will show the linear weights of the classifier assigned to the different features
         1. i.e. “there are 64 features for the first time window, one per channel”
      3. You get spatial filters here
         1. Some could be surface lapacian where you subtract the surrounding channels
         2. Some could pick up a dipole
      4. You can work with independent components to try and localize the error
         1. There are many papers on this
   9. How do I test my new model on some new data? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=20m7s)
      1. Load New data   
         Data Source -> Load Recording(s)
         1. If you use the same name such as ‘lastdata’ it will overwrite the data that you built your model with…
      2. Entering a command of ‘whos’ into the command window should show
         1. ‘lastapproach’, ‘lastdata’, ‘lastmodel’,’laststats’
      3. Now we will apply the new data to the model using  
         Offline Analysis -> Apply model to data…
         1. Source data set for prediction
            1. This should be the data you just loaded in
         2. Predictive model to use
            1. This is the model you are trying to test out… normally the model you just created
         3. This brings up a “Review Results” window that is documented above on how to use
   10. How do I simulate online analysis? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=22m27s)
       1. If you do not have an EEG then for example we can take dataset1 and play it in the background and then we classify online & visualize the outputs of the classifier with dataset2
       2. To load the dataset into the background  
          Online Analysis -> Read Input From… -> Dataset…
       3. By default it will use the most recent dataset normally called ‘lastdata’
       4. Now a call in the cmd window of ‘whos’
          1. Shows several new structs, the ones of interest are ‘laststream’
       5. You can use the MATLAB visualizer
   11. How do I use the MATLAB visualizer? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=23m33s)
       1. Send to the MATLAB visualizer  
          Online Analysis -> Write output to… -> MATLAB visualization…
       2. Can configure the analysis in several ways such as what you model you want to look at and which stream you are going to use
       3. Making sense of what we are seeing…
          1. We are seeing a bar diagram of the probability that the subject is in condition 1 (no error) or condition 2 (error)
   12. How do I perform online analysis with actual hardware?
       1. Online Analysis -> Read input from… -> Lab streaming layer…
   13. How do I run a batch analysis? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=26m8s)
       1. Offline Analysis -> Run batch analysis…
       2. By default it will list everything it finds in memory
       3. Compare approaches by listing multiple approaches under the ‘Approaches’ section
       4. For “Training Data Sets”
          1. Can use datasets that have been loaded into the workspace
          2. Can also use datasets that are located in files… they should end in ‘.set’  
             C:\test\\*.set
       5. Can really customize the entire pipeline by selecting “Edit…” next to “Load Arguments”, “Train Arguments”, “Predict Arguments”
   14. How do I change the Settings? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=28m27s)
       1. Settings -> Directory settings…
          1. Where your data lives
          2. Where your results should be stored
       2. Settings -> Cache settings…
          1. Configures if and where you want to use the cache to save intermediate calculations
          2. Cache saves you a lot of time if you want to calculate a result you already calculated before it is instantly available but you can run into trouble and make mistakes if you for example override your original data on disk and expect your results to change if you rerun your pipeline, but if they were stored in cache then you suddenly get the same results again.
   15. How do I go beyond the built in methods and customize them? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=30m17s)
       1. Offline Analysis -> Review/edit approach…
       2. Pick the approach you want to edit, select ‘OK’
       3. A dialog will appear with a ton of parameters which are fundamentally divided into two sections “SignalProcessing” and “Prediction”
       4. The plugins that you are turning on can be found in the code section of the top level BCILAB folder discussed earlier, you can really deep dive into the various filters and such and look at their code
5. Using scripting
   1. How do I learn about scripting? [(Video Link)](https://youtu.be/Sup2SaJyFQE?t=37m47s)
      1. Defining the training data  
         io\_loadset(‘filepath/file.vhdr’);
      2. Defining an approach
         1. Generally you say what paradigm you want to start with and then you can go on and override all the nested parameters that you want to change
         2. Basically boils down to just setting a bunch of parameters and choosing which paradigm you want to use
      3. Training a model  
         bci\_train
         1. This simply take the data, approach and markers that we defined and outputs the model and statistics
      4. Clear your workspace if necessary to free up some memory
      5. This easily allows you to quickly write up like ten different approaches
6. Plugins
   1. Signal processing plugins are in the Code/filters directory
      1. Every filter must have two lines  
         if ~exp\_beginfun(‘filter’) return; end  
         exp\_endfun;
      2. Theses line let you lazily get your pipeline data
   2. The line of code called declare\_properties allows the GUI to properly order a plugin
   3. The line of code called arg\_define is used for the GUI once again and it allows you to write nice plugin with multiple types such as Boolean, floats, and arrays