Matthew Schumaker
Charlie Fitzgerald

Acoustic Analysis Script User Manual:

Feature Overview:

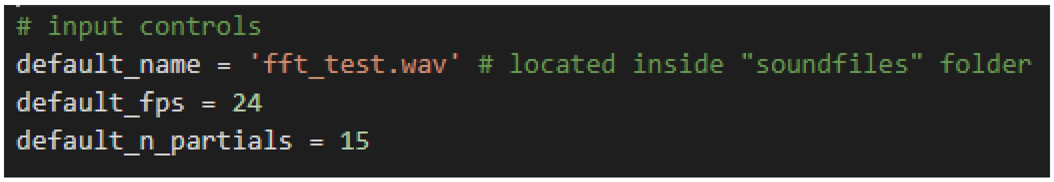
Currently, this script does spectral analysis using Librosa's STFT method as a Fourier Transform agent. Using the data acquired from this, it is capable of picking out the n loudest partials requested by the user, using a spectral denoising algorithm that checks if the frequency is a local maximum (i.e. amplitude for bin is greater than its immediate neighbor frequency bins). These partial frequencies are then written to an output csv file in order of their amplitudes at each moment in time.

The script also performs a spectral centroid calculation, denoising time frames with maximum amplitudes less than 1.

Inner Program Controls:

There are a series of global variables at the top of the script which control the input file, analysis settings and write behavior. The controls are as follows:

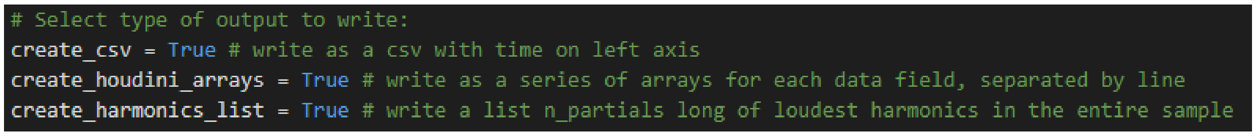
Input Controls:



```
# input controls
default_name = 'fft_test.wav' # located inside "soundfiles" folder
default_fps = 24
default_n_partials = 15
```

default_name includes the file to be analyzed, default_fps describes the number of analyses per second, default_n_partials determines the number of harmonics to include at each moment in time.

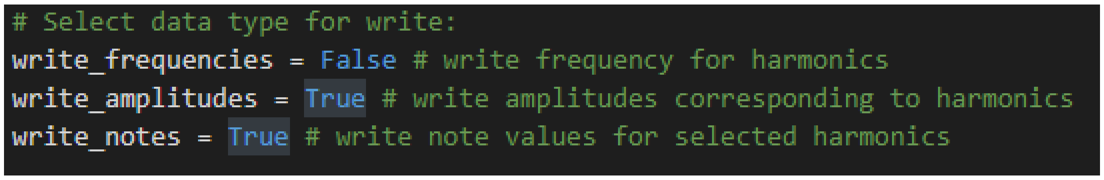
Output Type Controls:



```
# Select type of output to write:
create_csv = True # write as a csv with time on left axis
create_houdini_arrays = True # write as a series of arrays for each data field, separated by line
create_harmonics_list = True # write a list n_partials long of loudest harmonics in the entire sample
```

These 3 boolean variables control which files to generate. create_csv is the default output, writing the output where rows are separated by time with data being represented sequentially horizontally. create_houdini_arrays outputs the desired data in array format, with a separate array for each data field. Data fields will be grouped in 2D arrays, i.e. the harmonic frequencies will be displayed as a series of 1D arrays grouped in time. These arrays are intended for use in houdini per Matt Schumaker's current workflow. create_harmonics_list will write a singular list separated by line of the loudest midi values over the entire sample, with time disregarded. Amplitudes can also be included in this mode, but currently does not support any other data. Any combination of these file outputs should work.

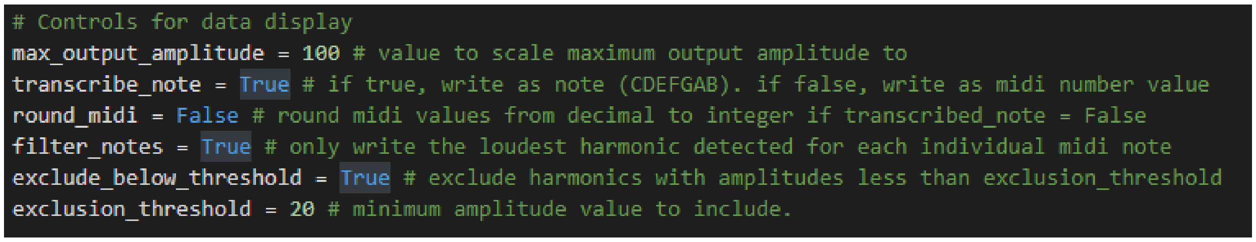
Data Type Controls:



```
# Select data type for write:
write_frequencies = False # write frequency for harmonics
write_amplitudes = True # write amplitudes corresponding to harmonics
write_notes = True # write note values for selected harmonics
```

These values are fairly straightforward, and describe which of the 3 fields of data you would like to output. write_frequencies toggles whether or not write the corresponding frequency for each harmonic. write_amplitudes toggles whether to include amplitudes in the output, and write_notes toggles whether the transcribed midi value for each harmonic frequency is output. It is worth noting the houdini output type will always output all fields.

Data Display Controls:



```
# Controls for data display
max_output_amplitude = 100 # value to scale maximum output amplitude to
transcribe_note = True # if true, write as note (CDEFGAB). if false, write as midi number value
round_midi = False # round midi values from decimal to integer if transcribed_note = False
filter_notes = True # only write the loudest harmonic detected for each individual midi note
exclude_below_threshold = True # exclude harmonics with amplitudes less than exclusion_threshold
exclusion_threshold = 20 # minimum amplitude value to include.
```

These are the most nuanced of the global controls. They work as follows:

- max_output_amplitude
    - describes the value to scale amplitudes to
    - the loudest amplitude in the sample will be represented as a 100 in this case
- transcribe_note
    - toggles whether to write the note for the corresponding harmonic frequency as a note value or a midi number value
- round_midi

    - if transcribe_note is False, round_midi will determine whether or not to round the midi number values or include decimal midi values (to 2 places)
    - this option may cause issues with the list_harmonics mode when False, toggle to True for correct behavior.
- filter_notes
    - if True, harmonics will be filtered such that only the loudest harmonic corresponding to each note value will be printed
    - this may cause the csv to not fill all the way to the right. Currently values are filtered after being collected into the 20 loudest bins, so unfortunately some values will be lost. To adjust for this, simply turn up the default_n_partials value to include more partials in the initial analysis
- exclude_below_threshold
    - if True, harmonics with an amplitude below the exclusion_threshold will be excluded
    - values are scaled to the maximum amplitude, in this case the maximum amplitude is 100 so an exclusion_threshold of 20 will exclude harmonics with an amplitude below 20% of the maximum amplitude

Command Line Controls:

```
basicfft.py <filename> <frames per second> <n loudest partials>
```

The filename, frames per second, and number of partials to include in the output can be controlled from the command line as well. The typical user input is as above.