The 'Measure Map': an inter-operable standard for aligning symbolic music

Mark Gotham
Durham U. and U. of Cambridge
Durham and Cambridge, UK

Johannes Hentschel
École Polytechnique Fédérale de Lausanne, Switzerland

Louis Couturier
Univ. Picardie Jules Verne, MIS
F-80000 Amiens, France

Nathan Dykeaylen
University of Cambridge
Cambridge, UK

Martin Rohrmeier
École Polytechnique Fédérale de Lausanne, Switzerland

Mathieu Giraud
Univ. Lille, CNRS, UMR 9189 CRIStAL
F-59000 Lille, France

ABSTRACT

Aligning *versions* of the *same source* material has been a persistent challenge in the field of digital libraries for musicology, and a barrier to progress. The growing number of publicly accessible symbolic datasets (of scores, analyses, and more) now increasingly cover multiple versions of the same works. As creators/curators/representatives of many such datasets and encoding standards, we came together in this project to coordinate platform-neutral interoperabilility for combining and comparing different sources, reliably and automatically. Here, we outline the main challenges and propose solutions centred on the 'measure map': a lightweight format for representing symbolic bar information alone. We offer new code for producing this representation from various formats, diagnosing differences, and even solving for those differences by modifying sources in-place. While we cannot solve for every possible discrepancy, we do provide corpus-scale demonstration; and while we focus on symbolic data, we consider the measure map also a useful basis for aligning audio, manuscripts and any source for which bar-relative location data provides a useful point of reference.

CCS CONCEPTS

• **Applied computing** → **Sound and music computing**; • **Human-centered computing** → **HCI theory, concepts and models**; • **Information systems** → **Music retrieval**; • **General and reference** → **Validation**; **Reliability**; **Evaluation**; **Reference works**.

KEYWORDS

Digital Music Library, Corpus study, Music Information Retrieval, Musical Scores, Musical Analyses, Alignment, Interoperability

ACM Reference Format:
Mark Gotham, Johannes Hentschel, Louis Couturier, Nathan Dykeaylen, Martin Rohrmeier, and Mathieu Giraud. 2023. The 'Measure Map': an inter-operable standard for aligning symbolic music. In *10th International Conference on Digital Libraries for Musicology (DLfM 2023), November 10, 2023, Milan, Italy*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3625135.3625136

DLfM 2023, November 10, 2023, Milan, Italy

ACM ISBN 979-8-4007-0833-6/23/11.
https://doi.org/10.1145/3625135.3625136

1 INTRODUCTION

Western music has been notated in 'bars' (US: 'measures') for hundreds of years (§2), partly to coordinate different instruments/voices and to measure musical time. For digital score encodings and the applications that create, manipulate, or evaluate them, bars play a crucial role for score addressability and alignment information.

Recent years have seen great growth in the number and range of datasets available and this now includes multiple versions of the same music (e.g., chorale X by composer Y in score editions A, B, & C and analyses P, Q, & R). Reliable alignment between these sources is clearly desirable for integration and comparison. Use cases that require this alignment include visualisations of the sources side-by-side, and machine learning for automatic analysis based on learning the relationship between a score and one or more (e.g., harmonic) analyses. And while there is a growing number of use cases that would benefit from alignment, most existing 'solutions' depend on hard-coded variants of their sources to ensure alignment (see [14], §2). We ought to be able to align sources *flexibly*, keeping those originals intact. (Potentially aligned) corpora stand as both a motivation for, and a test case of this alignment.

While the measurement of symbolic time may seem simple, there are many ways for two sources to diverge, including different: *pre-digital conventions* in the editions (§2), *digital encodings* of those sources (§3), and *accidental* divergence from these recognised norms. To address this, we propose the 'measure map': a platform-neutral representation of only the necessary information for each bar. We propose both a 'verbose' form with extensive data, and a lossless, 'compressed' version with only the minimum information needed to map back to the verbose form. Moreover, we provide functionality for diagnosing differences between sources at the level of these measure maps. In addition to being platform neutral, this also is very lightweight and can complement more computationally intensive comparisons on the source's actual content, for example with self-similarity matrices.

