From jSymbolic 2 to 3: More Musical Features

Cory McKay[1]

[1] Marianopolis College and the CIRMMT
cory.mckay@mail.mcgill.ca

Abstract. This demo will provide participants with the opportunity to experiment with the jSymbolic software, which extracts a broad range of statistical features from digital scores in formats such as MIDI. Participants will be able to use and compare both the current 2.2 release version and the pre-release jSymbolic 3. Past research using jSymbolic in diverse areas of computational musicology and music information retrieval (MIR) will be discussed, involving machine learning and statistical analysis. Participants will be encouraged to engage in dialogue on how jSymbolic might be incorporated into their own research, and on ideas for new features that could be added to the jSymbolic catalogue that would benefit their work. Research focusing on symbolic data as well as multimodal investigations will both be emphasized. jSymbolic is entirely open source.

Keywords: Symbolic music; Features; Computational musicology; MIR.

1 jSymbolic and its Motivation

Three essential advantages of applying computational methodologies to musicology are: 1) the ability to directly consider and compare corpora consisting of hundreds or thousands of pieces of music, more than would be feasible using manual techniques; 2) movement towards "objectively" analyzing music in ways that filter out at least some of the biases we are all subject to when manually analyzing music; and 3) the creation of opportunities to explore music in novel ways that can reveal musically meaningful insights in areas we might not have thought to consider using traditional techniques.

This demo presents the jSymbolic software, whose primary purpose is to help music researchers and scholars benefit from these advantages. It automatically extracts features (characteristic pieces of information) from digital scores encoded in symbolic file formats such as MIDI or MEI. Each feature describes a clearly defined characteristic of music that can be consistently extracted and compared, as a single number or as a vector of associated values. For example, a "range" feature could be defined as the number of semitones separating the lowest and highest pitches in the music being analyzed. Features extracted from multiple pieces can be aggregated into categories of interest (e.g., composers, genres, regions, etc.), which can themselves be compared collectively. Although jSymbolic allows features to be extracted over smaller windows of


This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

set length, typically features are extracted from pieces or major sections (e.g., mass movements) in their entirety, so as to more broadly characterize them.

Once extracted by jSymbolic, features can be used in a variety of ways, such as for training classifiers using supervised machine learning, or for exploratory clustering via unsupervised learning. For example, a classifier could be trained on jSymbolic features to model the compositional styles of various Renaissance composers, and this classifier could then be used to help identify probable authorship of unattributed or controversially attributed works. This approach, often carried out using relatively simple algorithms like support vector machines, is particularly appropriate in situations where there are relatively few extant training exemplars, as is often the case with early music, since deep learning alternatives (which tend to in effect learn their own features from raw musical data) can have too many parameters to perform well in such circumstances.

Another important advantage of engineered features like jSymbolic's is that they are largely musically interpretable. This enables the application of statistical techniques, such as information gain analysis or feature selection (e.g., with genetic algorithms), to determine which features most meaningfully separate different pieces or groups of music; this can be more musicologically important than actual classifications themselves. So, to continue the Renaissance sample use case, one might use jSymbolic features to gain insight into what specifically statistically differentiates the styles of composers like Josquin, de la Rue and Ockeghem, in *musically meaningful* terms.

Feature values can also be examined directly by domain experts if desired. They can be saved as CSV files (or in specialized machine learning-oriented formats), which can then be imported into spreadsheet or data analysis software for study or visualization.

Although much of the jSymbolic research to date has focused on either Western early music or popular music, jSymbolic can also be applied to many other musics, or combined with other types of data (e.g., audio) in multimodal research. All the jSymbolic features can be extracted from any kind of music that can be encoded as MIDI, which permits complex rhythmic structures and pitches outside the Western chromatic scale.

2 Previous Work

jSymbolic was first released in 2006 [1], was included in the multimodal jMIR music research suite in 2010 [2] and the current release version (2.2) was published in in 2018 [3]. jSymbolic has been used as a core part of many published research projects, in areas including popular music genres [2], investigating the origins of Renaissance genres [4], regional style [5], compositional style [6] and multimodal analysis [7].

Of course, jSymbolic is not the only research platform available for computational musicology or symbolic MIR. However, its focus specifically on standardized high-level summary features means it has quite different use cases from other excellent software like Humdrum [8], Music21 [9], pretty_midi [10], CRIM [11] and MIDI Toolbox [12], which emphasize tasks like retrieving specified instances of local events, or visualizing or manipulating user-specified elements. The alternatives are less suited to the macro statistical analysis that jSymbolic specializes in. Notably, Music21 can extract features, but mostly just a subset of ported-over jSymbolic 1.2 features.

