# Representing and Linking Music Performance Data with Score Information

Johanna Devaney
School of Music, The Ohio State University
1866 North College Road
Columbus, OH, USA
devaney.12@osu.edu

Hubert Léveillé Gauvin
School of Music, The Ohio State University
1866 North College Road
Columbus, OH, USA
leveillegauvin.1@osu.edu

## ABSTRACT

This paper argues for the need to develop a representation for music performance data that is linked with corresponding score information at the note, beat, and measure levels. Building on the results of a survey of music scholars about their music performance data encoding needs, we propose best-practices for encoding perceptually relevant descriptors of the timing, pitch, loudness, and timbral aspects of performance. We are specifically interested in using descriptors that are sufficiently generalized that multiple performances of the same piece can be directly compared with one another. This paper also proposes a specific representation for encoding performance data and presents prototypes of this representation in both Humdrum and Music Encoding Initiative (MEI) formats.

## CCS Concepts

**•Information systems → Data encoding and canonicalization; •Applied computing → Sound and music computing;**

## Keywords

Music data representations; Musical performance; Digital musicology

## 1. INTRODUCTION

Empirical methods for analyzing musical scores have already had an impact on music scholarship, and the examination of recorded performance is a growing area of interest that shows similar potential [3]. Recent work in the music information retrieval and music cognition communities have led to an increase in both techniques for extracting data from musical performances [1, 9, 12] and corpora of such data [2, 4, 18]. These data may be extracted from specialized equipment at the time of performance, such as a Disklavier or Bosendorfer SE piano, or may be extracted from existing audio recordings using manual or automatic methods (or some combination of the two). For these performance data to be useful for more traditional music scholars, who may not be trained in computational methods, they should ideally be stored in a format where they are unified with the score data to which they refer. Thus the encoding format needs to be linked to the musical notes and sufficiently generalized so that researchers working on different types of research questions can be accommodated. In the context of digital music libraries, having performance data directly linked to score data is important when building large-scale corpora where multiple performances of multiple pieces are properly encoded and indexed.

This paper describes best practices for encoding perceptually relevant descriptors related to the timing, pitch, loudness, and timbral aspects of performance at the note, beat, and measure levels. Building on the results of a survey of music scholars about their music performance data encoding needs, we propose best-practices for encoding perceptually relevant descriptors of the timing, pitch, loudness, and timbral aspects of performance. We are specifically interested in using descriptors that are sufficiently generalized that multiple performances of the same piece can be directly compared with one another. This paper also proposes a specific representation for encoding note-level performance data and presents prototypes of this representation through new spine definitions in the Humdrum syntax, and through an extension to the existing 'performance' module in the Music Encoding Initiative (MEI) format. It also describes our plans for future work related to encoding continuous data descriptors.

## 2. BACKGROUND

Currently, the available corpora of performance data are released in formats that are divorced from the musical material. Typically, they take the form of a list of values which correspond to the music at either the note or beat level, depending on the type of performance parameters being extracted and the type of musical performance parameter being considered, e.g., intonation data is generally provided as note-level descriptors of pitch whereas timing information may be provided at the beat level. Examples at the note level for monophonic instruments include the Ensemble Expressive Performance Dataset[1] from researchers in the Music Technology Group in the Universitat Pompeau Fabra [18], where onset times, offset times, and note pitch information are provided in a line-by-line text format for each

[1] http://mtg.upf.edu/download/datasets/eep-dataset



*DLfM '16, August 12 2016, New York, USA*



DOI: http://dx.doi.org/10.1145/2970044.2970052

instrument in a string quartet, and the singing dataset[2] by researchers in the Centre for Digital Music at Queen Mary, University of London [4], which provides lists of note-level timing and pitch data for monophonic vocal performances in CSV files. At the beat level, the Mazurka dataset[3], released by the Charm project [2], provides beat-by-beat timing and dynamic information from piano performances of Chopin Marzurkas as a set of Excel spreadsheets. In all of these datasets, as summarized in 1, there is no direct encoding of performance data with detailed score information, requiring people using the datasets to link different files together in order to explore any relationships that may exist between the performance data and the musical material.

The number and size of available corpora of performance data has been facilitated by a marked increase over the past ten years in software tools that allow music scholars to measure various aspects of musical performance. One such example is the Sonic Visualiser software developed at Queen Mary, University of London [1]. Sonic Visualiser is an open-source software with a graphical user interface (GUI) that allows the user to view different annotation layers simultaneously. Several plugins have been developed by different researchers which grandly facilitates tasks such as beat or onset detection. More automated methods have been developed by utilizing the information available in MIDI scores of the music performances either strictly in the MIDI domain, such as the automated score-performance matcher proposed in [12], or for audio recordings, as is done in the the Automatic Music Performance Analysis and Comparison Toolkit (AMPACT) [9]. One of the challenges with using such software, however, is the lack of available storage representations where the performance data is explicitly linked to the score data.

We believe that encoding the data separately from the musical material to which they refer both reduces the human-readability of the encoding and presents unnecessary technical challenges for users who want to work with the data as they need to create linkages between the musical material and the performance data on their own. An additional challenge is to create a linked encoding format that is sufficiently generalized and extendable, so that researchers working on different types of research questions can be accommodated.

