Music Generation using Markov Chains

Group Members -
Adish Jain (2021227)
Anand (2020280)
Nikhil Suri (2021268)
Rahul Raj (2020322)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Index

Introduction

- A Markov Chain is a statistical model that describes a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. This model has been highly effective in various fields, such as finance & natural language processing.

- In music, this translates to sequences of notes or chords. By analyzing patterns and probabilities, we can generate new music pieces that maintain the original's structure.

- This blend of creativity and computation opens new frontiers in music technology, demonstrating how mathematical models can generate art.

Motivation

- Unique Music Creation: Markov Chains allow for the generation of new and unique pieces of music, providing a fresh perspective on music creation.

- Versatility: Depending on the training data used, Markov Chains can generate music in a wide variety of styles, from classical to rock to jazz.

- Computational Efficiency: Markov Chains offer a simple and effective model for music generation that is not computationally expensive, making it feasible for real-time applications and accessible to a wide range of developers.

- Potential Therapeutic Applications: The ability to generate unique, soothing melodies can potentially contribute to the field of music therapy, with Markov Chains playing a role in personalizing music to individual's preferences or therapeutic needs.

- Machine Learning and AI: The application of Markov Chains in music generation expands our understanding of how statistical models can be applied to creative processes.

DataSet Description

- Firstly we have created a CSV input file that only contains the chords / notes, that we need work on. Eg: C#. These notes were taken from the song - Hello by Adele.
- Since there's no readily available dataset specifically for our purpose, we had to manually extract the musical notes & save them to a CSV file, which was then used for processing using Python & MIDI module.
- MIDI note numbers are a way of representing musical notes in the MIDI (Musical Instrument Digital Interface) protocol, which is a standard for communicating musical information between digital devices like synthesizers and computers.
- In MIDI, each note is assigned a specific number from 0 to 127. This range corresponds to a range of pitches, with each number representing a specific pitch. The number 72, for example, represents the note Middle C. The numbers increase as the pitch gets higher, so the number 73 would represent the note C# above Middle C, and so on.
- We're essentially mapping individual notes, to their corresponding MIDI values, using Python.

```
# Define a mapping from chord names to MIDI note numbers
chord_mapping = {
    'C': 72,
    'C#': 73, 'C#m': 73,
    'Db': 73, 'Dbm': 73,
    'D': 74, 'Dm': 74,
    'D#': 75, 'D#m': 75,
    'Eb': 75, 'Ebm': 75,
    'E': 76, 'Em': 76,
    'F': 77, 'Fm': 77,
    'F#': 78, 'F#m': 78,
    'Gb': 78, 'Gbm': 78,
    'G': 79, 'Gm': 79,
    'G#': 80, 'G#m': 80,
    'Ab': 80, 'Abm': 80,
    'A': 81, 'Am': 81,
    'A#': 82, 'A#m': 82,
    'Bb': 82, 'Bbm': 82,
    'B': 83, 'Bm': 83,
}
```

	A	B
1	chords	
2	F	
3	Em	
4	A	
5	Dm	
6	Dm	
7	Bb	
8	C	
9	F	
10	C	
11	Dm	

Pre – Processing

- The data is a one big sequence of chords / musical notes (states).

- As explained earlier, we manually took the musical notes from this website.

- Then, we saved the musical notes obtained earlier, into a CSV file, where each note is supposed to be in a 'newline'.

- This is the standard pre-processing we've done, for both the left_hand_notes & right_hand_notes.

Modelling the Data as Markov Chain

- The data can be modelled as Markov Chain as in a melody / song, the next chord depends on the past chords.

- The chords are the states of the Markov Chain.

- We've modelled the data, both as a Higher-Order Markov Chain, & also as a First-Order Markov Chain.