3 Details About jSymbolic 2.2

The current release version (2.2) of jSymbolic extracts 246 unique features, which total to 1497 feature values when feature vectors are expanded. This feature catalogue is designed to be diverse, so that it is applicable to as many characteristics of as many types of music as possible. The jSymbolic features can be divided into these groups:

- **Pitch Statistics:** How common are various pitches and pitch classes relative to one another? How are they distributed and how much do they vary?
- **Melodic Intervals:** What melodic intervals are present? How much melodic variation is there? What can be observed from melodic contour measurements?
- **Chords and Vertical Intervals:** What vertical intervals are present? What types of chords do they represent? What kinds of harmonic movement are present?
- **Rhythm:** Information associated with note attacks, durations and rests, measured in ways that are both dependent and independent of tempo. Information on meter and rhythmic variability, including rubato.
- **Instrumentation:** Which instruments are present, and which are emphasized relative to others? Both pitched and non-pitched instruments are considered.
- **Texture:** How many independent voices are there and how do they interact (e.g., parallel vs. contrary motion)? What is the relative importance of voices?
- **Dynamics:** How loud are notes and what kinds of variations in dynamics occur?

jSymbolic has a graphical user interface (Fig. 1) as well as a command line interface and a Java API for those wishing to use jSymbolic via scripting or to integrate it into their own software. There is also a detailed manual, which includes individual feature explanations, and a tutorial with worked examples.

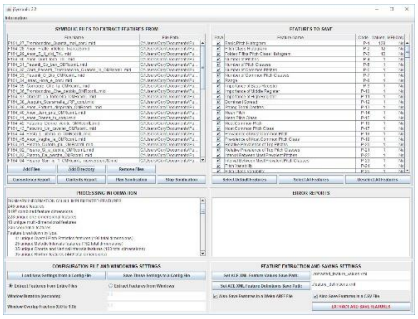

Fig. 1. The jSymbolic 2.2 graphical user interface.

In addition to being a ready-to-use application, jSymbolic is also intended as a platform for building new features, including ones of increasing sophistication built upon existing features. New features are added as plug-ins, and jSymbolic automatically schedules extraction to resolve feature dependencies. Many of the features added since jSymbolic was first released have resulted from consultation and collaboration with musicologists, theorists and MIR researchers, and it is hoped that this will continue.

4 Towards jSymbolic 3

The upcoming jSymbolic 3 is currently undergoing final testing and improvement before release. In addition to many miscellaneous usability improvements, it has a substantially expanded feature catalogue of 533 unique features and 2040 feature values in total. Of particular interest, these include a new *n-gram* group of features, which extract information from aggregated sequences of musical events, thus giving jSymbolic more insight into local context. Features are extracted from three types of n-grams: melodic interval, vertical interval and rhythmic.

During this demo, participants will be given the first public hands-on look at jSymbolic 3, and will be able to compare it with jSymbolic 2.2.

jSymbolic, its code and documentation are all available at http://jmir.sourceforge.net (version 2.2) and at https://github.com/DDMAL/jSymbolic2/ (version 3 development).

References

1. McKay, C., Fujinaga, I.: jSymbolic: A feature extractor for MIDI files. In: Proceedings of the International Computer Music Conference, pp. 302–305. Miami (2006).
2. McKay, C.: Automatic music classification with jMIR. Ph.D. Dissertation. McGill University, Canada (2010).
3. McKay, C., Cumming, J., Fujinaga, I.: jSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research. In: Proceedings of the International Society for Music Information Retrieval Conference, pp. 348–354. Paris (2018).
4. Cumming, J., McKay, C.: Using corpus studies to find the origins of the madrigal. In: Proceedings of the Future Directions of Music Cognition International Conference, pp. 38–42. Online (2021).
5. Cuenca, M. E., McKay, C.: Exploring musical style in the anonymous and doubtfully attributed mass movements of the Coimbra manuscripts: A statistical and machine learning approach. Journal of New Music Research 50(3), 199–219 (2021).
6. Rodríguez-García, E., McKay, C.: Composer attribution of Renaissance motets: A case study using statistical features and machine learning. In: The Anatomy of Iberian Polyphony around 1500, eds. Rodríguez-García, E., d'Alvarenga, J. P, 401–438. Edition Reichenberger, Kassel (2021).
7. Vatolkin, I., McKay, C.: Multi-objective investigation of six feature source types for multimodal music classification. Transactions of the International Society for Music Information Retrieval 5(1), 1–19 (2022).
8. Huron, D.: Music information processing using the Humdrum toolkit: Concepts, examples, and lessons. Computer Music Journal 26(2), 11–26 (2002).
9. Cuthbert, M. S., Ariza, C., Friedland, L.: Feature extraction and machine learning on symbolic music using the music21 toolkit. In: Proceedings of the International Society for Music Information Retrieval Conference, pp. 387–392. Miami (2011).
10. Raffel, C., Ellis, D. P. W.: Intuitive analysis, creation and manipulation of MIDI data with pretty_midi. In: International Society for Music Information Retrieval Conference Late Breaking and Demo Papers. Taipei (2014).
11. CRIM Intervals Search Tools, https://crimintervals.streamlit.app/, last accessed 2023/07/31.
12. Eerola, T., Toiviainen, P.: MIR in Matlab: The MIDI Toolbox. In: Proceedings of the International Conference on Music Information Retrieval, pp. 22–27. Barcelona (2004).