While there currently exists a handful of different encoding formats used by music researchers, none of them were designed to encode both score and performance data with the music scholar in mind. It is possible to annotate performance data using the MIDI protocol, however, the range of available performance-related MIDI descriptors, particularly for tuning, loudness, and timbre is limited. Also, MIDI is generative rather that descriptive, meaning that it is meant to produce a sound rather that annotate it, which makes it less than ideal for music research because it aims to provide too much low-level information about the sound that is to be produced, rather than more generalized descriptors of the perceptual attributes of the recording it is describing. Furthermore, the MIDI encoding representation is difficult to easily interpret without extensive experience. For example, pitches are identified using a value between 0 and 127, rather than through scientific pitch notation (e.g. C4). Similarly, the MIDI encoding representation for note duration is calculated as the time between the onset and offset of a note, meaning that a quarter note at 60 BPM and a quarter note at 120 BPM would be encoded using different values. While this makes sense from a technical point of view, it is divergent from Western music notation and thus counter intuitive for most music scholars. Another potential candidate is MusicXML. Like MIDI, MusicXML is a commercially-backed endeavor, though its purpose is to share music-related data (usually scores) between different software packages rather than provide a standard format for music generation. Like the MIDI format, MusicXML is machine-readable rather human-readable. The MIDI-compatible component of MusicXML allows for some MIDI descriptors, but not those related to performance. Hirata, Noike, and Katayose [15] proposed extensions to MusicXML to incorporate additional MIDI descriptors, including note-level timing and dynamics information, but these have not been incorporated into the MusicXML protocol.

In the academic world, there are two other encoding formats that currently offer ways to encode both score and performance data: Humdrum and MEI. The Humdrum syntax was designed by David Huron in the 1990s [16] as a framework to represent a wide range of musical parameters. Although the Humdrum syntax does not currently offer include performance data, such descriptors can be easily defined and documented. The syntax organizes the data in a tab-separated format, similar to a spreadsheet. Vertical spines, or columns, represent individual descriptors, while horizontal lines usually represent individual musical objects (e.g. note, beat, syllable). This linear organization makes Humdrum easy to read by humans, which makes the learning curve more gentle than other formats, and thus makes it an interesting format for scholars with limited computational experience. We shall see in Section 4, however, that this linearity creates some restrictions in its ability to encode music data simultaneously at the note and beat levels.

MEI is a formalized XML-based schema designed to encode musical representation that was developed by Perry Roland in the early 2000s [20]. In contrast to MusicXML, MEI was designed to allow for structured metadata to be defined and encoded with the musical materials and to encode information about the visual layout of the musical materials within musical scores. It was inspired by the Text Encoding Initiative (TEI) [17], a machine-readable format designed in the 1980s to digitally archive documents and widely used by libraries and individual scholars. TEI was designed as a standard coding syntax for digital humanities to create and publish electronic editions in a shareable format. While the original TEI guidelines used the Standard Generalized Markup Language (SGML), the organization switched to XML in 2003. While the MEI is not affiliated with the TEI, it shares common goals. The MEI format currently includes a performance module that can be used to link a score to a distinct moment in a recording. For example, linking the start of a specific section in the score with the relevant moment in the recording. This module, however, was not originally intended for lower-level descriptors. Our goal is to expand this to allow for note-, beat-, and measure-level performance descriptors to be represented. Unlike the linear Humdrum syntax, MEI is hierarchically structured in terms of defined elements and their attributes. As we will see later, these differences have an impact on the encoding approaches to be preferred.

[2]https://code.soundsoftware.ac.uk/projects/dai2015analysis-resources

[3]http://www.mazurka.org.uk/

| *Dataset* | *Format* | *Encoding Level* | *Score Data* | *Performance Data* |
|---|---|---|---|---|
| EEP | Text | Note | Pitch | Onset, Offset |
| C4DM | CSV | Note | Pitch, Duration | Pitch, Onset, Duration |
| Mazurka | XLS | Beat/Measure | Measure number, Beat number | MIDI velocity |
| Mazurka | Text | Beat | Event number | Onset |

**Table 1: Summary of encoding formats used in [2, 4, 18].**

In Section 4, we describe our proposed extensions to both Humdrum and MEI to facilitate the encoding of note-level performance descriptors and discuss the challenges of encoding both higher-level data, at the beat- and measure-level, as well as continuous data. The design decisions for these extensions were informed in part by a survey of music scholars that we performed, the results of which are summarized in the next section.

## 3. SURVEY OF MUSIC SCHOLARS

We conducted an online survey to ensure that our proposed extensions of Humdrum and MEI are able to encode parameters that music scholars are interested in working with. The goal of the survey was to solicit information from researchers who are interested in analyzing musical performances with a range of technical skill levels. To this end the survey was sent to both the International Society of Music Information Retrieval and the Society for Music Theory mailing lists. For scholars with less technical skill, we want to ensure that that encoding format facilitates the easy sharing of performance analyses with others and the creation of databases of musical performances. For scholars with more technical skill, we want to ensure that the encodings allow for the range and detail of performance parameters their work requires. Overall, we had 79 respondents who began the survey, 54% of whom were currently or had previously conducted research on musical performances using computational tools (with the remainder interested in doing so in the future). 59 respondents completed the survey. These 59 were predominantly based in North America (34%) and Europe (32%), but were also drawn from Asia (7%) and South America (3%), and two thirds of the respondents were between 25 and 44 years of age (see Figure 1 (a)). Nearly three quarters of the respondents highest degree in music was at the graduate level, either at the Master's level (24%) or the Doctoral level (59%) (see Figure 1 (b)). The respondents represent a wide range of music sub-disciplines, with the majority coming from music theory, followed by music technology and musicology. Other represented fields include performance, composition and music education (see Figure 1 (c)). The majority of the respondents were faculty members (63%), although graduate students, postdoctoral researchers, and people working outside of academia were also represented (see Figure 1 (d)).

