

1 Introduction

In 2018, two of my favorite bands – The Front Bottoms and Manchester Orchestra – released a song they collaborated on called Allen Town. In a statement to Noisey (Ross, 2018) – the music arm of Vice – Andy Hull of Manchester Orchestra recalled that the creation of this track started when Nate Hussey of All Get Out sent him the first four lines of the track. Andy Hull worked out the melody and music and shared it with Brian Sella of The Front Bottoms, who then helped develop the chorus.

This brings us to an interesting question: which band contributed most to the song?

To attempt to answer this question, we purchased all releases before “Allentown” except joint albums, live albums, and single releases contained in a full album or an Extended Play (EP), a release that is more than a single release but shorter than a full album. We have 42 tracks by All Get Out, 77 by Manchester Orchestra, and 61 by The Front Bottoms. This totals 180 tracks, there are 181 including “Allentown.”

We aimed to use Essentia – an open-source program for music analysis, description, and synthesis – to create data about what each band’s tracks “sound like.” We had trouble installing the program onto Windows and Linux. However, Essentia provides precompiled versions of the program that you can run on individual tracks using the command line¹. Instead of programming, we can call the executable file to process each track (a .WAV file we have).

Below is a sample call to execute the program for a track saved as EXAMPLE.WAV and an output specified as EXAMPLE.json.

```
streaming_extractor_music.exe "EXAMPLE.wav" "EXAMPLE.json"
```

The primary problem with this approach is that I have to write this 181 times – once for each track (one extra for the collaborative track).

One solution to this problem is to create a batch file that contains all 181 commands. Then, we can execute the batch file to process all the songs.

In the tasks below, I provide you with the details necessary for completing this task on a smaller and un-copyrighted set of .WAV files. This lab will enable you to practice (1) installing, loading, and learning to use libraries; (2) working with character objects; (3) coding `for()` loops; and (4) accessing elements of vectors and lists.

2 Task 1: Build a Batch File for Data Processing

Write code that will explore a file directory called MUSIC. The first level of subdirectories are artists, and the second level are albums. The files will be in the album subdirectories and generally have the format [track number]-[artist]-[track name].WAV.

Step 0: Download the fake directory of songs [here](#). Also, install the `stringr` package for R (Wickham, 2023).

Step 1: Find the command required to list all of the directories in a specific directory and use it to find all subdirectories of MUSIC/.

Step 2: Use the `str_count()` function to count the number of forward slashes (/) in each directory. Use this count to subset all album subdirectories from the vector of all subdirectories in **Step 1**.

Step 3: For each album subdirectory, complete the following tasks.

¹<https://essentia.upf.edu/extractors/>

1. Find the command required to list all the files in a specific directory and use it to find all files in the current album subdirectory.
2. Use the `str_count()` function to count the number of times `.WAV` occurs in each file. Use this count to subset all `.WAV` files from the current album subdirectory from the vector of all files in step (a).
3. Create an empty vector called `code.to.process`. We will fill this by completing the following for each track in the current album subdirectory.
 - (a) Create an object containing the track file location using the `paste()` function to paste the current album subdirectory and the current track title together. Add a quotation mark to the beginning and end, and make sure you are careful about spaces.
 - (b) Create an object containing the current track's filename. Use `str_sub()` to remove the `.WAV` at the end of the current track. Next, use the naming convention of files to extract *just* the track name using `str_split()`.
 - (c) Create an object containing the desired output file. We would like it to be `[artist name]-[album name]-[track name].json`.
 - (d) Use the `paste()` function to paste `streaming_extractor_music.exe` the track file and the desired output filename together to create the command line prompt for the current track. Save the result to the `code.to.process` vector.

Step 4: Use the `writeln()` function to write the `code.to.process` vector to a `.txt` file called `batfile.txt`.

3 Task 2: Process JSON Output

As an example, I have provided the `.JSON` output for the song `Au Revoir (Adios)` on the `Talon Of The Hawk` album by `The Front Bottoms`.

Step 0: Install the `jsonlite` package for R (Ooms, 2014).

Step 1: Use the `str_split()` function to extract the `artist`, `album`, and `track` from the file name. Note that while you can do this manually for one file, we will want to automate this process for all files later.

Step 2: Load the JSON file into R using the `fromJSON()` function. The resulting object is a very large list.

Step 3: Extract the `average_loudness`, the mean of `spectral_energy`, `danceability`, `bpm` (tempo in beats per minute), `key_key` (musical key), `key_scale` (musical mode), `length` (duration in seconds). You will find the documentation [here](#) rather helpful as you explore the list object.

References

- Ooms, J. (2014). The `jsonlite` package: A practical and consistent mapping between json data and r objects. *arXiv:1403.2805 [stat.CO]*.
- Ross, A. R. (2018). Manchester orchestra and the front bottoms are finally together on “allentown”. <https://www.vice.com/en/article/manchester-orchestra-and-the-front-bottoms-are-finally-together-on-allentown/>.
- Wickham, H. (2023). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.5.1.