soundbendor lab

AlgoRhythms

Workout with Your Own Music

Dalton Shults
Kameron Keller
Dr. Patrick Donnelly

Oregon State University

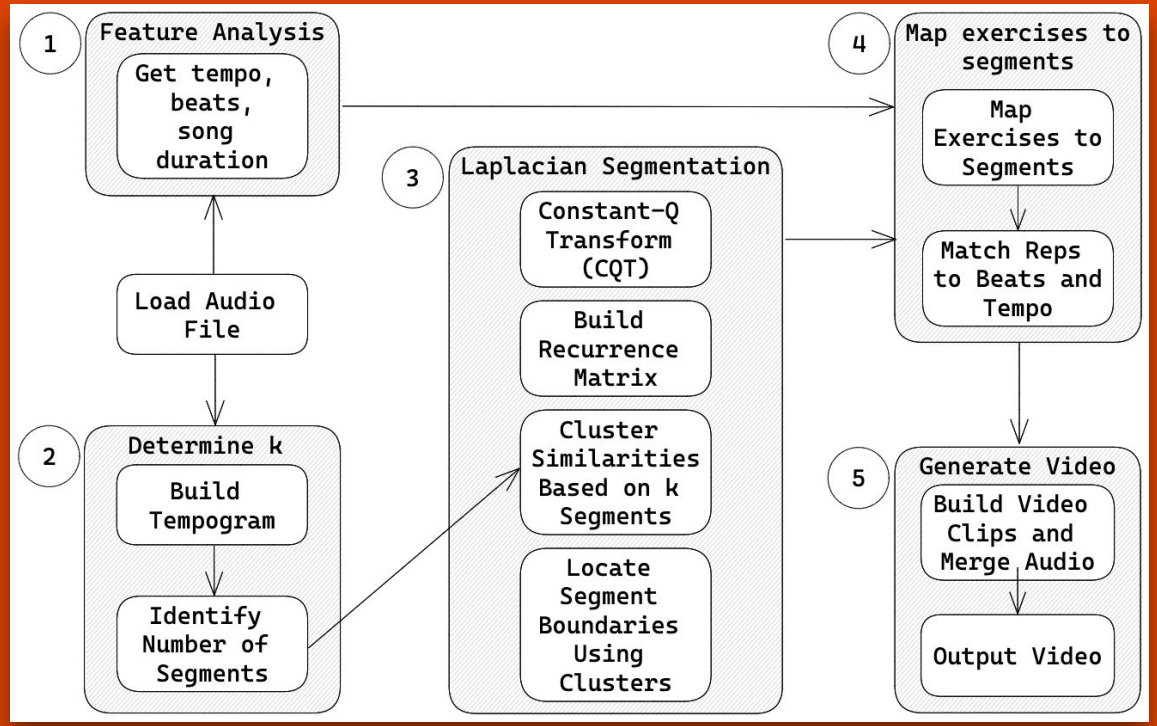
What is AlgoRhythms?

- A Python pipeline that generates a workout from a given song file (WAV/MP3)
- It uses machine learning to map workouts from a set of exercises to identified sections of a song
  - Attempts to match the rhythm of each song to the workout



Overview

1. Feature Analysis
   a. Get tempo
   b. Beats
   c. Song duration
2. Determine the number of clusters (*K*)
   a. Build tempogram
   b. Identify *K*
3. Laplacian Segmentation
   a. Constant-Q Transform
   b. Build a recurrence matrix
   c. Cluster similarities w/ *K* segments
   d. Locate segment boundaries using clusters
4. Map Exercises to segments
   a. Map exercises to segments
   b. Match reps to beats and tempo
5. Generate a video
   a. Build video clips and merge audio
   b. Output video