In the survey, we asked respondents to rank how important to them the following parameters are in their performance research: timing, loudness, pitch, and timbre. The results are shown in Figure 2. Timing is more often considered the most important parameter, followed by pitch, as evidenced by the high number of first and second positions assigned to both of these parameters. Timbre and loudness are more or less tied in third and fourth position, with more people selecting timbre as their most important parameter but with a large number of people selecting loudness as their second most important parameter. This shows that while timing parameters have traditionally dominated the performance analysis literature, it is important to accommodate other types of descriptors, some of which may be better suited to be encoded as continuous data, such as certain pitch-related data, than as discrete note-, beat-, or measure-level descriptors.

Similarly, we asked respondents to rank the different level of description they would like to encode performance data at: Continuous, Note-level, Beat-level, and Measure-level. The results are shown in Figure 3. Continuous and note-level data ranked first and second, respectively, with beat-level being a close third. Beat-level descriptors were the most commonly ranked as the second most important parameter, which may suggest that people are interested in looking at them in combination with either continuous or note-level data, rather than in isolation. Measure-level descriptors, in contrast, garnered both the lowest number of first place rankings and the highest number of last place rankings. This does not, however, mean that measure-level descriptors should be eliminated from our encoding format. Since the question regarding encoding levels forced the survey participants to linearly order the encoding levels, we are only able to assess relative preference and not their views on how useful the individual levels would be to them. Thus, we cannot assume that just because the measure-level descriptors came last that they are of no interest at all to researchers.

Our main take-away message from this survey question is that while we may be initially focusing on implementing note-level descriptors, for an encoding format to be truly useful for the range of applications that music scholars are interested in, encoding of beat-level, measure-level, and, particularly, continuous descriptors must also be facilitated.

## 4. ENCODING

This section presents extensions to both Humdrum and MEI for encoding note-level descriptors. These extensions are exemplified using the performance descriptors detailed in Table 2 extracted from a recording of an excerpt from Schubert's 'Ave Maria', shown in Figure 4. It also compares the ease with which one can encode beat- and measure-level descriptors in Humdrum and MEI, as well as the challenge of encoding continuous data in either format.

The proposed encoding format has a number of potential applications. Within a piece of music, the performance data related to similar (or identical) musical materials can be compared to assess the why in which the performer marks particular materials within the piece, e.g., [10, 11]. Across multiple performances of the same piece, the amount of consistency within performances by the same person and across different performers can be evaluated to explore performer