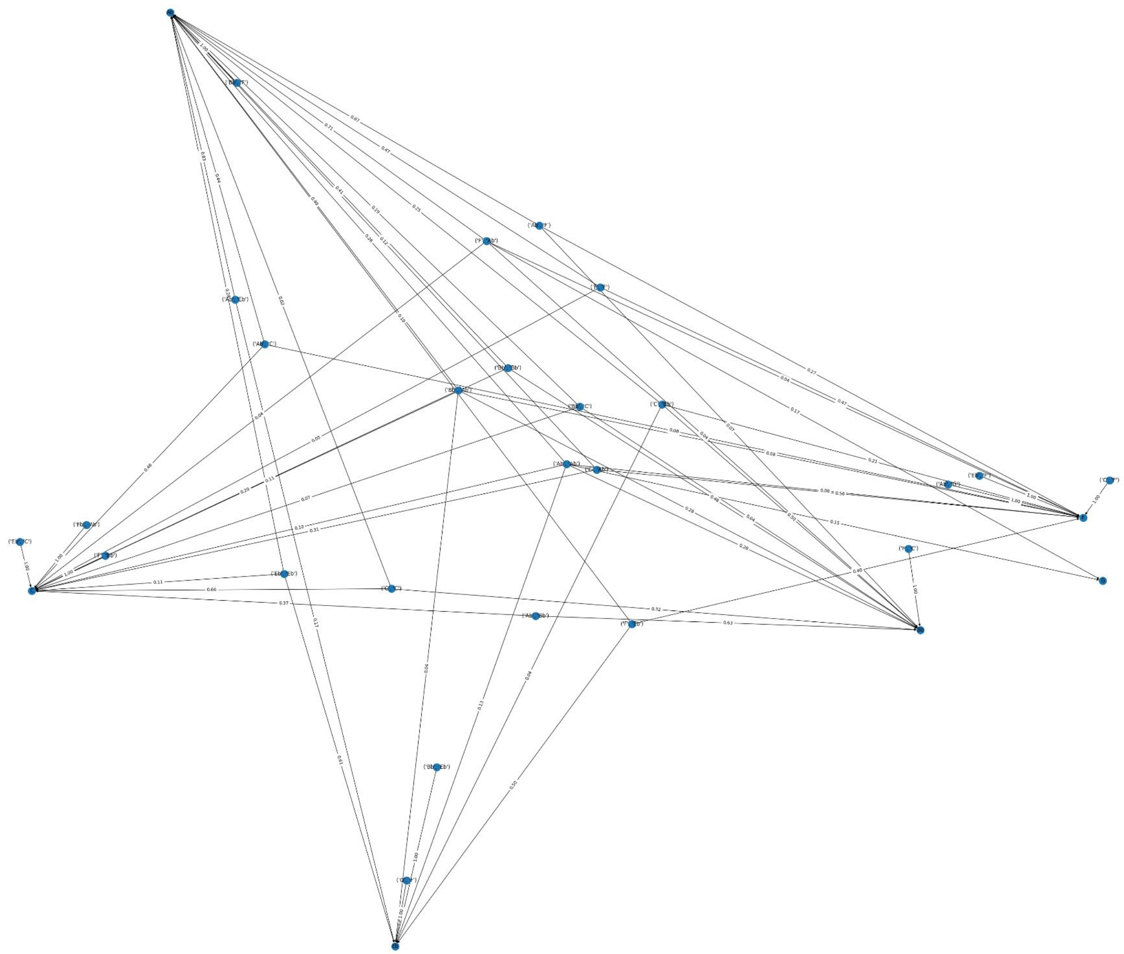
- For the Higher-Order Markov Chain part, we've kept the order = 2, that is, the next state depends on previous 2 states

Reasons for Modeling as HOMC

- The need to model the data, as a Higher-Order Markov Chain (HOMC), was felt necessary, since the Music Output from the First-Order Markov Chain didn't sound great.
- Higher-order Markov chains consider longer sequences of events or states. In music, this allows the model to capture longer-term dependencies between chords or notes, leading to more coherent and structured compositions.
- Music often exhibits patterns that extend beyond the immediate preceding events. Higher-order Markov chains provide a mechanism to capture and replicate these larger musical structures, such as chord progressions or motifs.
- Higher-order Markov chains can help mitigate the issue of repetitiveness that may arise in lower-order models.
- Put simply, Higher-Order Markov Chains sound better!

