Tool-based Identification of Melodic Patterns in MusicXML Documents

Manuel Burghardt (manuel.burghardt@ur.de),

Media Informatics Group, University of Regensburg, Germany

Lukas Lamm (lukas.lamm@stud.uni-regensburg.de),

Media Informatics Group, University of Regensburg, Germany

David Lechler (david.lechler@stud.uni-regensburg.de),

Media Informatics Group, University of Regensburg, Germany

Matthias Schneider (Matthias.Schneider@stud.uni-regensburg.de),

Media Informatics Group, University of Regensburg, Germany

Tobias Semmelmann (Tobias.Semmelmann@stud.uni-regensburg.de),

Media Informatics Group, University of Regensburg, Germany

Introduction: Digital musicology

Computer-based methods in musicology have been around at least since the 1980s[1]. Besides the creation of digital editions (cf. Kepper et al., 2014; Veit, 2015), scholars in this area of study have also been interested in quantitative approaches for musicological analyses (cf. Müllensiefen and Frieler, 2004; Vigilanti, 2007).-Such quantitative analyses rely on music information retrieval (MIR) systems, which can be used to search collections of songs according to different musicological parameters. There are many examples for existing MIR systems, all with specific strengths and weaknesses. Among the main downsides of such systems are:

- **Usability problems**, i.e. tools are cumbersome to use, as they oftentimes only provide a command-line interface and also require some basic programming skills to utilize them; example: Humdrum[2]

- **Restricted scope of querying**, i.e. tools can only be used to search for musical incipits; examples: RISM[3], HymnQuest[4]

---

[1] The popular series „Computing in Musicology" started around 1985. For an overview of all volumes of the series cf. http://www.ccarh.org/publications/books/cm/; Note: All URLs mentioned in this text were last checked on March 3, 2016.
[2] http://www.humdrum.org/
[3] https://opac.rism.info/
[4] http://hymnquest.com/

- **Restricted song collection**, i.e. tools can only be used for specific collections of music files; various examples of MIR tools for specific collections are described in Typke et al. (2005)

A particularly promising MIR tool can be found in Peachnote[5] (Viro, 2011), which uses optical music recognition (OMR) software to index more than one million sheets from the Petrucci Music Library[6], aiming to provide a search interface for musicology which can be seen as an analog of the Google Books Ngram Viewer[7]. Despite many existing software solutions, we believe that accurate OMR is still a major challenge in digital musicology. At the same time, there are numerous databases[8] at hand, that provide machine-readable music documents, fully annotated with MusicXML (Good, 2001) markup.

On this account, we designed MusicXML Analyzer, a generic MIR system that is trying to overcome the weaknesses of existing MIR tools, and that allows for the analysis of arbitrary documents encoded in MusicXML format.

MusicXML Analyzer: Basic functionality and implementation details

MusicXML Analyzer can be used to analyze songs in a quantitative manner, and to search for specific melodic patterns in a collection of songs. The results of the analyses are rendered as virtual scores and can be viewed in any recent web browser. In addition, the queries and the results can be played as a synthesized audio file; all analyses can also be exported as PDF or CSV files.

The tool comprises three main components: (1) the upload function, (2) the analysis function, and (3) the search function. After one or more files in MusicXML format have been uploaded via an intuitive drag-and-drop dialog, the analysis component parses the data and calculates basic frequencies; the results are stored in an SQL database and can be displayed in a dashboard view (cf. Fig. 1).

---

[5] http://www.peachnote.com/
[6] http://imslp.org/
[7] https://books.google.com/ngrams
[8] http://www.musicxml.com/music-in-musicxml/

Figure 1: Snippet from the dashboard view, showing basic frequencies for a corpus of MusicXML documents.

The dashboard displays the following information, either for an individual song, or for a corpus of multiple songs:

- Overall statistics for single notes, rests and measures

- Types of instruments used in the song (if described in the MusicXML data)

- Frequency distribution for single notes, intervals, keys, note durations and meters

Via a dedicated search function, a corpus of MusicXML documents can be queried for melodic patterns on different levels of information:

- Search for a sound sequence; example: c', c', g', g'

- Search for a rhythmic pattern; example: eighth note, eighth note, quarter note, quarter note

- Search for melodic patterns, i.e. a combination of sound sequence and rhythm; example: c' / eighth note, c' / eighth note, g' / quarter note, g' / quarter note

Search queries can be entered via a virtual staff that was realized with the VexFlow library[9] (cf. Fig. 2). Once a search pattern has been entered, it can also be played as a synthesized Midi sequence, which was realized with the Midi.js library[10].
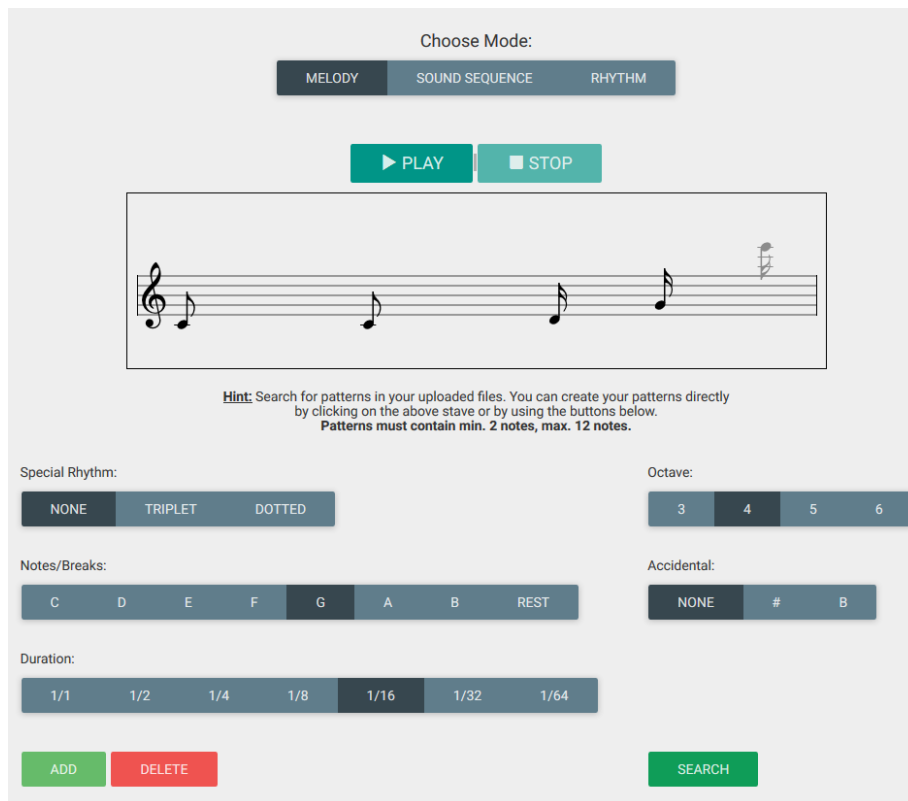