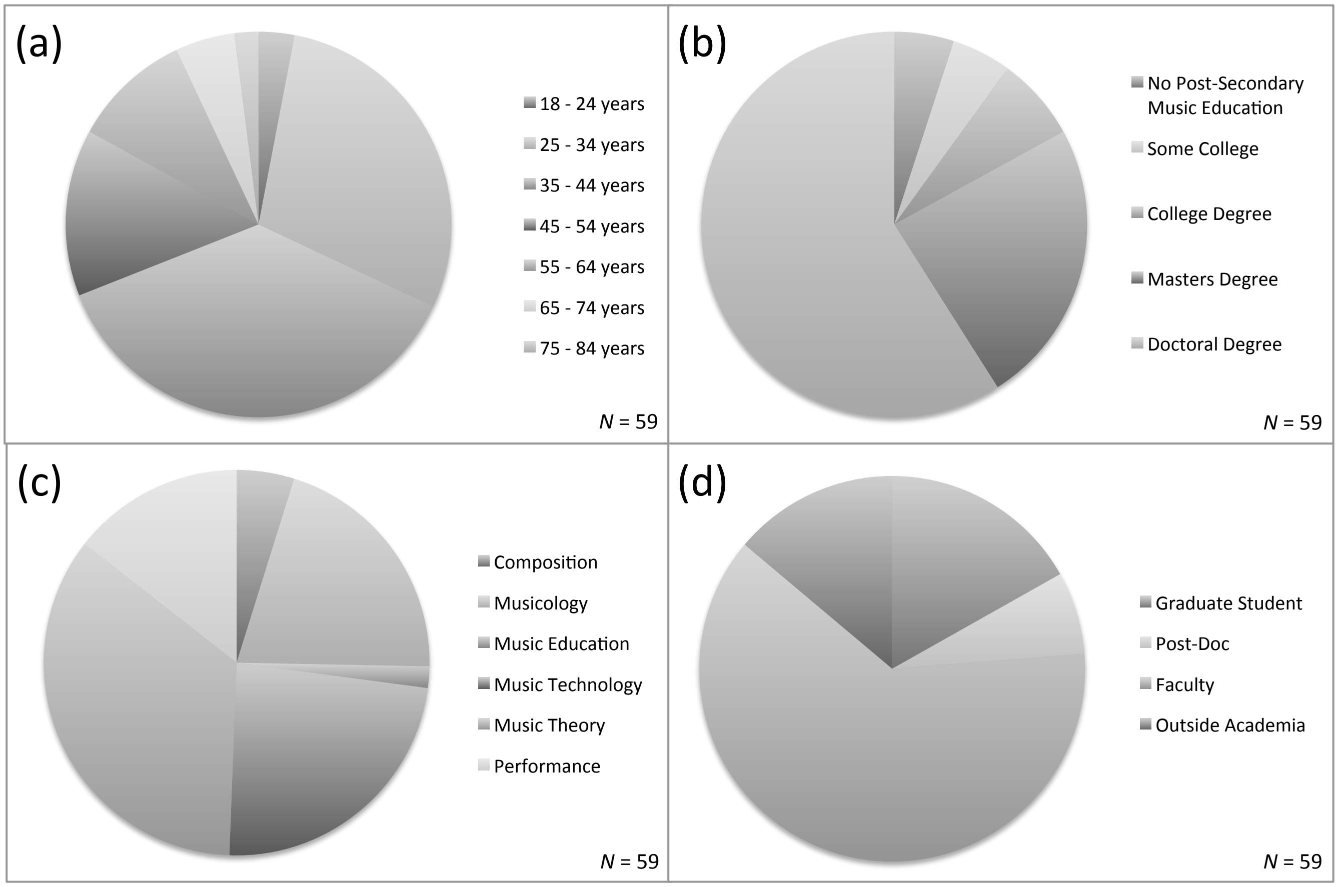


**Figure 1: Demographics of the music scholars who participated in our survey on encoding music performance data.**

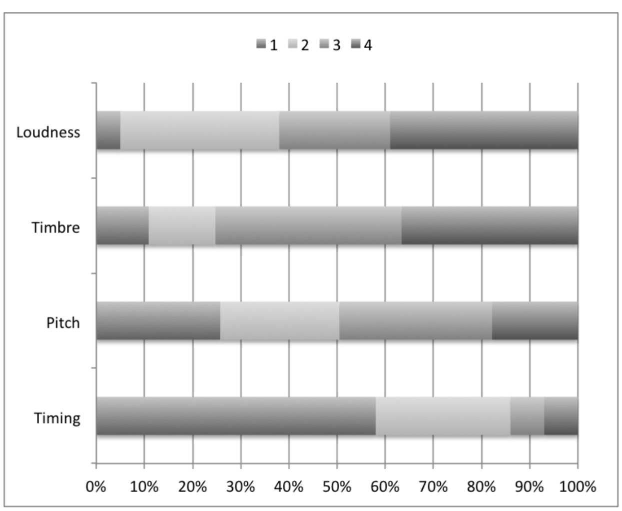


**Figure 2: Rankings of different performance parameters by survey participants.**

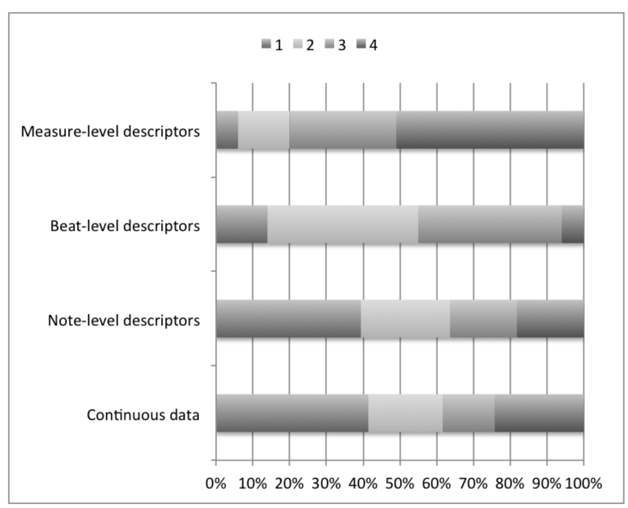


**Figure 3: Rankings of levels of description by survey participants.**

intentionality, e.g., [7]. Likewise, the encodings would facilitate longitude studies of performances of the same piece across time or geographical distance, e.g., [21].

Published research, such as cited in the previous paragraph and in Section 2, has demonstrated that it is possible to perform quality performance research without encoding the score and performance data. The limitation, however, is the re-usability of the data, either when shared with other researchers or used by the same researchers to address different research questions. Table 2 summarizes the performance corpora highlighted in Section 2, from it one can see that the difference between the levels of annotation in the absence of score data would make it difficult to ask the same research question across the datasets. An example of where this would be useful in practice is moving between note-level and beat/measure-level annotations. When only the performance data is stored, it becomes complicated to link the score post-hoc in order to figure out beat positions from note-level data or specific notes from beat/measure-level data, thus limiting the usability of the data. The main limitation of this approach is that generally more information will be encoded than is immediately useful to a single researcher, making the files larger and a little more unwieldy than they otherwise would be, but we believe that the benefits of sharing data widely amongst the research community outweigh this limitation.