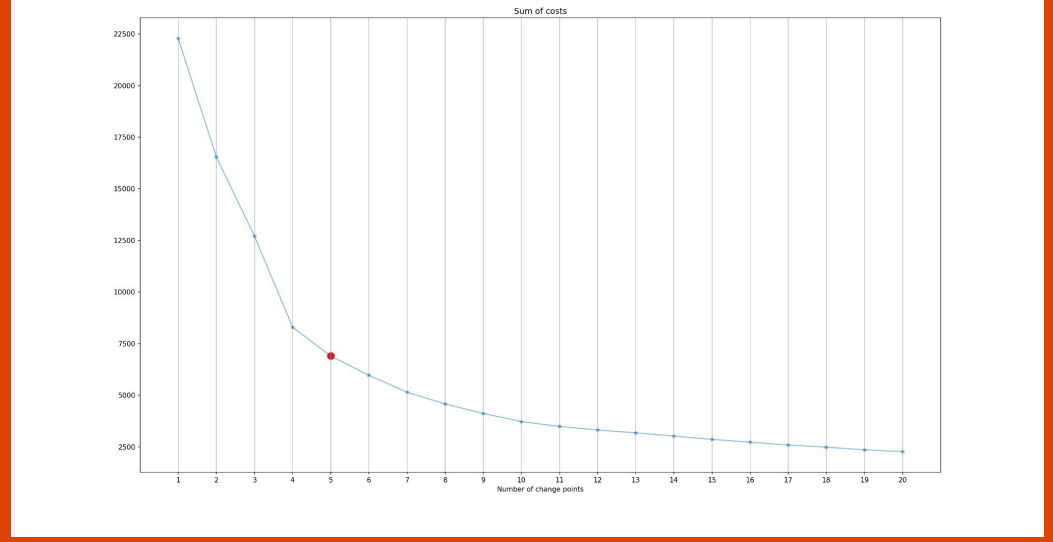


Feature Analysis

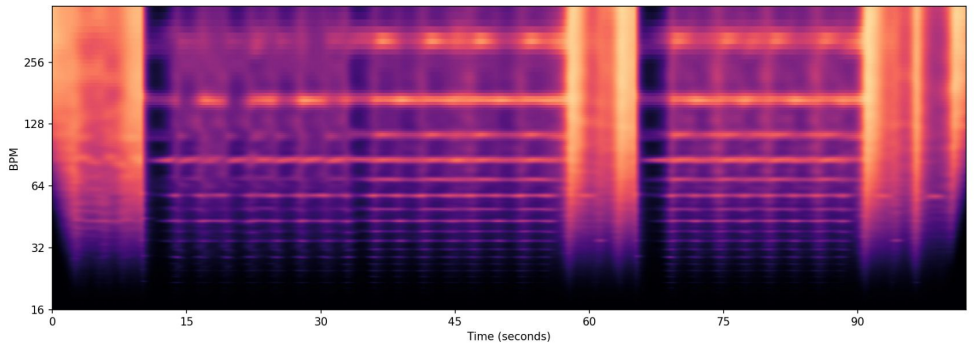
- `Librosa` is used to identify features of the song
  - Load Song
    - Get sampling rate
  - Tempo
  - Beats
  - Song duration