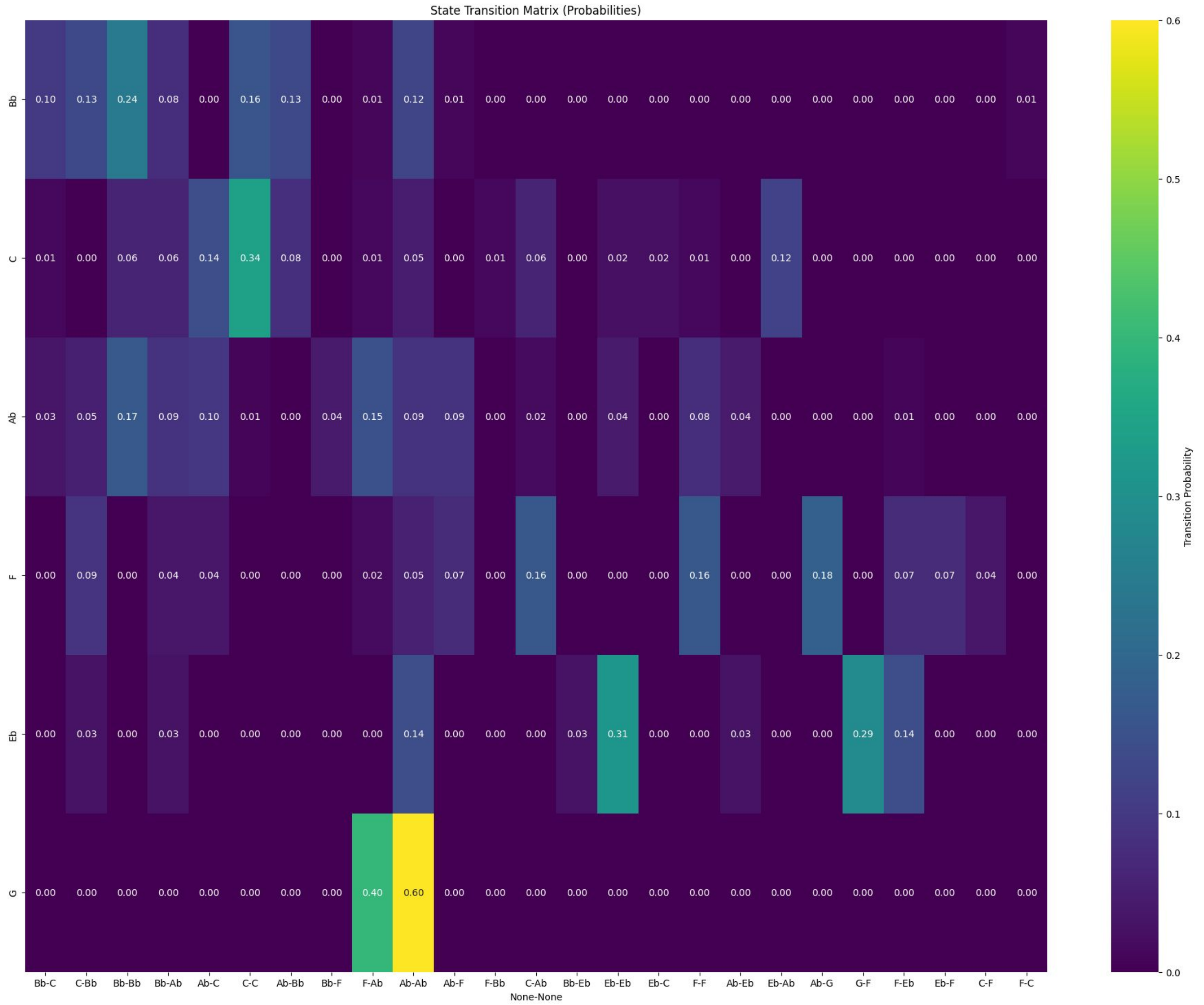
Observations

- State Transition Diagram for Higher-Order Markov Chain, for the right_hand_notes.