The goal with defining encoding standards for both Humdrum and MEI is to provide both a human-readable format for researchers who may wish to encode some information manually and to accommodate researchers who use either Humdrum or Music21 toolkits. To facilitate the latter, one of our future plans is to expand the current Music21 Humdrum and MEI parsers to be able to read the performance encodings and to develop new Humdrum file processing programs and Music21 functions to perform analysis of the performance data. This is part of a larger-scale project entitled the Automatic Music Performance Analysis and Comparison Toolkit[4] (AMPACT) [9].

## 4.1 Note-level descriptors

The note-level performance descriptors shown in Table 2 are divided into four categories: timing, loudness, pitch, and timbre. Timing descriptors consist of inter-onset intervals, the amount of time between the start of sequential notes, and note duration, the amount of time between the onset and offset of each note. Loudness descriptors consist of the long-term loudness of each note, an estimate of perceived loudness for time-varying tones described in [13], and note-wise shimmer calculations, the mean of difference between frame-wise power calculations across each note. Pitch descriptors, based on frame-wise fundamental frequency estimates obtained using the algorithm described in [5], consist of perceived pitch of the sung pitch, using the model described in [14], the slope and the curvature of the F0 trajectory over the duration of the note, using the technique described in [8], estimates of the vibrato depth and rate, using the technique described in [6] and jitter, the mean of difference between frame-wise fundamental frequency estimates across each note. Timbre descriptors, all calculated using [19], consist of several attributes of the spectrum, namely its flatness, slope, flux, and centroid, as well as the overall harmonics-to-noise ratio.


**Figure 4: Opening vocal melody for Schubert's 'Ave Maria' used in encoding examples in Figures 5 and 6.**

### 4.1.1 Humdrum

Figure 5 shows the implementation of the descriptors in Table 2 at a note-level in Humdrum for the music shown in Figure 4. On the left of Figure 5 is the standard Humdrum notation for encoding score information in vertical spines, specifically the timing (**beat), pitch (**kern) and lyric (**silbe) information. These representation formats are pre-defined in the Humdrum User Manual [16] to facilitate shareability between researchers. But one of the strength of the Humdrum syntax is that it allows users to generate custom representation formats based on their specific needs. As such, the right side of the figure shows our newly-defined vertical spines for encoding performance data related to timing (**rtstart and **rtdur), pitch (**freq, **slopeF0, **curveF0, **vibdepth, **vibrate, and **jitter ), loudness (**loudness and **shimmer), timbre (**specflat, **specslope, **specflux, **speccent, and **hnr). The syntax proposed in Figure 5 allows for a standardized way to store performance data, facilitating easy comparison between different performances through the visual alignment of performance data with the score data, although each performance would have to be stored as a separate document. Since our proposed representation conforms with the Humdrum syntax, it is compatible with the Humdrum Toolkit[5], a set of more than 70 inter-related commands designed to manipulate and extract data. Furthermore, since the toolkit is UNIX-based, it grants scholars with coding skills the power to easily create custom tools that are compatible with the existing set of commands.

### 4.1.2 Music Encoding Initiative

Figure 6 shows the note-level descriptors for the same performance data encoded in Humdrum in Figure 5 in the **beat, **kern, and **silbe spines. The MEI protocol already includes a performance module, which can be used to link audio or video files. The links are created using XML IDs. XML IDs are attributes designed to create internal cross-references within an XML document, meaning they can be used to link different objects together, such as linking a score object to the corresponding performance attributes. In Figure 6, an XML ID was created for each note of the music excerpt and we have used these XML IDs to associate each note in the score to a list of performance descriptors. Each element in this figure correspond to the performance spines of Figure 5. Using this approach, we are able to encode data for several performances within the same document and query the document to retrieve and extract specific information.

[4]http://www.ampact.org

[5]http://www.humdrum.org/

Score | Performance: Timing | Pitch | Loudness | Timbre

```
!!!COM: Schubert, Franz Peter
!!!CDT: 1797-1828
!!!OTL: Ellens Gesang III ("Ave Maria")
!!!OPS: Opus 52
!!!ONM: No. 6
!!!SCT: D 839
!!Only first 3 mm. of the vocal part
**beat  **kern      **silbe  **rtstart  **rtdur    **freq     **slopeF0    **curveF0     **vibdepth  **vibrate  **jitter  **loudness  **shimmer  **specflat  **specslope  **specflux  **speccent  **hnr
*       *clefG2     *M4/4    *u=msec    *u=msec    *clefG2    *            *             *           *          *         *           *          *           *            *           *           *
*       *k[b-e-]    *        *MM60      *MM60      *k[b-e-]   *            *             *           *          *         *           *          *           *            *           *           *
*M4/4   *M4/4       *        *          *          *M4/4      *            *             *           *          *         *           *          *           *            *           *           *
*MM60   *MM60       *        *          *          *MM60      *            *             *           *          *         *           *          *           *            *           *           *
=1      =1          =1       =1         =1         =1         =1           =1            =1          =1         =1        =1          =1         =1          =1           =1          =1          =1
1       4.b-        A-       0          2900.5     456        -0.10373     -0.15858      27          4.5        0.3       23.484      18.8       0.06444     -0.00518     0.00089     0.02867     0.10361
2.5     16a         ve       2900.5     417.56     433        -0.053198    -0.09806      35          4.8        0.3       20.344      17.5       0.0703      -0.00516     0.001       0.02989     0.10961
2.75    16b-        Ma-      3406       484.86     460        -0.00041444  -0.093974     16          6.2        0.3       23.818      17.2       0.06633     -0.00518     0.001       0.02873     0.1223
3       4..dd       ri-      3901       3380.7     581        -0.030237    -0.11501      10          4.1        0.1       32.989      11         0.05972     -0.00517     0.00055     0.02913     0.13322
4.75    16cc        .        7344.9     319.25     508        0.026659     -0.0016503    10          3.1        0.1       20.321      10.5       0.05956     -0.00519     0.00067     0.02836     0.12893
=2      =2          =2       =2         =2         =2         =2           =2            =2          =2         =2        =2          =2         =2          =2           =2          =2          =2
1       4b-         a        7814.6     2116.3     455        0.0057001    0.27152       20          4.3        0.2       21.27       8.7        0.05987     -0.00518     0.00078     0.02897     0.12201
2       4r
3       2r
==      ==          ==       ==         ==         ==         ==           ==            ==          ==         ==        ==          ==         ==          ==           ==          ==          ==
```