We provide conversion to measure maps from several of the score and analysis formats in use today. One strength of this report is that it comes from several currently active protagonists of the corpus creation movement, who collectively represent many of those different standards and formats. While we certainly do not claim to have solved for every use case, we have at least dealt with this problem in a range of real contexts (e.g., see §5). And while we focus on symbolic cases, the tools are relevant to wider alignment issues (e.g., with audio) wherever bars are used as a point of reference.

2 PRE-DIGITAL HISTORY

Now-familiar notations like barlines, bar numbers, and rehearsal figures entered Western notation gradually over the centuries. Barlines were standardised in the mid- to late- seventeenth century, initially on tablature music.[1] These barlines emerged to serve several functions. On the more *musical* side, barlines interact with related symbols (time signature, beaming, …) to indicate *metrical structure*, with barlines falling before strong beats and with bar-relative positions likewise carrying musical (metrical) meaning. On the *practical* side, barlines provide a method for coordinating multiple parts and referring to specific moments.[2]

The need for coordination and reference increases as works become longer, larger, and more variously set out. For instance, where there is a fixed-page layout, bars are sometimes referred to by reference to their location. In 'L'Art de toucher le Clavecin' (1716), François Couperin references to *'les deux dernieres mesures des portées 5, et 6'* ('the two last bars of staff 5 and 6').[3] This is still familiar in choirs, where everyone (all singers and any conductor involved) typically use the same, full score. And this practice has continued alongside the emergence of other ways for referring to bars, including with *rehearsal numbers or figures* (early 19th century, after older practices) and later still the *bar number* (early 20th).[4]

2.1 Within- and/or between-edition consistency

To be clear, we have so far discussed *within-edition* coordination and consistency. Composers, publishers, and anyone else responsible for producing a single score must clearly aim to produce internally consistent notation. At least for 'common practice' music, if reference points like bar numbers or rehearsal marks differ between the parts then that is an error and rehearsals will be in chaos.

By contrast, there is not the same expectation for *between-edition* consistency. There is only *some* expectation to observe recognised conventions that will allow musicians to read the music fluently without undue barriers. In the creation of an *alternative* edition to an existing work, it may even be in a publisher's interest to establish clearly unique elements in their score to make the case for why musicians should buy and use this version over another, and to avoid appearing to copy another editor's work.

Between-edition consistency might be considered useful by consumers comparing those editions, (and certainly when we try to do this at scale in corpus study) but it is simply not a motivating force for editors. As such we can expect between-edition difference to include the page and system layout, the use of rehearsal markings (especially where these are not provided by the composer), and yes, even the *measure numbering*.

[1] For histories, see [6, 11, 26].

[2] Barring is not without its detractors, of course. Many early music specialists lament this historical development as pedantic and inhibiting to the natural flow of music. And likewise much modern music dispenses with this practice. We focus here on Western music of the 'common practice' and return to the limits of that remit in §5. All this is beyond the scope of our focus on common practice notation.

[3] See https://gallica.bnf.fr/ark:/12148/bpt6k1168974z/f60.item, p.48

[4] Beethoven's *Grosse Fuge* (pub.1827) 'appears to be the first work ever to have been allocated rehearsal letters. […] Although the fifteenth-century use of the *signum congruentiae* in vocal parts is somewhat similar, no such device was in use in the early nineteenth century, and bar numbers were not used for this purpose until nearly a century later. Rehearsal letters can be found in orchestral scores by Mendelssohn and Spohr from the early 1830s, but none are known from the 1820s in scores by these or other composers.' [4].

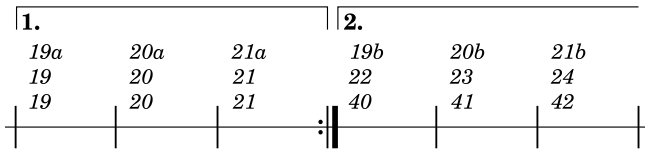

Figure 1: Three ways of numbering first and second endings.