Determine *K*



Knee Graph: Drop It

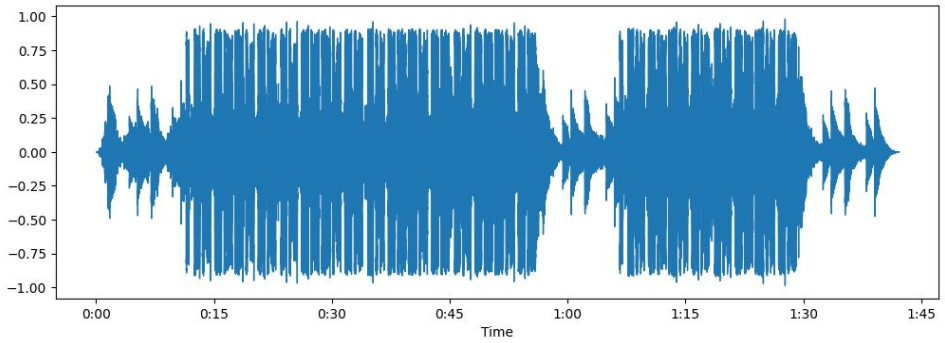


Tempogram: Drop It

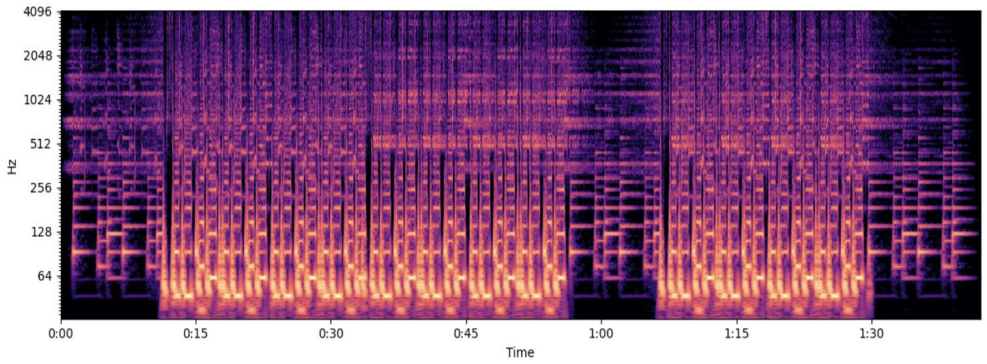
- Goal: Determine how many clusters to use to segment the song
- How we did it:
  - We created a tempogram from the song file loaded by `Librosa`
  - We iterated through different values of *K* and measured the sum of costs for each value to generate a knee graph
  - We passed the knee graph data to Kneedle algorithm to select the optimal value for *K*

Laplacian Segmentation: Constant-Q Transform

- Goal: Move from a waveform (time series) signal into a frequency domain.
  - Move from one domain to another
- We used Constant-Q Transform (CQT) for the conversion
  - "Amplitude/Phase against a log frequency"
    - Creates fewer frequency bins
  - "Reduces frequency resolution"
    - Mirrors what the human ear can detect, and reduces the song to those frequencies



