Follow each and every step mentioned in the official build on the inria website.<http://openvibe.inria.fr/tutorial-1-implementing-a-signal-processing-box/>

For our CCA box we have one signal input, one stimulation input and one signal output.

You can customize this during the skeleton generator and change the type and number of inputs/outputs.

Once the src code is downloaded, dependencies installed and the skeleton is generated.

Copy these .cpp and .h files and replace them at the destination.

(Sample:G:\openvibe-1.3.0-src\openvibe-1.3.0-src\contrib\plugins\processing\signal-processing\src) <https://drive.google.com/open?id=0B4a3IAedywxAZ0VBRm1uTWZ2SW8>

The actual CCA code is embedded in the OVPsignalflattener.ccp. Also make sure the standalone CCA code compiles and returns the proper values corresponding to the matlab implementation.

The ccp code: <https://drive.google.com/open?id=0B4a3IAedywxAWnZrR29qTkRNRE0>

The matlab code: <https://drive.google.com/open?id=0B4a3IAedywxAOEJldjM0cHpmYUU>

\*Important: We are using the Eigen Library to implement some matrix functions in our code. To include this library open the CMAKElists.txt and add the following line INCLUDE("FindThirdPartyEigen")

If you are using some other uncommon library then you to find a way to include them or contact Jussi for help.(or be really good at CMAKE)

Now some information about CCA implementation

We are implementing a one class problem, i.e trying to detect a specific frequency or not.

However you will notice in the code that we are finding Eigen values, in our scenario, for 15Hz, 10Hz and some 6Hz. This is purely to enhance the performance of the scenario( eliminate the possibilty of it detecting 15hz when the subject has looked away or due to some movement/random events).

The window size we have taken @250samples/sec is 500 samples and the window moves every 25 samples i.e the window duration is 2s and overlap is 0.1s. U can experiment with shorter window length forsake of performance but accuracy might suffer(Not tested).

\*IMPORTANT\*

When we tested the real time scenario, sometimes performance would suffer and you would a warning message in OpenVIBE console saying that “Player cannot real time”. You never want to get into this situation as this means the scenario is not able to process the data in real time.

The solution right now is to increase window overlap time say to .15s or .2s or Rewrite a much efficient CCA code :)