2.2 Different conventions for numbering bars

There are different typographical conventions in terms of *when and where* bar numbers are indicated.[5] More significant for between-edition consistency is the question of *how to number those bars*. For example, there are at least three conventions for numbering first/second time bars, counting:

(1) 1st/2nd time *versions* with suffixes (fig.1, top);
(2) each bar *printed* in the score separately (fig.1, middle);
(3) each *performed* bar separately, including double counting of repeated bars (fig.1, bottom, which assumes the start repeat is at b.1, so b.22–39=1–18).

So, even for *pre-digital* sources, for a consideration as apparently simple as *numbering bars*, and assuming *no inadvertent errors*, we may already face significant differences between versions of the same material. And this is only one of the many issues found in pre-digital sources.[6] We will address that wider range of problems (§3.2) as part of addressing specifically digital considerations (§3).

3 DIGITAL, SYMBOLIC ENCODING

Despite the much shorter history (since c.1950), *digital representations* of notated music have also used a range of approaches that rivals their pre-digital counterparts. In an age of increasing activity in empirical music research [24, 32], this range of encoding strategies poses a problem for comparison.

This section provides an overview of existing approaches to encoding the bar information of a score (§3.1), the problems that arise when bringing together sources from different encoding practices (§3.2), and an account of whether and how these problems have been addressed in prior work (§3.3). The uses of symbolic formats can be understood in relation to a three-way categorisation:[7]

(1) *print* – the visual layout, engraving, and display;
(2) *logic* – encoding semantic function/meaning; and
(3) *performance* – usually a combination of the first two.

Existing functionality focuses on these aspects to varying degrees in ways that necessarily affect how they handle bar information. As a running example, we use the frequently-seen case of a single *complete* bar visually split by an additional, within-bar 'barline' (or as [2] call it, a *pseudobarline*).

[5] Examples include marking *all* bar numbers, *none*, those at *each new system*, every fifth one [34], and at structurally significant moments [37]. They may also vary in *placement*: at the start of the bar, the middle, or the end (fig.4).

[6] For instance, The Chamber Music Conference gives advice for good practice in numbering bars and lists more than 600 works with bar counts for each movement (http://cmceast.org/resources/numbering-measures.php). More than 50 of these works have detailed comments due to repeats or discrepancies between editions or parts.

[7] We take inspiration for this categorisation from the separation of 'graphical', 'rational', and 'phonological' (or 'gestural') contexts in [31, p.7].

3.1 Print, logic, performance

The historically influential engraving system **SCORE** [33] gives full control over even minute typographical aspects of score elements. In this case, bars are present in the encoding solely in terms of the graphical elements required for humans to parse them visually. In this paper, we refer to *printed bars* when we speak of those visual aspects as they appear on the page. In SCORE, the typesetter sets a barline wherever one is to appear in the score and does not (need to) semantically determine whether it appears between- or within-measure. The consequence of this distinction is present only in the bar numbering, (if included), which is also created manually wherever a number is to be *printed*.

The **LilyPond** typesetting system [27] is also primarily concerned with encoding the visual aspects of the score and, hence, of the *printed* bars. In principle, however, it does not require users to explicitly declare each barline and bar number because the compiler includes the *logic* of how to print and count barlines according to the conventional rules (in combination with a wider array of commands and settings). Since the software has the logic built in, it can warn about inconsistencies in the encoding, leaving the human typesetter to focus on visual deviations from the norm such as our running 'pseudobarline' example.[8] These are treated visually and therefore do not encumber the bar-counting and bar-checking logic.

The documentation and design of **Humdrum**'s `**kern` format [21, 22],[9] emphasise the distinction between visual, logical and performance-related aspects of encoded scores. Users must encode bars both *logically and visually*. This includes writing the equal symbol =, followed by a bar number (and optionally a single letter) as well as additional *visual* specifications of the barline, if any. The difference becomes clearest in the guideline to encode an invisible barline ('=1-') at the beginning of a (non-anacrustic) piece [23, ch.2]. This is not printed but informs processing software about the beginning of the first bar. As in the LilyPond case, the logical bar can be split by introducing an additional (pseudo)barline—one without bar number in this case.