Figure 2: Interface for entering queries to identify tonal, rhythmic, or melodic patterns in a corpus of MusicXML documents.

After a query has been submitted, all results – i.e. the songs that contain the search pattern – are displayed in a list view. The list contains the name of the song and also the number of total occurrences of the search pattern in that song. By clicking on one of the song items in the list, a virtual score is rendered for the whole song; the search pattern is highlighted whenever it occurs in that virtual score (cf. Fig. 3). The whole song can be played directly in the web browser, or downloaded for further analyses as a PDF (realized with the jsPDF library[11]).

---

[9] http://www.vexflow.com/
[10] http://mudcu.be/midi-js
[11] https://parall.ax/products/jspdf

Figure 3: Virtual score rendering of a document from the results list; the search pattern is highlighted in red color.

MusicXML Analyzer was implemented by means of standard web technologies (HTML, CSS, JavaScript, PHP), in particular by utilizing the following libraries and frameworks: Laravel[12], jQuery[13], D3.js[14], Bootstrap[15], Typed.js[16], Dropzone.js[17].

A short demo video that showcases the main functionality of the tool is available at

- https://dl.dropboxusercontent.com/u/4194636/MusicXML-Analyzer.mp4

A fully functional online demo[18] of MusicXML Analyzer is available at

- http://music-xml-analyzer.herokuapp.com/

MusicXML Analyzer can also be downloaded and modified (according to the MIT open source license) from GitHub:

- https://github.com/freakimkaefig/Music-XML-Analyzer

---

[12] http://laravel.com/
[13] https://jquery.com/
[14] http://d3js.org/
[15] http://getbootstrap.com/
[16] http://www.mattboldt.com/demos/typed-js/
[17] http://www.dropzonejs.com/
[18] Due to some technical limitations of our server environment, the initial access to the online demo may take a few seconds to wake up the server from *idle mode*.

Future directions

In its current implementation, MusicXML Analyzer performs an exact match search, i.e. only documents which have the exact same value in their MusicXML markup will be found by the search function. We are planning to implement a more sophisticated melodic similarity algorithm (cf. Grachten et al., 2002; Miura and Shioya, 2003) that allows for the configuration of different similarity thresholds.

At the same time, we are adapting MusicXML Analyzer for a recent project on a large corpus of German folksongs. Besides monophonic melodies, this collection of folksongs also contains machine-readable metadata (region, date, etc.) and lyrics. Accordingly, we are trying to enhance the search features of MusicXML Analyzer in a way it can not only search songs for melodic patterns, but also for metadata parameters and keywords from the lyrics. Such an enhanced MIR system could be used to analyze the following research questions:

- Are there characteristic melodic and linguistic patterns for German folksongs, from a diachronic perspective as well as from a regional perspective?

- Are there melodic-linguistic collocations, i.e. do certain melodic patterns co-occur with certain keywords or phrases?

References

Good, M. (2001). MusicXML for Notation and Analysis. In Hewlett, W. B. and Selfridge-Field, E. (eds.), *The Virtual Score: Representation, Retrieval, Restoration*. Cambridge (MA) and London (UK): MIT Press, pp. 113–24.

Grachten, M. A., Josep, L. and Mántaras R. L. (2002). A comparison of different approaches to melodic similarity. *Proceedings of the 2nd International Conference in Music and Artificial Intelligence (ICMAI) 2002*.

Kepper, J., Schreiter, S. and Veit, J. (2014). ‚Freischütz' analog oder digital – Editionsformen im Spannungsfeld von Wissenschaft und Praxis. *Editio,* 28: 127–50.

Miura, T., & Shioya, I. (2003). Similarity among melodies for music information retrieval. *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM) 2003*.

Müllensiefen, D. and Frieler, K. (2004). Optimizing Measures Of Melodic Similarity For The Exploration Of A Large Folk Song Database. *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR) 2004*, pp. 274–80.

Typke, R., Wiering, F. and Veltkamp, R. C. (2005). A survey of music information retrieval systems. *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR) 2005*, pp. 153–60.

Veit, J. (2015). Music notation beyond paper. On developing digital humanities tools for music editing. *Forschungsforum Paderborn,* 18: 40-48.

Viglianti, R. (2007). MusicXML: An XML Based Approach to Musicological Analysis. Digital Humanities 2007: Conference Abstracts, pp. 235–37.

Viro, V. (2011). Peachnote: Music Score Search and Analysis Platform. *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR) 2011*, pp. 359-62.