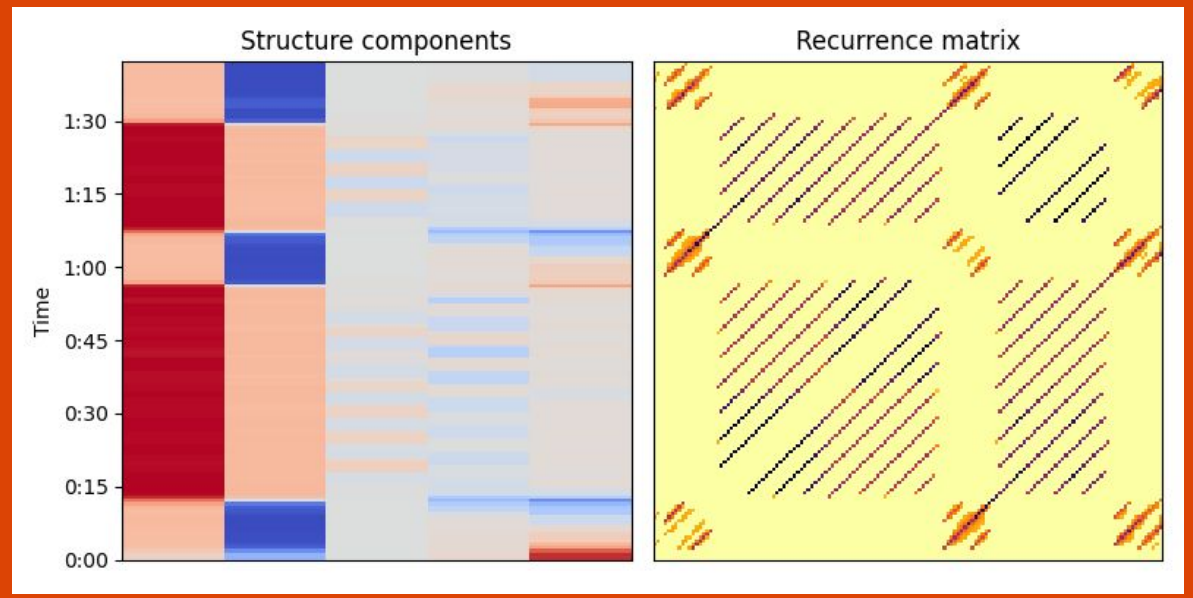
Wave Form: Drop It



Spectral Form, after (CQT): Drop It

Laplacian Segmentation: Recurrence Matrix & Clustering

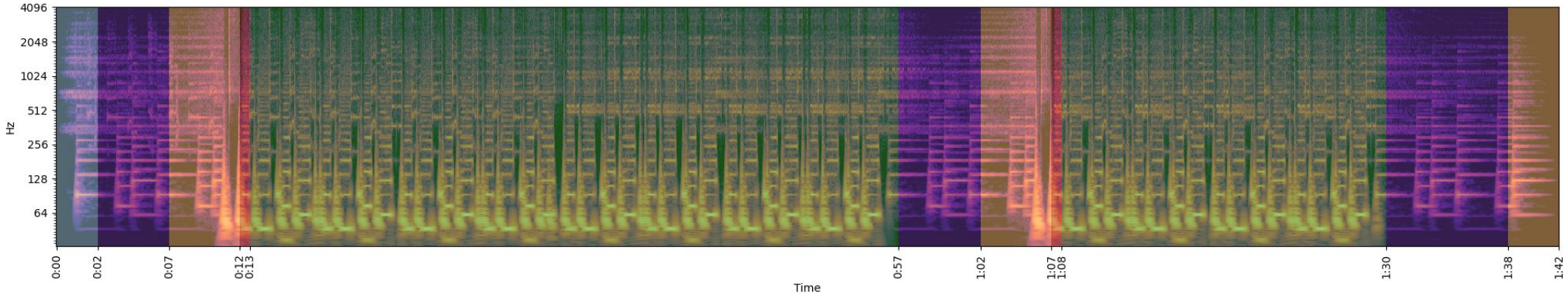
- Goal: Locate the segments of the song, and label the clusters
- A recurrence matrix is built from the entire song
  - Allows for the comparison of each index of the song with each other index of the song
    - Creates a matrix where similar sections are identified
  - Filter the recurrence matrix so that each section can be identified.
- A normalized Laplacian matrix is created from the recurrence matrix
  - eigenvectors are generated