Today, most symbolically encoded music notation is produced in commercial or open-source graphical score editors based on XML schema (Dorico, Finale, MuseScore, Sibelius) and offer interoperability via **MusicXML** [13]. These programmes frequently have a focus on the visual aspects of *printed* notation and on synthesised playback. By contrast, the XML schemas of the music encoding initiative (**MEI**) focus more on logical aspects.[10] Growing adoption of MEI would significantly affect the overall picture.[11]

Few editors and formats are able to encode split bars and count them correctly. Instead, most encode two bars with irregular lengths. Users must then prevent the second from being counted in its own right. Figure 2 shows an example of a common default behaviour which has to be corrected, either manually or with advanced settings like MuseScore's 'score wizard' or the 'MEI-friend' editor's support for standard re-numbering correction.[12]

[8] In practice, many encoders use the pipe | to segment voices and stay oriented.
[9] Humdrum itself has been designed for the coordinated representation of any type of symbolically encoded, potentially multi-layered information organised in time.
[10] https://music-encoding.org/resources/schemas.html
[11] We note imminent plans for MEI as an export format of MuseScore v.4.2.
[12] https://mei-friend.github.io/docs/basic/settings/#renumber-measures

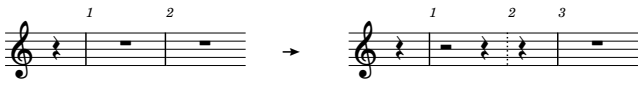

Figure 2: Incorrect numbering after splitting a bar.

3.2 Common problems

The advent of digital scores improved many of the problems that were common in the pre-digital era, particularly for contemporary, unpublished, and other musical repertoires that have not benefited from extensive editorial fine-tuning and the scrutiny that comes with multiple performances. The automatic creation of parts directly from a full (conductor's) score did away with the time consuming and error-prone engraving of parts separately, manually. Furthermore, digital score encodings have blazed a trail for empirical studies on notated music, eventually contributing to the foundation of new research fields such as Music Information Retrieval (MIR). It can also be speculated that it is the use of software that made it possible in the first place to conveniently—because automatically—include bar numbers by default.

At the same time, new technical solutions widen the scope of possibilities and challenges, create an appetite for further exploration and advancements, and generate new promises, stakes, and issues. In the disciplines mentioned, for example, the availability of symbolic score encodings at scale has supported the field of musical corpus studies [32], but has also created a need for a better worldwide alignment of research data, in terms of semantic networks of metadata [e.g., 28, 36]) quality assessment [e.g., 8, 10], the interoperability of formats [e.g., 3, 18], or the alignment of resources [e.g., 14, 25]. We focus here on that last topic, and on dealing with the most frequent causes for mis-alignments between scores.

For the following itemisation of main issues, let us consider the real-world example of an *analytical annotation* dataset that is *manually produced off-score* in consultation with printed scores but then needs to be aligned with *digital* copies of those scores. Aligning off-score labels with the corresponding score encoding raises the problem of unambiguously translating bar number and beat positions from one source to another. The annotator reads bar and beat numbers from the *source* (e.g., a printed, scholarly edition); but does the digital edition *target* count bars in the same way?

3.2.1 Split bars. We begin with the aforementioned (and common) case of split bars. These splits can occur for many different reasons, for instance, to introduce repeat markings, line- or section-breaks, or to highlight an anacrusis (and not necessarily only at the start, see fig.3). Most XML-based codes divide this music into two individual `<measure>` tags which correspond to *printed* bars as shown in fig.2. All subsequent bar numbers are affected.

