

Lab 2 – MATH 240 – Computational Statistics

Ben Horner
Colgate University
Math Department
bhorner@colgate.edu

February 6, 2025

Abstract

Keywords: Batch File, Data Processing, Music, JSON

1 Introduction

In 2018, two of Professor Cipolli's favorite bands - The Front Bottoms and Manchester Orchestra - released a song they collaborated on called Allen Town. In a statement to Noisey () - 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?

1.1 Task

To attempt to answer this question, we first need to create a batch file for data processing, aiming to use Essentia - an open-source program for music analysis, description, and synthesis - to create data about what each band's tracks "sound like." In this Lab, we will begin by creating the code for building this batch file using a set of example .wav files. Then we will process the .JSON output and extract key descriptors.

2 Methods

We divide our overall goal in to two tasks. **Task 1**, where we build the batch files for data processing, and **Task 2**, where we process the data. For this Lab, we use test files created by Professor Cipolli located in a directory labeled "MUSIC."

2.1 Task 1: Building a Batch File for Data Processing

To write the code to create the desired batch file, we will need to convert them from the format of [track number]-[artist]-[track name].wav to [artist name]-[album name]-[track name].json.

1. First, we install the `stringr` package for R (). We will be using it to split the file name apart to isolate the artist and album and to change the file to .json.

```
library(stringr)
```

2. Next, we isolate the subdirectories (artists and albums) and files (songs) within the MUSIC directory using the `list_dirs`, `list_files` and the `str_count()` function to count the number of forward slashes.

```
artist.dirs = c()
album.dirs = c()
for (directory in music.dirs){
  dir.level <- str_count(directory, pattern = "/")
  if (dir.level == 1){
    artist.dirs = append(artist.dirs, directory)
  }
  else if (dir.level == 2){
    album.dirs = append(album.dirs, directory)
  }
}
```

3. For each album, we isolated each .wav file using `str_count()` to subset all .wav files from the current album subdirectory. For each file (song) we used `str_split()` to extract just the track name, which we then pasted (`paste()` function) together with the artist name, track name, and .json to create an object of [artist name]-[album name]-[track name].json. We also pasted `streaming_extractor_music.exe` to the file name to create the command line prompt for the current track.

```
json.files = c()
for (album in album.dirs){
  ##Splits the string of the directory into Music, Artist, Album
  split.string = str_split(album, pattern = "/", simplify = TRUE)
  album.name = split.string[1, 3]
  artist.name = split.string[1, 2]
  art.n.album = paste(artist.name, album.name, sep = "-") #artist and album
  #comined in desired format

  ##Lists the music files in each album, goes through them, creates the final
  #desired string of [artist name]-[album name]-[trackname].json
  music.files <- list_files(album)
  for (music.file in music.files){
    if (str_count(music.file, pattern = ".wav")){
      #Makes sure we are only dealing with .wav files
      file.path = str_split(music.file, pattern = ".wav", simplify = TRUE)
      file.path = str_split(file.path, pattern = "-", simplify = TRUE)
      song.name = file.path[1, 3]
      final.path = paste(art.n.album, "-", song.name, ".json", sep = "")
      json.files = append(wav.files, final.path) #will contain all of the .
    }
  }
}

code.to.process = paste(json.files, "streaming_extractor_music.exe")
```

4. Finally, using the `writeLines()` function, we wrote the file names to a .txt file called batfile.txt.

Describe the data you are working with, if applicable. Describe the specific process you will follow to answer the question at hand. It should provide a clear and concise narrative that flows from the problem specification in the Introduction to how you will approach answering it. This is where I would expect to see some citations for R packages you will use to conduct the statistical analysis reported in the Results section.

3 Results

Tie together the Introduction – where you introduce the problem at hand – and the methods – what you propose to do to answer the question. Present your data, the results of your analyses, and how each reported aspect contributes to answering the question. This section should include table(s), statistic(s), and graphical displays. Make sure to put the results in a sensible order and that each result contributes a logical and developed solution. It should not just be a list. Avoid being repetitive.

3.1 Results Subsection

Subsections can be helpful for the Results section, too. This can be particularly helpful if you have different questions to answer.

4 Discussion

You should objectively evaluate the evidence you found in the data. Do not embellish or wish-terpet (my made-up phrase for making an interpretation you, or the researcher, wants to be true without the data *actually* supporting it). Connect your findings to the existing information you provided in the Introduction.

Finally, provide some concluding remarks that tie together the entire paper. Think of the last part of the results as abstract-like. Tell the reader what they just consumed – what’s the takeaway message?

Bibliography: Note that when you add citations to your bib.bib file *and* you cite them in your document, the bibliography section will automatically populate here.

References

5 Appendix

If you have anything extra, you can add it here in the appendix. This can include images or tables that don't work well in the two-page setup, code snippets you might want to share, etc.