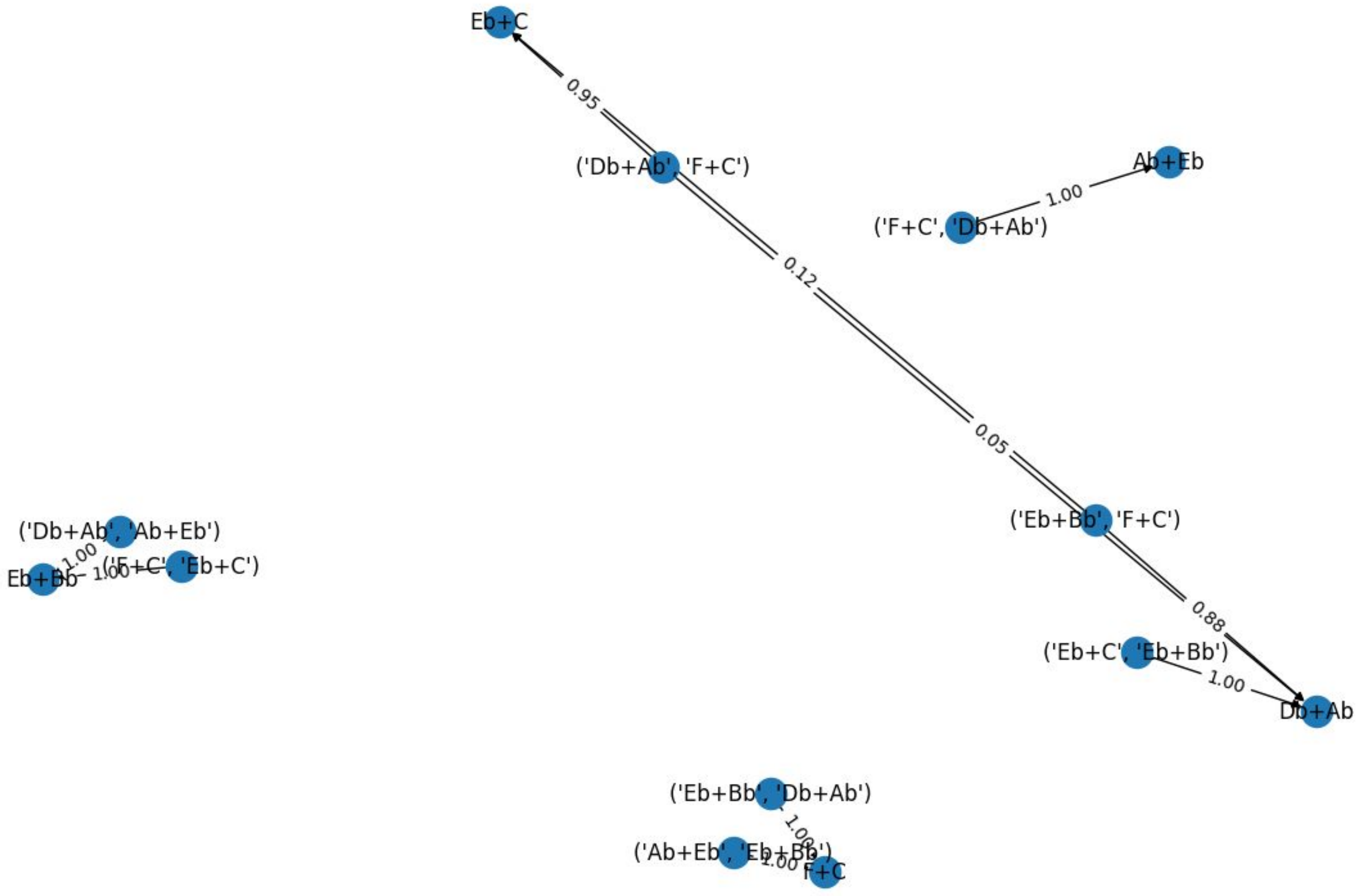
Observations

- State Transition Matrix for Higher-Order Markov Chain, for the right_hand_notes.



