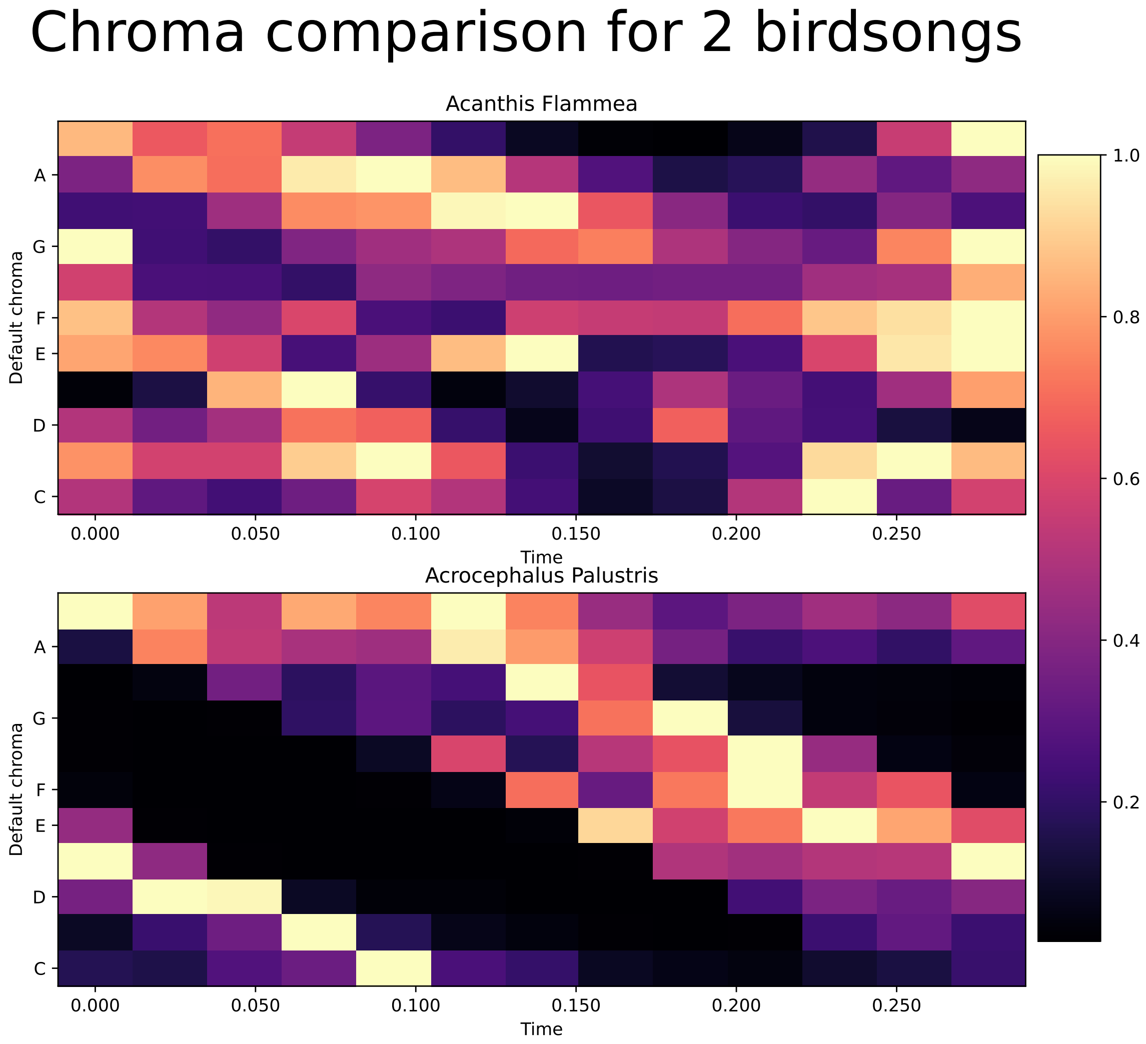
Log

```
#!/usr/bin/env python3

"""
Script for cleaning and visualising the provided birdsong data.
