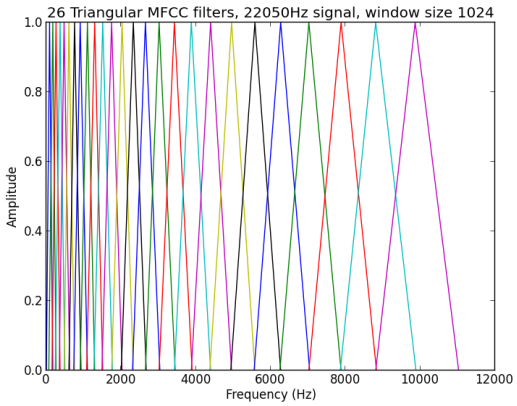
Assignment 2: Perceptual Features Extraction

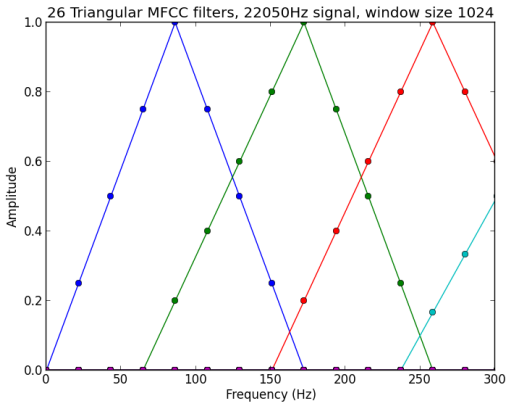
CS 4347: Sound and Music Computing

due February 11 (Monday), 11:59 pm

0. This assignment will use the same "music / speech" dataset that we used for assignments 1.

1. Follow the following steps to complete this assignment:
   - Read the ground truth `music_speech.mf` file.
   - Load each wav file and splits the data into buffers of length 1024 with 50% overlap. Only keep complete buffers.
   - Calculate the MFCCs for each window as specified in the lecture notes. Here are more detailed steps:
     - Given input $x(t)$ and output $y(t)$, the pre-emphasis filter should be
     $$y(t) = x(t) - 0.95x(t-1).$$
     - Use a Hamming window before the mag-spectrum calculation.
     - Mel-scale of frequency f is:
     $$Mel(f) = 1127 \ln(1 + \frac{f}{700}).$$
     - Calculate 26 mel-frequency filters, covering the entire frequency range (from 0 Hz to the Nyquist limit). To calculate the filters,
       * find the X-axis points of the filters (left side, top, right side). All points must be convereted into integer FFT bins; the left side should use the `floor()` operation; the top point should use `round()`; the right point should use `ceil()`.
       * assign the left bin to be 0, top bin to be 1.0, right bin to be 0; linearly interpolate between the rest.
     - the log step should be log base 10.
     - scipy has DCT built-in: `scipy.fftpack.dct()`
     - do not calculate any delta-features
   - Calculate the mean and standard deviation for each MFCC bin over the entire file. So if there are M MFCC bins in each buffer, you will end up with a feature vector of length $2M$ for each song.
   - Write the data to an ARFF file (each line should contain the 26 means, followed by the 26 standard deviations, and finally the class).
   - Make two plots: the overall range of the triangular windows, and the triangular windows from 0 to 300 Hz. They should match the examples below.





2. Submit a zip file to IVLE containing your program's source code ((a single `.py` file), the ARFF file and 2 plots. Name the zip file using your student number (e.g. `A0123456H.zip`). Late submissions will receive no marks.

3. **Note:** You may use any python standard libraries, numpy (including pylab / matplotlib) and scipy. No other libraries are permitted.

4. Grading scheme:

   - **4/9 marks**: correct ARFF file.
   - **2/9 marks**: 2 correct plots.
   - **3/9 marks**: readable source code (good variable names, clean functions, necessary comments).