**Figure 5: Proposed syntax to represent and link music performance data with score information using the Humdrum encoding format.**

| *Timing* | *Loudness* |
|---|---|
| Inter-onset interval | Long-term loudness |
| Note duration | Shimmer (Mean) |
| *Pitch* | *Timbre* |
| Perceived pitch | Spectral flatness (Median) |
| Slope of F0 trajectory | Spectral slope (Median) |
| Curvature of F0 trajectory | Spectral flux (Median) |
| Vibrato depth | Spectral centroid (Median) |
| Vibrato rate | Harmonics-to-Noise ratio |
| Jitter (Mean) | |

**Table 2: List of note-level descriptors used in encoding examples in Figures 5 and 6.**

## 4.2 Beat- and measure-level descriptors

Higher-level descriptors, such as tempo, tuning drift, average dynamic-level, and average spectral colour, can be calculated from the note-level descriptors summarized in section 4.1. These higher-level descriptors can be calculated as needed or explicitly encoded with the note-level descriptors.

There are some challenges, however, in simultaneously encoding data at different levels, such as the beat and/or measure level. For example, if we are interested in encoding both at the note- and the beat-level in Humdrum, one way to do so is to add the missing beats in the beat spine, to make sure that we have one event corresponding to each downbeat. This approach forces the encoder to add a dot character as a placeholder where there is no event in the kern spine. It can also be problematic when used in conjunction with some specific descriptors, such as the **rtdur descriptor proposed above. Since **rtdur is meant to calculate the duration between the current event and the preceding event, the results will be nonsensical when events are encoded simultaneously at more than one level. There are some ways to work around this issue; for example, one could write a small tool to separate information based on the encoding level, which would alleviate the above-mentioned problem. However, considering our goal to propose best encoding practices, this solution is less than ideal.

Alternatively, the MEI structure is more flexible. If a beat element was implemented in the MEI protocol, we could use the XML ID to link this beat element to the performance data in the same way we do for the note element. Using this approach, we don't have to worry about simultaneously encoding at different levels, since each level is associated with a distinct element.

## 4.3 Continuous data

Encoding data as continuous data is more challenging. At the moment, Humdrum does not allow to encode vectors of numbers rather than individuals values, making it impossible to associate more than one value to each element in the score. At the note level, it is possible to provide summaries of continuous data, such as the perceived pitch and the slope and curvature of the F0. These summaries have the advantage over continuous data of facilitating comparisons across notes of different lengths and provide an opportunity to use perceptually-relevant descriptors. These summaries cannot, however, capture all of the detailed variance in the original continuous data, making them less than ideal for some applications, such as timbral descriptors for which perceptually-driven summaries do not currently exist.

This limitation is not as severe in MEI, as multiple tokens can be associated with a single xml id. For example, a note-level descriptor such as perceived loudness could be annotated with multiple values to represent the evolution of the descriptor over different points in time (e.g. beginning, middle, and end) using the following syntax:

```
<when xml:id=”note_1”
perf:loudness=”23.484 29.354 31.365”/>
```

While this method is not continuous per se, it does provide an opportunity to evaluate the evolution of a descriptor over time.

## 5. CONCLUSIONS AND FUTURE WORK

Our main goal in this project is to make performance data available to music scholars through encoding formats that explicitly connect score information to performance data. To ensure that we were developing an approach to encoding performance data that would be useful to other scholars, we conducted a survey among researchers from a wide-range of music sub-disciplines. Based on these responses, we've proposed ways to encode different performance descriptors such as timing, loudness, pitch, and timbre in both Humdrum and MEI. From this survey, we learned that while note-level descriptors are a useful place to start, there is sufficiently strong interest in beat-level, measure-level, and, particularly, continuous data, that all of these levels of performance data must be accommodated for an encoding format to truly facilitate the range of applications that music scholars are interested in working on.