Hierarchical Clustering & Recurrence Matrix: Drop It

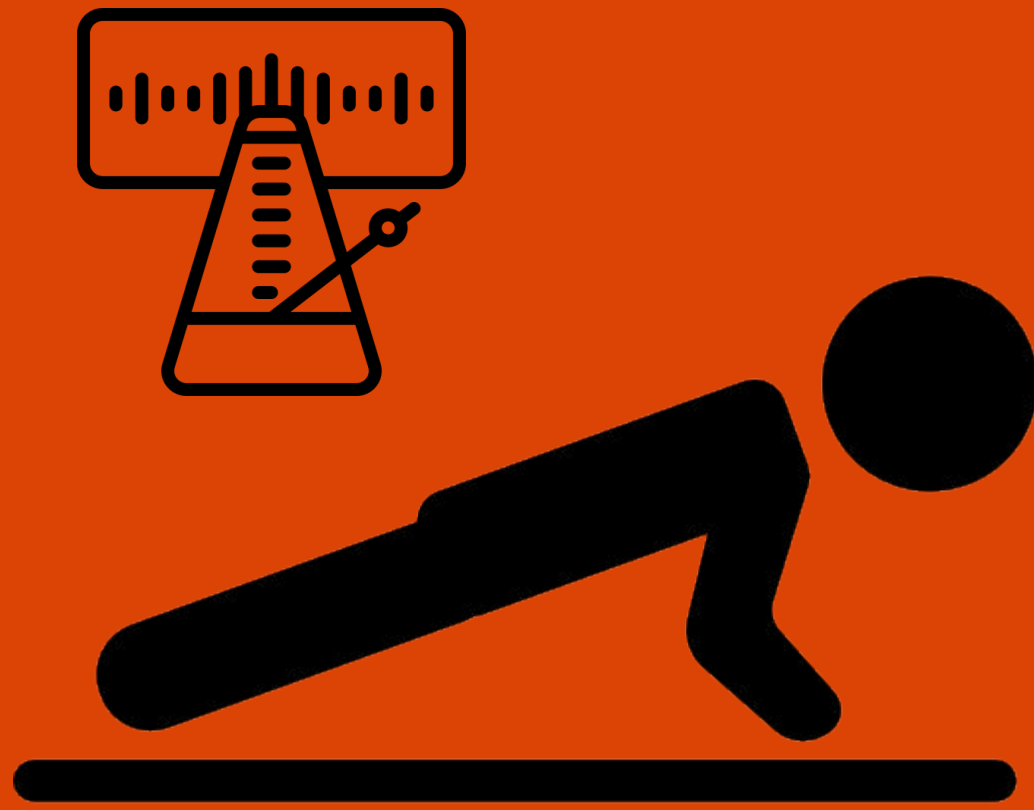
Laplacian Segmentation: K-Means and Overlay w/ Clusters

- Goal: Identify the location of each segment in the song
- *K*-Means algorithm is used to analyze the eigenvectors
    - Generates:
        - Cluster Labels
        - Segments
- Clusters labels identify the similar segments of the song
    - Allows for analyzing the features of each individual segment
- An array of cluster labels generated by the *K*-Means algorithm provides the ordering and an identity for each cluster
    - Used to locate segment boundaries



Mapping Exercises onto Segments

- Exercises are mapped to the segments
  - Exercises chosen at random
- Exercise Data:
  - Generated our exercise data
    - Timed reps for a given workout, and generated data from a normal distribution



Video Generation

- Library used: `MoviePy`
- A video for each segment is generated
  - Using
    1. Mapped exercises
    2. Extracted features (beat locations and BPM)
- For each segment:
  - Create a video clip for the duration of the segment.
    - Include workout title, image, and up-next exercise
    - Identify the number of beats per rep that provides the closest match to the rep duration
    - Display a count down when the segment has less than 10 seconds remaining
- Merge clips



Outcomes

- Python Pipeline
  - Automated program that receives any WAV/MP3 song file and generates a custom workout video for that song
- Machine Learning
  - Successfully used to identify important information about the song
- Automated Video Generation





What's Next?

New Features

- Database for storing data of analyzed music
  - Improves performance of the application
- Workout difficulty settings
- Muscle group selection
- User interface for application
- Generate a text version of the workout
- Save workouts for later

Improvements

- New animations
- Optimize *K* selection
- Move out of Jupyter Notebook
- Duration selection
- Adding rests
- Handling segments that are too short
- Giving the user transition time

Paths

- Move away from video and towards a user interface
- Web Application/Mobile Application
- Linking multiple songs together
- Spotify/Apple Music/YouTube integration

Sources

C. Truong, L. Oudre, N. Vayatis. Selective review of offline change point detection methods.*Signal Processing*, 167:107299, 2020

Free icons and stickers - millions of images to download. Flaticon. (n.d.). https://www.flaticon.com/

McFee, B., & Ellis, D.P. (2014). Analyzing Song Structure with Spectral Clustering. International Society for Music Information Retrieval Conference, 3-6.

Spotebi. (2021, August 19). *Illustrated Exercise Guide*. Spotebi: Fitness and Nutrition. https://www.spotebi.com/exercise-guide/

THANK YOU



Oregon State
University