This is simple enough in some cases, but can quickly get more complex, particularly when the lengths of the two irregular `<measure>` tags do *not* add up. This sometimes occurs at section breaks, particularly where they also include a time signature change as shown in fig.3. In this case, the *logical* bar 237 is composed of three *printed* bars, which, in turn, group into two *performed* bars: the first ending (a complete bar which may be labelled `237a`), the second ending (`237b` perhaps?), which in this case is an irregular split bar consisting of a cadenza-like component of length 4/4, and finally the 3/8

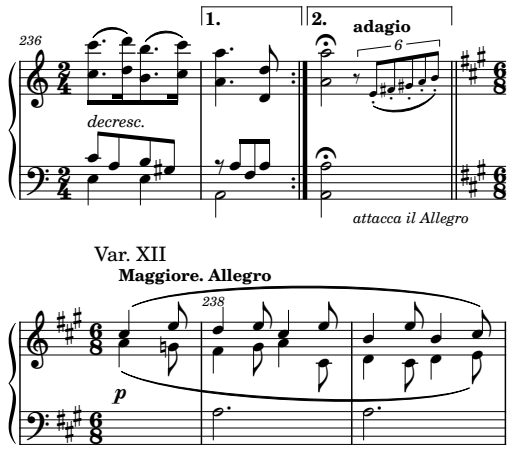

Figure 3: Transition from var.XI to XII in Beethoven's *12 Variations on the Russian Dance from 'Das Waldmädchen'*.

duration anacrusis to bar 238 at the start of the new section. Even if the annotator's source is precise enough to specify `237b` (rather than `237`), plenty can still go wrong both in the source encoding and the target decoding, especially when originally specified as a time signature-dependent beat. (Which beat unit should one assume for this tripartite bar of length 11/8?). Cadenzas and other un-metered passages complicate this matter yet further.

There are many viable solutions to such a problematic case, the issue is that none of them can be considered standard or expected from a new source or a human annotator, and that considerable manual labour needs to go into accessing the score encodings and manually adjusting addresses to achieve a reliable alignment.

3.2.2 The 'anacrusis' or 'pickup'. It is a widely accepted convention that the first full bar of a piece is numbered as '1' and an anacrustic ('pickup') bar before this does not count (is numbered 0). Bars that are split in two parts face the same behaviour: numbering of the second is skipped to ensure the continuity of the full bar numbering overall. Whereas most printed editions take this rule into account, this is rarely the default behaviour of music notation softwares. So in addition to fig.2's confusion over the split bar counting, it may also have defaulted to starting with a pickup bar numbered 1, and thus adding 1 to all the numbers shown.

3.2.3 Section breaks. This same anacrusis / pickup issue applies to incomplete bars beginning new sections, and thus to the start of the *Maggiore* section in fig.3. What we see is a typical example of contradictory norms. In this case, while many printed, scholarly editions opt to reset bar numbers at the beginning of a new section, many digital encodings default to 'counting through' an entire work. Other common contexts for this possible re-start of the numbering at section breaks include minuet movements with an internal trio section, and predominantly fast movements with a slow introduction.

3.2.4 Alternative endings. As we have already seen in §2.2, there are at least three different conventions for counting the bars that are part of different alternative endings, namely counting *printed*, *logical*, or *performed* bars. As an additional source of confusion, the brackets that indicate alternative endings are often encoded separately from the bars in question,[13] and are prone to misalignment or even simply *getting lost entirely* when converting between formats.

3.2.5 Repeated segments, jumps, flow control. Repeat signs, alternative endings, and jumps such as *da capo al fine* or *dal segno al coda* originated for pre-digital reasons such as the price and availability of paper, and the practicality of (physical) page-turns. As digital media increasingly replace paper formats, it is not surprising to find this practice become rarer, with (some) digital editions preferring to write out the flow of the music without (or with fewer) of these repeat/jump markers. In addition, this development may be influenced by the growing role of the Digital Audio Workstation where sequencers and loop machines are visually arranged over an audio timeline, i.e., one that is measured in *performance* time rather than the alternatives. Aligning compressed repetitions with uncompressed equivalents require decompressing the former or compressing the latter. In the case of score encodings, this boils down to a doubling of potential problems that can arise from the other possible discrepancies discussed.

3.3 Existing (digital) approaches

Since aligning sources plays an important role in the context of (1) research or software projects, (2) the encoding philosophy around a particular format, and (3) corpus building initiatives, several (partial) solutions for alignment have emerged. This section will outline the general principles that underlie them and which, by extension, inform the standard that we propose to bridge the existing approaches into something more interoperable.