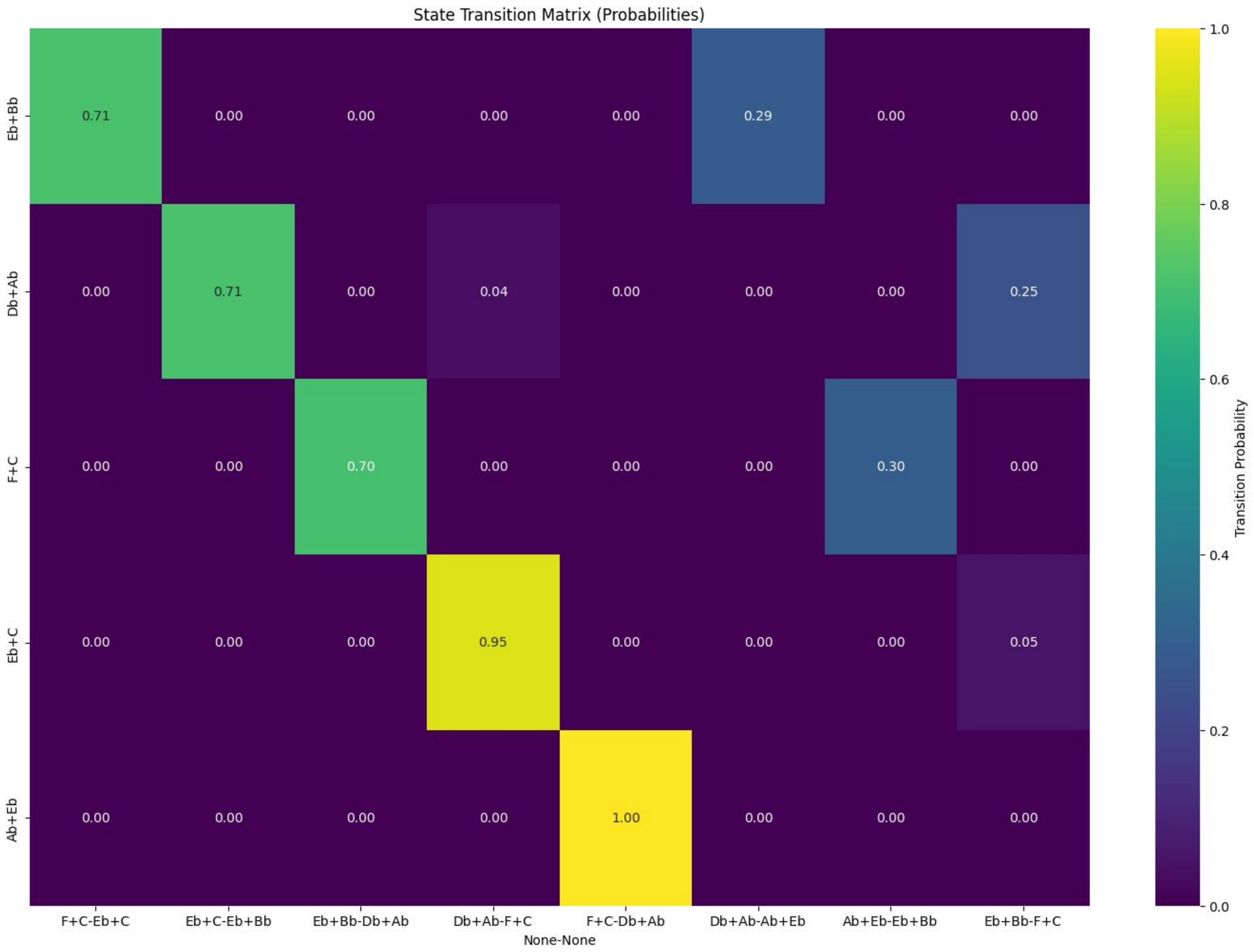
Observations

- State Transition Diagram for Higher-Order Markov Chain, for the left_hand_notes.



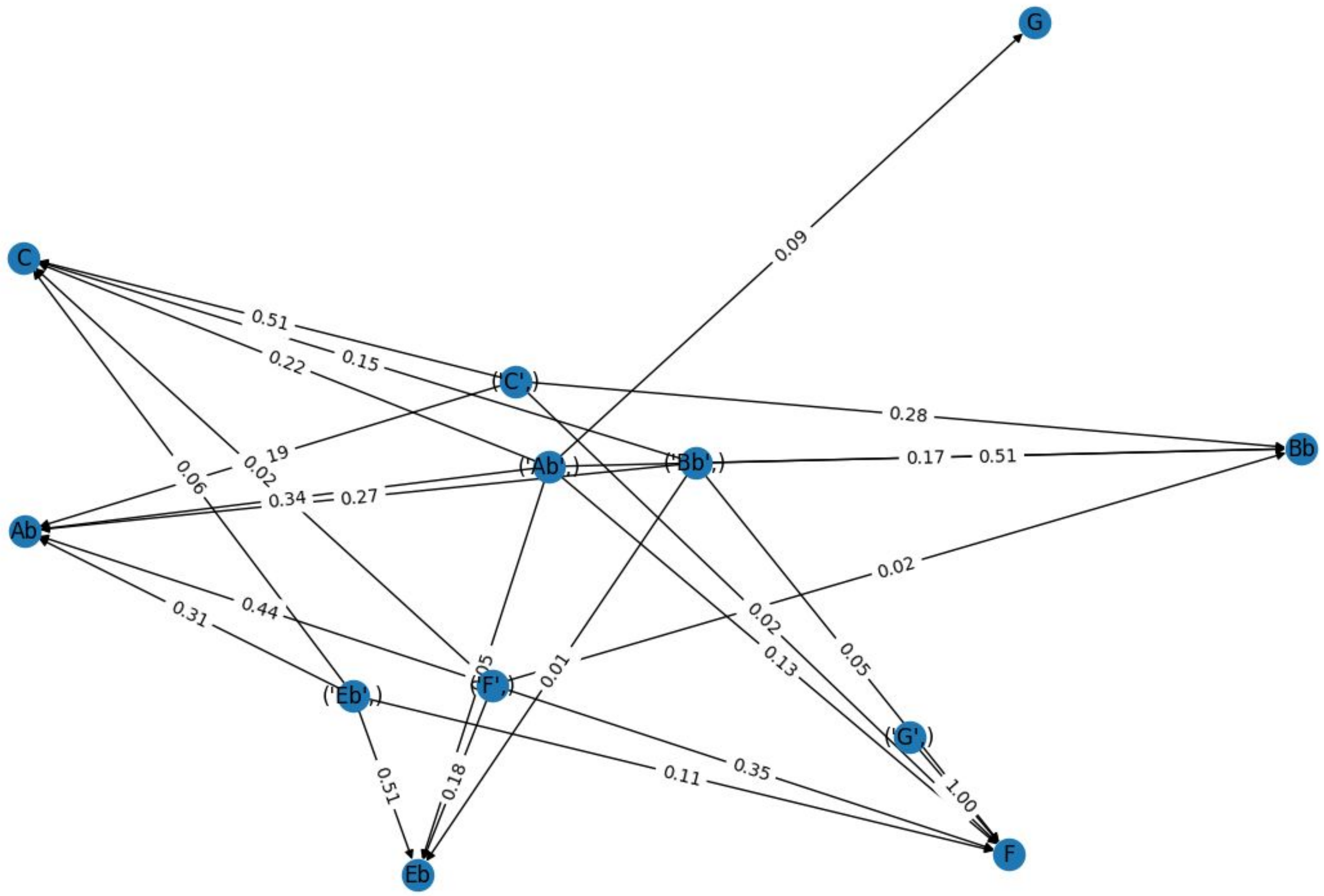
Observations

- State Transition Matrix for Higher-Order Markov Chain, for the left_hand_notes.