"""
```

```
__author__ = "Niek Scholten"

# Imports
import librosa
from librosa import display
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

with open('data/train.csv', newline='') as csvfile:  # Open CSV and load the data into numpy
    my_data_train = np.genfromtxt(csvfile, delimiter=',')
    csvfile.close()
# Delete the last columns containing other data
my_data_train = np.delete(my_data_train, slice(-1, -16, -1), axis=1)
my_data_train = np.delete(my_data_train, 0, axis=1)  # Delete ID's
my_data_train = np.delete(my_data_train, 0, axis=0)  # Delete column names

with open('data/test.csv', newline='') as csvfile:  # Open CSV and load the data into numpy
    my_data_test = np.genfromtxt(csvfile, delimiter=',')
    csvfile.close()
# Delete the last columns containing other data
my_data_test = np.delete(my_data_test, slice(-1, -16, -1), axis=1)
my_data_test = np.delete(my_data_test, 0, axis=1)  # Delete ID's
my_data_test = np.delete(my_data_test, 0, axis=0)  # Delete column names

with open('data/train.csv', newline='') as csvfile:  # Open CSV and load the column names into numpy
    my_column_names = np.genfromtxt(csvfile, dtype=str, delimiter=',', skip_footer=1760)
    csvfile.close()
# Delete the last columns containing other data
my_column_names = np.delete(my_column_names, slice(-1, -16, -1))
my_column_names = np.delete(my_column_names, 0)  # Delete ID's

# The given data was sorted by alphabetical order, but this results in broken sequences
# Rearrange the data to the correct format for librosa
index = [0, 39, 52, 65, 78, 91, 104, 117, 130, 143, 13, 26,  # Chromogram 1
         1, 40, 53, 66, 79, 92, 105, 118, 131, 144, 14, 27,  # Chromogram 2
         5, 44, 57, 70, 83, 96, 109, 122, 135, 148, 18, 31,  # Chromogram 3
         6, 45, 58, 71, 84, 97, 110, 123, 136, 149, 19, 32,  # Chromogram 4
         7, 46, 59, 72, 85, 98, 111, 124, 137, 150, 20, 33,  # Chromogram 5
         8, 47, 60, 73, 86, 99, 112, 125, 138, 151, 21, 34,  # Chromogram 6
         9, 48, 61, 74, 87, 100, 113, 126, 139, 152, 22, 35,  # Chromogram 7
         10, 49, 62, 75, 88, 101, 114, 127, 140, 153, 23, 36,  # Chromogram 8
         11, 50, 63, 76, 89, 102, 115, 128, 141, 154, 24, 37,  # Chromogram 9
         12, 51, 64, 77, 90, 103, 116, 129, 142, 155, 25, 38,  # Chromogram 10
         2, 41, 54, 67, 80, 93, 106, 119, 132, 145, 15, 28,  # Chromogram 11
         3, 42, 55, 68, 81, 94, 107, 120, 133, 146, 16, 29,  # Chromogram 12
         4, 43, 56, 69, 82, 95, 108, 121, 134, 147, 17, 30]  # Chromogram 13
my_data_train = my_data_train[:, index]  # Apply index to the train data
my_data_test = my_data_test[:, index]  # Apply index to the test data
my_column_names = my_column_names[index]  # Apply the index to the collumn names

flammea_1 = np.empty((12, 13), int)  # Create empty array for this birdsong
# Add multiple columns form the original data as a new row
flammea_1 = np.append(flammea_1, my_data_train[0:1, 0:13], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 13:26], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 26:39], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 39:52], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 52:65], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 65:78], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 78:91], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 91:104], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 104:117], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 117:130], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 130:143], axis=0)
flammea_1 = np.append(flammea_1, my_data_train[0:1, 143:], axis=0)

flammea_1 = np.delete(flammea_1, slice(0, 13), axis=0)  # Delete empty cells

palustris_1 = np.empty((12, 13), int)  # Create empty array for this birdsong
# Add multiple columns form the original data as a new row
palustris_1 = np.append(palustris_1, my_data_train[20:21, 0:13], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 13:26], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 26:39], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 39:52], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 52:65], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 65:78], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 78:91], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 91:104], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 104:117], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 117:130], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 130:143], axis=0)
palustris_1 = np.append(palustris_1, my_data_train[20:21, 143:], axis=0)

palustris_1 = np.delete(palustris_1, slice(0, 13), axis=0)  # Delete empty cells

fig, ax = plt.subplots(nrows=2, figsize=(10, 9))  # Create empty canvas for plots
img1 = librosa.display.specshow(flammea_1, y_axis='chroma', x_axis='time', ax=ax[0])
ax[0].set_title('Acanthis Flammea')
ax[0].set(ylabel='Default chroma')
ax[0].set(xlabel='Time')

img2 = librosa.display.specshow(palustris_1, y_axis='chroma', x_axis='time', ax=ax[1])
ax[1].set_title('Acrocephalus Palustris')
ax[1].set(ylabel='Default chroma')
```

```
ax[1].set(xlabel='Time')

cbar_ax = fig.add_axes([0.91, 0.15, 0.05, 0.7])  # Set axis for the colorbar
fig.colorbar(mappable=img1, cax=cbar_ax)
```

```
fig.suptitle('Chroma comparison for 2 birdsongs', fontsize=32)

df = pd.DataFrame(my_data_train, columns=my_column_names)  # Export clean training data to csv
df.to_csv('data/dataframe_train.csv', index=False, header=True, sep=',')

df = pd.DataFrame(my_data_test, columns=my_column_names)  # Export clean testing data to csv
df.to_csv('data/dataframe_test.csv', index=False, header=True, sep=',')

plt.show()
```



This figure shows the chroma signature comparison for 2 fragments of different bird-species songs. It was created by transforming the input data to the correct format that is normally outputted by librosa, because the sound data is created using librosa.