Going forward, we plan to explore ways to encode continuous data directly, not just note-level summaries. We are also going to work on the issues related to encoding beat- and measure-level information, specifically the ability to encode this type of data along with note-level and/or continuous data. This will allow for higher-level descriptors, at longer-range time scales, to be captured and described. Second, we are interested in creating tools to help researchers convert spreadsheets into a standardized format. We want to create a web-based application that can take information from a spreadsheet and format it in a standardized way. We also want to create tools for music scholars to easily import their data in existing toolkits such as Music21[6]. Ultimately, we believe that music scholars, as well as researchers working in other social fields, would benefit from a basic training in computer sciences. But in the mean time, we are hoping that these measures will encourage music scholars lacking strong technological background to consider a computational approach when warranted by their research endeavors.

# 6. ACKNOWLEDGMENTS

This work was supported by the National Endowment for the Humanities' Office of Digital Humanities under Grant number HD-228966-15.

# 7. REFERENCES

[1] C. Cannam, C. Landone, M. Sandler, and J. P. Bello. The Sonic Visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 324–7, 2006.

[2] N. Cook. Performance analysis and Chopin's Mazurkas. *Musicae Scientiae*, 11(2):183–207, 2007.

[3] N. Cook. *Beyond the Score: Music as Performance.* Oxford University Press, 2014.

[4] J. Dai, M. Mauch, and S. Dixon. Analysis of intonation trajectories in solo singing. In *Proceedings of the International Society for Music Information Retrieval conference (ISMIR)*, pages 42—6, 2015.

[5] A. de Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917–30, 2002.

[6] J. Devaney. Recapturing the data in seashore's musical performance measurements. *Musicae Scientae*, 19(2):214–22, 2015.

[6] http://web.mit.edu/music21/

[7] J. Devaney. Inter- versus intra-singer similarity and variation in vocal performances. *Journal of New Music Research*, In Press.

[8] J. Devaney, M. Mandel, and I. Fujinaga. Characterizing singing voice fundamental frequency trajectories. In *Proceedings of the Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 73–6, 2011.

[9] J. Devaney, M. Mandel, and I. Fujinaga. A study of intonation in three-part singing using the automatic music performance analysis and comparison toolkit (AMPACT). In *Proceedings of the International Society for Music Information Retrieval conference (ISMIR)*, pages 511—6, 2012.

[10] J. Devaney, J. Wild, and I. Fujinaga. Intonation in solo vocal performance: A study of semitone and whole tone tuning in undergraduate and professional sopranos. In *International Symposium on Performance Science (ISPS)*, pages 219–24, 2011.

[11] A. Dodson. Expressive asynchrony in a recording of chopin's prelude no. 6 in b minor by vladimir de pachmann. *Music Theory Spectrum*, 33(1):59–64, 2011.

[12] B. Gingras and S. McAdams. Improved score-performance matching using both structural and temporal information from midi recordings. *Journal of New Music Research*, 40(1):43—57, 2011.

[13] B. R. Glasberg and B. C. J. Moore. A model of loudness applicable to time-varying sounds. *Journal of the Audio Engineering Society*, 50(5):331–42, 2002.

[14] H. Gockel, B. C. Moore, and R. P. Carlyon. Influence of rate of change of frequency on the overall pitch of frequency-modulated tones. *Journal of the Acoustical Society of America*, 109(2):701–12, 2001.

[15] K. Hirata, K. Noike, and K. Haruhiro. A proposal for a performance data format. In *Proceedings of the Workshop on Methods for Automatic Music Performance and their Applications in a Public Rendering Contest (IJCAI-03)*, pages 65—9, 2003.

[16] D. Huron. *The Humdrum Toolkit: Reference Manual.* Center for Computer Assisted Research in the Humanities, Menlo Park, California, 1995.

[17] N. Ide and J. Véronis. *Text Encoding Initiative: Background and contexts*, volume 29. Springer Science & Business Media, 1995.

[18] M. Marchini, R. Ramirez, P. Papiotis, and E. Maestre. The sense of ensemble: a machine learning approach to expressive performance modelling in string quartets. *Journal of New Music Research*, 43(3):303–317, 2014.

[19] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams. The Timbre Toolbox: Extracting audio descriptors from musical signals. *Journal of the Acoustical Society of America*, 130(5):2902–2916, 2011.

[20] P. Roland. The Music Encoding Initiative (MEI). In *Proceedings of the First International Conference on Musical Applications Using XML*, pages 55–59, 2002.

[21] R. Timmers. Vocal expression in recorded performances of schubert songs. *Musica Scientiae*, 11(2):237–68, 2007.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<mei>
    <meihead>
      <fileDesc>
        <titleStmt>
          <title>Ellens Gesang III ("Ave Maria"), D 839</title>
          <title type="subtitle">op. 52, no. 6</title>
            <respStmt>
              <persName role="creator">Schubert, Franz Peter</persName>
            </respStmt>
        </titleStmt>
        <pubStmt>
          <date>1797-1828</date>
        </pubStmt>
        <annot>Only first 3 mm. of the vocal part</annot>
      </fileDesc>
    </meihead>
<music>
    <body>
      <score>
        <scoreDef>
          meter.count="4"
          meter.unit="4"
          key.sig="2f" key.mode="major"
          clef.shape="G"
          clef.line="2"</scoreDef>
        <section>
          <staff n="1">
            <measure n="1">
              <beat
                xml:id="measure_1_beat_1"
                tstamp="1"
                dur="4"/>
              <beat
                xml:id="measure_1_beat_2"
                tstamp="2"
                dur="4"/>
              <note
                xml:id="note_1"
                tstamp="1"
                dur="4"
                dots="1"
                oct="4"
                pname="b"
                accid.ges="f"
                syl="A-"/>
              <note
                xml:id="note_2"
                tstamp="2.5"
                dur="16"
                oct="4"
                pname="a"
                syl="ve"/>
              <note
                xml:id="note_3"
                tstamp="2.75"
                dur="16"
                oct="4"
                pname="b"
                accid.ges="f"
                syl="Ma-"/>
              <note
                xml:id="note_4"
                tstamp="3"
                dur="4"
                dots="2"
                oct="5"
                pname="d"
                syl="ri-"/>
              <note
                xml:id="note_5"
                tstamp="4.75"
                dur="16"
                oct="5"
                pname="c"/>