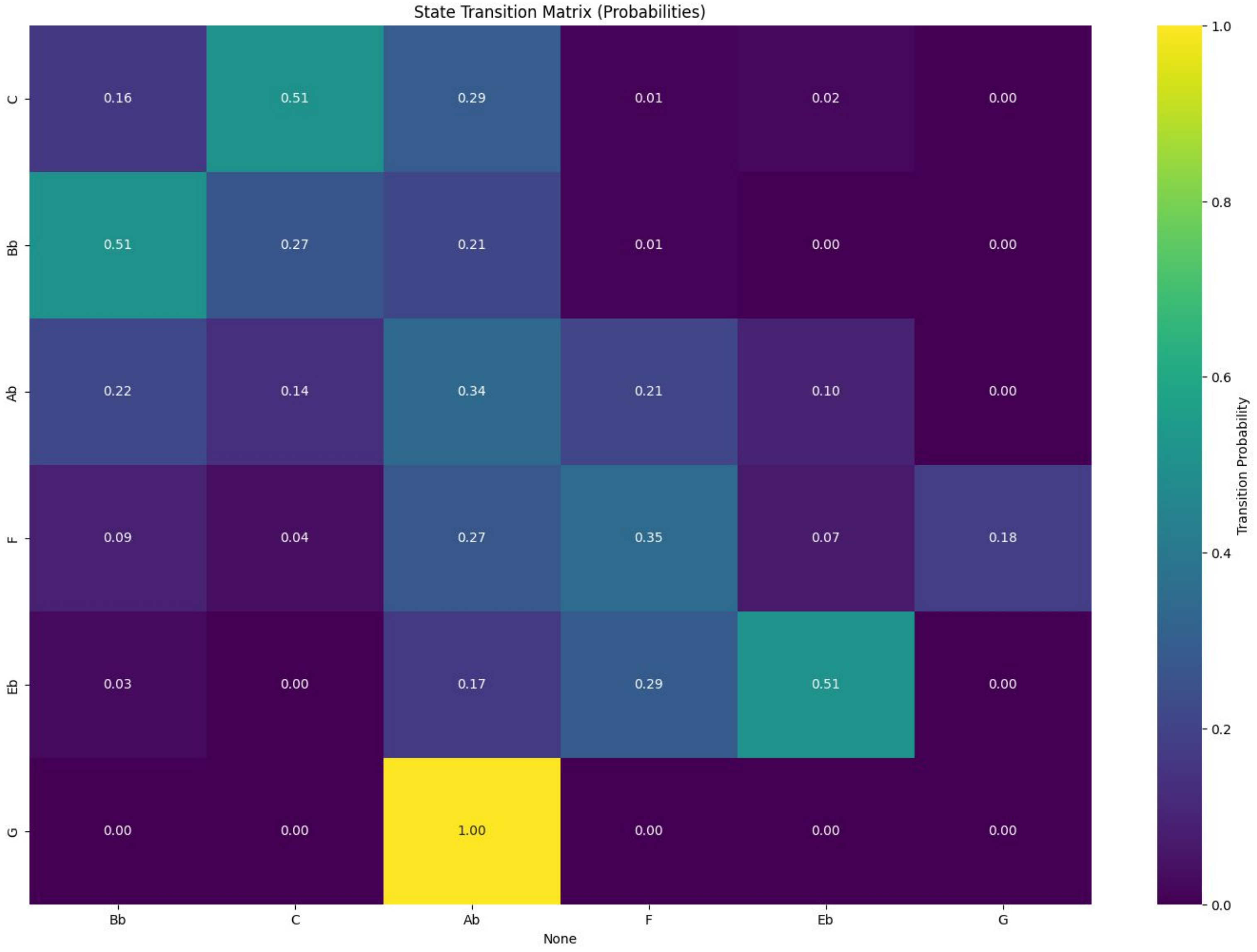
Observations

- State Transition Diagram for First-Order Markov Chain, for the right_hand_notes.



