Audio Content Analysis
Assignment # 2
Instructor: Juan Pablo Bello

A Implementation [7 pts]

In this section you will be implementing a series of functions, and finally a script tying them all together. Each function should be saved as a separate .m file, where the filename is identical to the function name. The functions definitions should ***exactly*** *follow the given code. Submit all code to NYU classes as a single zip file named "YourLastName2.zip".*

Details for each function can be found at the end of the assignment.

0. Write a Matlab function that reads any audio (wav) file, and returns the signal $x(t)$, the vector of time points t in seconds, and the sample rate f_s. If the audio is stereo, the second channel should be disregarded. (This will be a very short function). [0.5 pt]

```
[x_t, fs, t] = import_audio(filepath)
```

```
>> %% Example call %%
>> [x_t, fs, t] = import_audio('synth.wav')
>> soundsc(x_t, fs); % plays the audio
```

1. For each of the following, write a Matlab function that computes the specified novelty function, the vector of times (in seconds) corresponding to the points in the novelty function, and the sample rate of the novelty function. If a window function is needed, use a hamming window. [2 pts]

   (a) Log Energy Derivative
   ```
   [n_t_le, t_le, fs_le] = compute_novelty_le(x_t, t, fs, win_size, hop_size)
   ```

   (b) Rectified Spectral Flux ($SF_R(m)$ in the slides)
   ```
   [n_t_sf, t_sf, fs_sf] = compute_novelty_sf(x_t, t, fs, win_size, hop_size)
   ```

2. Write a Matlab function that computes onsets by peak picking a novelty function. Before peak picking, it should first pre-process the novelty function by (1) smoothing using a low-pass first-order butterworth filter, (2) normalizing so that the maximum value is 1, and (3) adaptive thresholding using the local median. The MATLAB functions `butter`, `filtfilt`, `medfilt1`, and `findpeaks` may be useful. [2 pts]

```
[onset_a, onset_t, n_t_smoothed, thresh] = ...
    onsets_from_novelty(n_t, t, fs, w_c, medfilt_len, offset)
```

3. Using the previously written functions, write a Matlab function that takes the name of a wav file and plots the following, with the x-axis representing the time in seconds (as subplots in a single figure):

   - the waveform
   - the log energy derivative novelty function

- the rectified spectral flux novelty function

Within the last two subplots, plot the novelty function (solid red line) with the following plots overlaid: the smoothed novelty function (solid black line), the adaptive threshold (dotted blue line), and the detected onsets (black circles). Your plots should be titled and labeled appropriately. [2 pts]

```
create_novelty_plots(filepath, win_size, hop_size, ...
                     w_c, medfilt_len, offset)
```

4. Write a Matlab script called `assignment2.m` which runs the code used to write your report (see the Analysis section below) [0.5 pt].

B Analysis [3 pts]

Write a report addressing each of the questions below. Please submit your report as a pdf file to NYU Classes.

In this section, you will be exploring how different onset detection methods perform on different types of audio signals.

1. Download the `.wav` files from the course homepage. Perform the following steps for each file. Use the default parameters `win_size=1024, hop_size=512, w_c=4 Hz, medfilt_len=8, offset=0.01`. [1.5 pts]
   (a) Use the `create_novelty_plots` function above to visualize all the novelty functions and detections.
   (b) Comment on the characteristics of the sound.
   (c) Comment on the resulting plots, their differences, and their relationship to the characteristics of the sound.
2. Using the `create_novelty_plots` function to guide you, explore for each novelty function different parameter settings of the variables `win_size, hop_size, w_c, medfilt_len`, and `offset`. [1.5 pts]
   (a) Discuss your observations about how the value of each variable changes the resulting novelty function.
   (b) Report the 3 configurations you found that had the best performance across all files.

Function Details

The functions you write for section A should begin with the headers below, and the inputs/outputs should follow the specifications in the function's doc-strings.

```
function [x_t, fs, t] = import_audio(filepath)
%    Import an audio signal from a wave file.
%
%    Parameters
%    ----------
%    filepath : string
%        path to a .wav file
%
%    Returns
%    -------
%    x_t : 1 x T array
%        time domain signal
%    t : 1 x T array
%        time points in seconds
%    fs : int
%        sample rate (samples per second)

function [n_t_le, t_le, fs_le] = compute_novelty_le(x_t, t, fs, win_size, hop_size)
%    Compute log energy derivative novelty function.
%
%    Parameters
%    ----------
%    x_t : 1 x T array
%        time domain signal
%    t : 1 x T array
%        time points in seconds
%    fs : int
%        sample rate of x_t (samples per second)
%    win_size : int
%        window size (in samples)
%    hop_size : int
%        hop size (in samples)
%
%    Returns
%    -------
%    n_t_le : 1 x L array
%        novelty function
%    t_le : 1 x L array
%        time points in seconds
%    fs_le : float
%        sample rate of novelty function

function [n_t_sf, t_sf, fs_sf] = compute_novelty_sf(x_t, t, fs, win_size, hop_size)
%    Compute spectral flux novelty function.
%
%    Parameters
%    ----------
%    x_t : 1 x T array
%        time domain signal
%    t : 1 x T array
%        time points in seconds
%    fs : int
%        sample rate of x_t (samples per second)
```

```
%    win_size : int
%        window size (in samples)
%    hop_size : int
%        hop size (in samples)
%
%    Returns
%    -------
%    n_t_sf : 1 x L array
%        novelty function
%    t_sf : 1 x L array
%        time points in seconds
%    fs_sf : float
%        sample rate of novelty function

function [onset_a, onset_t, n_t_smoothed, thresh] = ...
    onsets_from_novelty(n_t, t, fs, w_c, medfilt_len, offset)
%    Peak pick a novelty function.
%
%    Parameters
%    ----------
%    n_t : 1 x L array
%        novelty function
%    t : 1 x L array
%        time points of n_t in seconds
%    fs : float
%        sample rate of n_t (samples per second)
%    w_c : float
%        cutoff frequency for Butterworth filter (Hz)
%    medfilt_len : int
%        Length of the median filter used in adaptive threshold. (samples)
%    offset : float
%        Offset in adaptive threshold.
%
%    Returns
%    -------
%    onset_a : 1 x P array
%        onset amplitudes
%    onset_t : 1 x P array
%        time values of detected onsets (seconds)
%    n_t_smoothed : 1 x L array
%        novelty function after smoothing.
%    thresh : 1 x L array
%        adaptive threshold.

function [] = create_novelty_plots(filepath, win_size, hop_size, ...
                                    w_c, medfilt_len, offset) % ground_truth_filepath)
%    Plot waveform, novelty functions, preprocessing steps, and onsets.
%
%    Parameters
%    ----------
%    filepath : string
%        path to a .wav file
%    win_size : int
%        window size for novelty function (in samples)
%    hop_size : int
%        hop size for novelty function (in samples)
%    w_c : float
%        peak picking cutoff frequency for Butterworth filter (Hz)
%    medfilt_len : int
```

```
%         peak picking length of the median filter used in adaptive threshold. (samples)
%     offset : float
%         peak picking offset in adaptive threshold.
%
%     Returns
%     -------
%     None
```