ms3: A parser for MuseScore files, serving as data factory for annotated music corpora

Johannes Hentschel and Martin Rohrmeier

École Polytechnique Fédérale de Lausanne, Switzerland

Corresponding author

DOI: 10.21105/joss.05195

Software
- Review
- Repository
- Archive

Statement of need

Music scores represent relations between sounding events by graphical means. Music notation software therefore is often concerned with problems of layout and easy-to-read rendering of symbols in line with the multitude of notational conventions (Read, 1979; Ross, 2001); rather than with the explicit encoding of the musical relations themselves. For example, the Free and Open Source Software MuseScore provides a full-featured yet intuitive interface for engraving music, but its native XML format does not explicitly encode the temporal positions of events such as notes and rests. Hence the need for a parser that extracts the implicit information and stores it in an interoperable format.

Despite being one of the most widespread score encoding formats, current score parsers (e.g., Cancino-Chacón et al., 2022; Cuthbert, 2010; Pugin et al., 2014), do not handle it without first performing a lossy conversion to the musicXML format. The Python library ms3 fills this gap. It loads the XML tree of a MuseScore file into working memory, computes the temporal positions of all encoded elements, and transforms those requested by the user into DataFrames (Petersohn, 2021). The DataFrames can be used by other Python programs and scripts, or written to Tab-Separated Values (TSV) to enable processing with other software and facilitate version control. The most typical aspects that users extract from a score are tables containing notes, measures (bars), metadata, and text labels, in particular those representing analytical annotations. Moreover, ms3 allows the user to transform scores by removing analytical labels after their extraction or by (re-)inserting annotations from TSV files (whether previously extracted or generated from scratch). This functionality turns MuseScore into a convenient score annotation tool enabling users to graphically insert into a score arbitrary textual labels,

---

1For example, musicXML’s implicit encoding of temporal positions is limited to those where a note or rest event occurs. When converting MuseScore XML to musicXML, all score elements occurring between two such events are misplaced.

2Version control is facilitated by the TSV files because, unlike the original XML source, they present score information with timestamps.
to then have ms3 extract them with their temporal positions for further analysis. It comes with a command line interface that makes its data extraction, transformation, and validation functionalities accessible for productive everyday workflows.

ms3 has already been used for creating several datasets, namely version 2 of the Annotated Beethoven Corpus (Neuwirth et al., 2018), the Annotated Mozart Sonatas (Hentschel, Neuwirth, et al., 2021), and an annotated corpus of 19th century piano music (Hentschel, Rammos, et al., in press). It has been successful in formatting training and validation data for a chord inference algorithm and for inserting its analytical outputs into the respective scores (Mcleod & Rohrmeier, 2021). Moreover, the library is at the heart of a semi-automated annotation workflow running on GitHub (Hentschel, Moss, et al., 2021) and a dependency on the music corpus analysis library DiMCAT (Hentschel, McLeod, et al., in press).

Acknowledgements

Development of this software tool was supported by the Swiss National Science Foundation within the project “Distant Listening – The Development of Harmony over Three Centuries (1700–2000)” (Grant no. 182811). This project is being conducted at the Latour Chair in Digital and Cognitive Musicology, generously funded by Mr Claude Latour.

References