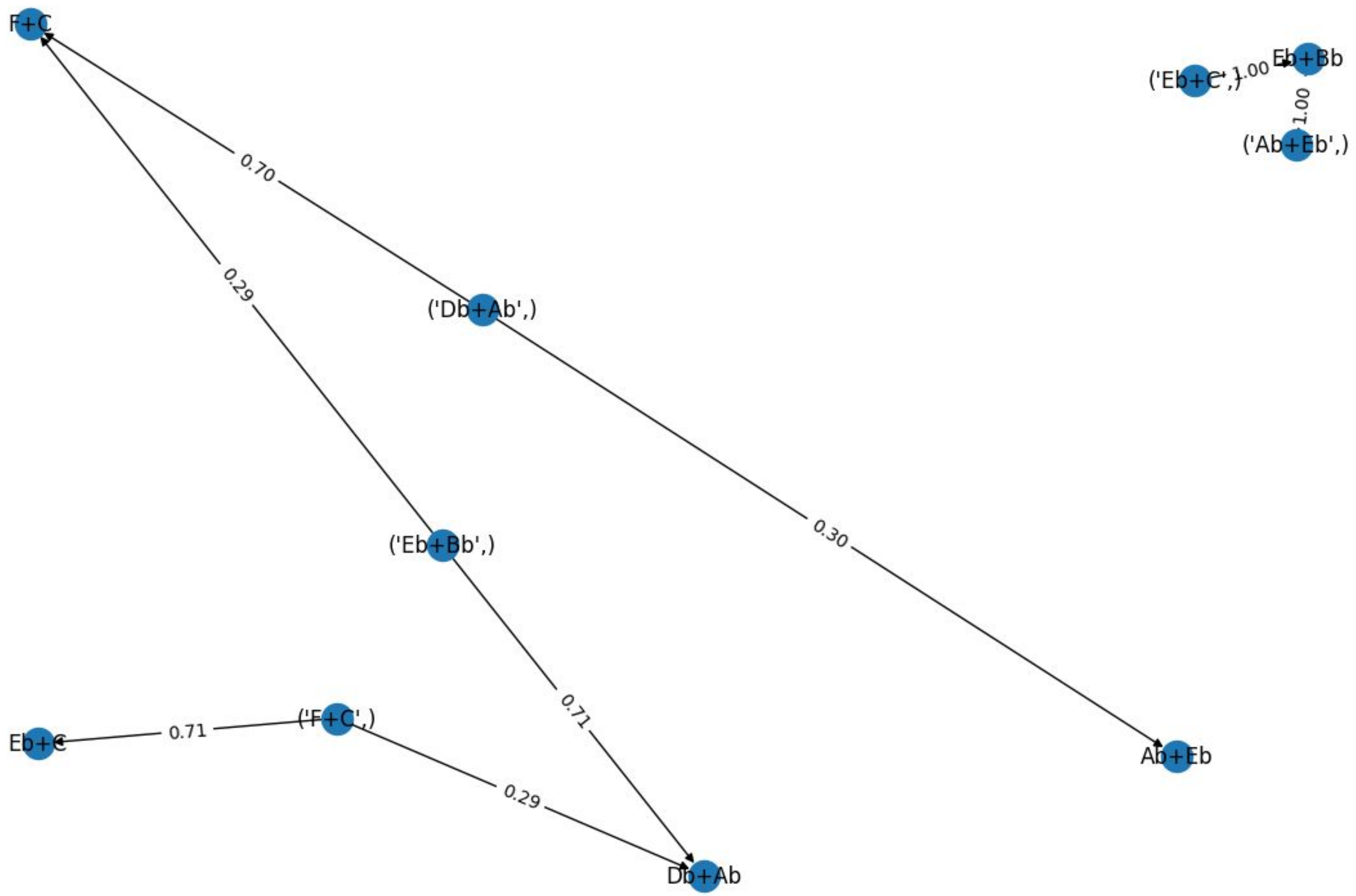
Observations

- State Transition Matrix for First-Order Markov Chain, for the right_hand_notes.



