

The Pendular Graph: Visualising Hierarchical Repetitive Structure in Point-set Representations of the POP909 Music Dataset

Chenyu Gao¹ and Tom Collins^{1,2}

¹*University of York*

²*MAIA, Inc.*

chenyugao.cs@gmail.com

Abstract

“Structure” in music can mean many things: repetition, tonality, the existence of and focus on different “musical dimensions”, such as rhythm, timbre, etc. Here, we are concerned with repetitive structures in music, such as sections that repeat within a song (verses, choruses, etc.). We are also concerned mainly with hierarchical repetition (e.g., within a verse, there may be a phrase or riff that recurs multiple times). Existing annotated music datasets tend to be either small in terms of items in the corpus, but with detailed annotations, or larger as a corpus, but with linear annotations only. In this paper, we 1) develop a method for taking a linear annotation as input, and converting it to a hierarchical annotation as output, where such hierarchies exist in the input, and 2) introduce a web-based interface¹ where hierarchical annotations of 909 songs can be explored and played back, in synchrony with a visual representation of note content.

1 Hierarchicalisation of the POP909 Annotations

Repetitions in music tend to be hierarchical in nature [3], such as a phrase being repeated within a verse section that itself repeats. Dai et al. [2] propose single, phrase-level (non-hierarchical) labels for the POP909 dataset. For example, the song “123 Pinocchio” by Hey Girl is annotated “i4A4A4B9b4A4B9b4B9X5o2”, where “A4” means phrase “A” consists of 4 bars. Each time “A4” appears in the sequence, there is an (in)exact occurrence of the phrase “A” in the song.

Inspired by Deutsch and Feroe [3], we propose an algorithm to process the phrase-level labels into a hierarchical annotation. Repetitive substrings with the longest length are labelled as a new pattern by our algorithm. For example, the substring “ABb” in “123 Pinocchio” is gathered as a group and occurs twice, while substrings that never repeat (e.g., “i”, “X” and “o”) or appear only in a larger pattern will not be retained in the hierarchical annotation. For instance, “b” in “123 Pinocchio” does not feature in the hierarchical annotation because “b” never occurs independently of “ABb”.

2 Interface

Our interactive interface visualises the repetitive patterns of songs hierarchically, as a graph consisting of vertices and edges. The web interface is developed by using JavaScript packages: the client side is built mainly using p5.js; the server side is developed with Fastify and the Node Fetch APIs. The MAIA Util package [1] is used when pre-processing and processing music data, and Tone.js [4] enables the songs to be played in the browser.

¹<https://pendular-graph.glitch.me/>

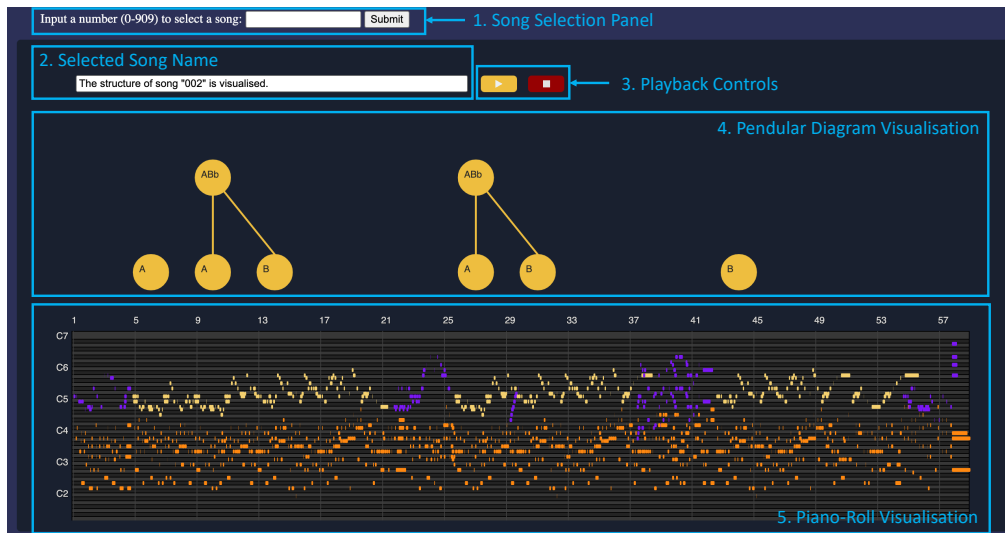


Figure 1: Our pendular graph interface

A screenshot of our interface is shown in Figure 1, indicating five elements:

1. **Song selection panel** at the top of the page. A user inputs a number between 1 and 909 to select a song to visualise.
2. **Selected song name** panel tells the user which song is selected and visualised.
3. **Playback controls** can be clicked to start/pause and stop a song.
4. **Pendular graph visualisation** is the main component of our interface, so-called because the vertices/nodes resemble pendulums. The code behind this component parses the hierarchical structure labels annotated by our algorithm (Section 1), and draws them as a pendular graph, where each vertex represents a repetitive pattern. Horizontal location is determined by appearance in the piece; patterns containing more points are visualised vertically higher up, while patterns containing fewer points are lower down. An edge will connect two vertices if one pattern's point-set representation is a subset of that of another. Users can click nodes in the graph to listen to the corresponding part of the music.
5. **Piano-roll visualisation** at the bottom of this page, in which the melody is coloured in yellow, the secondary melody/lead instruments is coloured in purple, and the accompaniment is coloured in orange.

References

- [1] Collins, T., Coulon, C.: MAIA Util: an npm package for bridging web audio with music-theoretic concepts. In: Proceedings of the Web Audio Conference. pp. 47–52 (2019)
- [2] Dai, S., Zhang, H., Dannenberg, R.B.: Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music. In: Proceedings of the Joint Conference on AI Music Creativity (2020)
- [3] Deutsch, D., Feroe, J.: The internal representation of pitch sequences in tonal music. *Psychological Review* **88**(6), 503–522 (1981)
- [4] Mann, Y.: Interactive music with Tone.js. In: Proceedings of the Web Audio Conference (2015)