            </measure>
            <measure n="2">
              <note
                xml:id="note_6"
                tstamp="4.75"
                dur="4"
                oct="4"
                pname="b"
                accid.ges="f"
                syl="a"/>
              <rest
                tstamp="4.75"
                dur="4"/>
              <rest
                tstamp="4.75"
                dur="2"/>
            </measure>
          </staff>
        </section>
      </score>
    </body>
    <performance>
      <when xml:id="measure_1_beat_1"
          perf:rtstart="00:00:00.0"
          perf:rtdur="00:00:02.4004"
          perf:loudness="23.484"/>
      <when xml:id="measure_1_beat_2"
          perf:rtstart="00:00:02.4004"
          perf:rtdur="00:00:02.5065"
          perf:loudness="24.427"/>
        <when xml:id="note_1"
          perf:rtstart="00:00:00.0"
          perf:rtdur="00:00:02.9005"
          perf:freq="456"
          perf:slopeF0="-0.10373"
          perf:curveF0="-0.15858"
          perf:vibdepth="27"
          perf:vibrate="4.5"
          perf:jitter="0.3"
          perf:loudness="23.484"
          perf:shimmer="18.8"
          perf:specflat="0.06444"
          perf:specslope="-.00518"
          perf:specflux="0.00089"
          perf:speccent="0.02867"
          perf:hnr="0.10361" />
        <when xml:id="note_2"
          perf:rtstart="00:00:02.9005"
          perf:rtdur="00:00:00.41756"
          perf:freq="433"
          perf:slopeF0="-0.053198"
          perf:curveF0="-0.09806"
          perf:vibdepth="35"
          perf:vibrate="4.8"
          perf:jitter="0.3"
          perf:loudness="20.344"
          perf:shimmer="17.5"
          perf:specflat="0.0703"
          perf:specslope="-0.00516"
          perf:specflux="0.001"
          perf:speccent="0.02989"
          perf:hnr="0.10961" />
        <when xml:id="note_3"
          perf:rtstart="00:00:03.406"
          perf:rtdur="00:00:00.48486"
          perf:freq="460"
          perf:slopeF0="-0.00041444"
          perf:curveF0="-0.093974"
          perf:vibdepth="16"
          perf:vibrate="6.2"
          perf:jitter="0.3"
          perf:loudness="23.818"
          perf:shimmer="17.2"
          perf:specflat="0.06633"
          perf:specslope="-0.00518"
          perf:specflux="0.001"
          perf:speccent="0.02873"
          perf:hnr="0.1223" />
        <when xml:id="note_4"
          perf:rtstart="00:00:03.901"
          perf:rtdur="00:00:03.3807"
          perf:freq="581"
          perf:slopeF0="-0.030237"
          perf:curveF0="-0.11501"
          perf:vibdepth="10"
          perf:vibrate="4.1"
          perf:jitter="0.1"
          perf:loudness="32.989"
          perf:shimmer="11"
          perf:specflat="0.05972"
          perf:specslope="-0.000517"
          perf:specflux="0.00055"
          perf:speccent="0.02913"
          perf:hnr="0.13322" />
        <when xml:id="note_5"
          perf:rtstart="00:00:07.3449"
          perf:rtdur="00:00:00.31925"
          perf:freq="508"
          perf:slopeF0="0.026659"
          perf:curveF0="-0.0016503"
          perf:vibdepth="10"
          perf:vibrate="3.1"
          perf:jitter="0.1"
          perf:loudness="20.321"
          perf:shimmer="10.5"
          perf:specflat="0.05956"
          perf:specslope="-0.00519"
          perf:specflux="0.00067"
          perf:speccent="0.02836"
          perf:hnr="0.12893" />
        <when xml:id="note_6"
          perf:rtstart="00:00:07.8146"
          perf:rtdur="00:00:02.1163"
          perf:freq="455"
          perf:slopeF0="0.0057001"
          perf:curveF0="0.27152"
          perf:vibdepth="20"
          perf:vibrate="4.3"
          perf:jitter="0.2"
          perf:loudness="21.27"
          perf:shimmer="8.7"
          perf:specflat="0.05987"
          perf:specslope="-0.00518"
          perf:specflux="0.00078"
          perf:speccent="0.02897"
          perf:hnr="0.12201" />
    </performance>
</music>
</mei>
```

**Figure 6: MEI syntax example with standard score information encoding format and the proposed performance encoding format.**