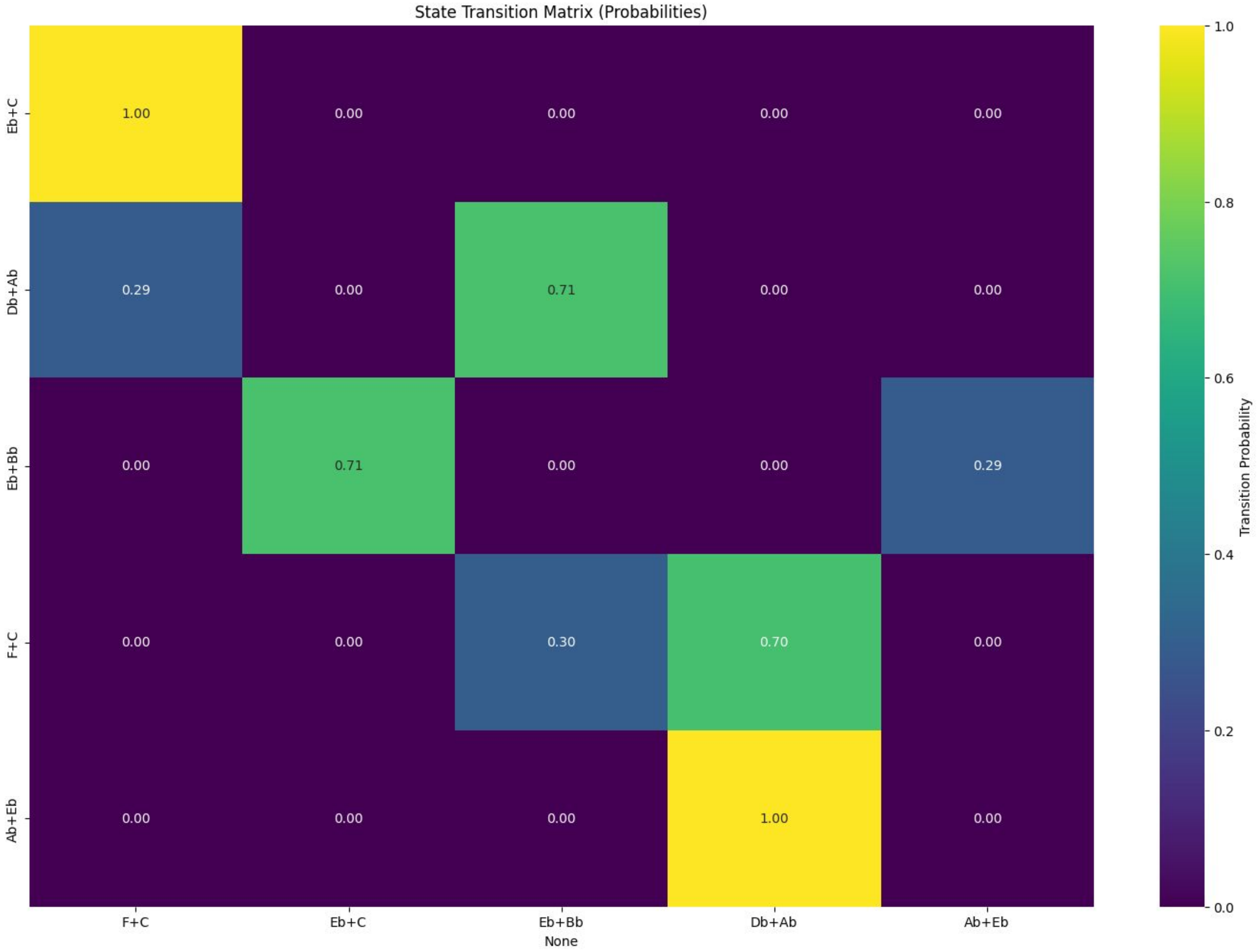
Observations

- State Transition Diagram for First-Order Markov Chain, for the left_hand_notes.



Observations

- State Transition Matrix for First-Order Markov Chain, for the left_hand_notes.



Results

- FOMC Audio File is attached below, which contains the merged audio data, from the left_hand_notes.mid file & right_hand_notes.mid file.

- A few notes don’t seem in sync, due to FOMC’s limited ability to capture longer-term dependencies.

Results

- HOMC Audio File is attached below, which contains the merged audio data, from the left_hand_notes.mid file & right_hand_notes.mid file.

- Sounds much better, & perhaps more harmonic than the output of HOMC's code - due to its ability to capture long-term dependencies.

Future Scope

- **Neural Network Integration:** Explore the integration of neural network models, such as recurrent neural networks (RNNs) or long short-term memory networks (LSTMs), for more advanced and expressive music generation.

- **User Interface Development:** Create a user-friendly interface that allows users to interact with and customize the generated music, providing options for real-time adjustments and feedback.

- **Genre-specific Models:** Develop specialized models for different music genres to better capture the unique characteristics and structures present in genres like classical, jazz, rock, etc.

- **Dynamic Tempo and Timing:** Implement dynamic tempo and timing adjustments within the generated sequences to add more realism and variety to the musical output.

- **Incorporate Lyrics Generation:** Extend the project to generate not only musical sequences but also corresponding lyrics, creating a more comprehensive music composition system.

Thank You!

References

- https://towardsdatascience.com/markov-chain-for-music-generation-932ea8a88305
- https://readthedocs.org/projects/midiutil/downloads/pdf/latest/
- https://sites.math.washington.edu/~conroy/m381-general/markovMusic/markovMusic.htm#:~:text=Here%20is%20an%20example%20of,twelve%20pitches
- https://www.piano-keyboard-guide.com/how-to-play-hello-by-adele-easy-piano-tutorial-for-beginners/