Viable solutions all encode a registry of addresses, sometimes (but not necessarily) in a meaningful order. Continuing with the terminology introduced in §3, such a registry needs to convey a means to calibrate information along at least one of the three possible timelines: the *visual*, the *logical*, and/or the *performed*.[14] We can think of the existing approaches in terms of *measuring rods*.

3.3.1 Different measuring rods for scores. **Humdrum**'s `**kern` format is inherently close to the idea of a musical timeline. It resembles a printed score in a single, long system turned clockwise by 90°, and encodes simultaneous score events on the same *line* of a text file, and time passes from top-to-bottom, with subsequent lines indicating visual (or logical) succession. Thanks to the semantic encoding of both visual and logical bars, it is trivial to retrieve a list of the encoded bar numbers and/or barlines as needed.

Furthermore, the position of each score element can be flexibly represented by offset from another position (such as the beginning of a bar) or from the piece's beginning. This can be achieved, for example, by summing durations converted to absolute time using a metronome mark (e.g., as encoded in the score).

We highlight Humdrum here because it is one of the oldest and influential codes to have survived, and it is certainly among the

[13] This is partly because they span several bars. See, for instance, the music21 `spanner`.
[14] 'Timeline' is appropriate for all three conceptions assuming that a sequence of bar-like units can be translated to a duration in *musical*- and therefore *real* time.

most flexible and malleable representations available in terms of musical timelines.

3.3.2 Global offset in musical durations. Many formats use a grid of fixed musical durations for this 'score measuring rod', usually with the 'quarter note' serving as the reference unit. For example, the **Verovio** viewer [29] used by many projects to render **kern and MEI [17, 30] as scalable vector graphics (SVG) can also be used to create so-called 'timemaps' which represent all relevant addresses of a score in terms of an alignment between two timelines.

A timemap consists of an ordered array of JSON objects, each of which represents an address in the score at which the duration of at least one note head starts or ends.[15] Each address is expressed in terms of a `qstamp` and a `tstamp` both measured from the score's beginning (i.e., the first note or rest encoded). `qstamp` expresses that duration in terms of the symbolic 'quarter note' value (hence 'q'), while `tstamp` is the clock time duration (in milliseconds).

This segmentation of a score by all timestamps where a note head starts or ends, in combination with the IDs of the starting and ending heads, is immensely useful in interactive (web) applications e.g., to control a caret during playback. However, this representation does not reveal any information about the number, metre, or lengths of the bars encoded in the score, nor about the metrical positions of notes within them, and will therefore not be equally well suited for alignment with bar-based addressing schemes.

3.3.3 Global offset in bars and beats. Audio-to-score alignment is an MIR task that links the *performance* realm of audio recordings (clock time) with the *printed* and *logical* data of the score (symbolic musical time and sometimes visual placement). Score addressability plays a pivotal role in this endeavour and requires careful handling of any potential differences between the two domains. This includes all of the issues raised.

[9] proposed the *match* file format specification for this problem. Although it does not solve score-performance matching, it does offer a comprehensive addressing scheme that once again includes global `qstamp` and `qstamp` equivalents, but also bar numbers and metrical positions within bars. These files can be created from several encoding formats [3] and include all notes as well as the 'control flow' of the score, i.e., sections, repeats and more.

Altogether, this allows for programmatic manipulation of the score information into the right shape to match up with the performance, without need for creating an additional version of the score itself. That said, the documentation is silent on how ubiquitous problems such as bar numbering or split bars are to be handled, despite their playing a crucial role in alignment, as discussed above. Even if the inclusion of global offset positions resolves the internal handling of these cases, the format will benefit from our proposal through the increased interoperability that can be provided by making the associated intricacies public while using shared terms and concepts.

3.4 Towards a common practice

All things considered, there seems to be a convergence here with many projects, initiatives, and researchers settling on the relatively small and common denominator of the quarter note as a base unit for musical timelines. We briefly speculate on 3 possible reasons:

(1) music cognition: the quarter note has the advantage of being the basic beat unit in much common practice (and beyond);
(2) processing: smaller units produce slightly smaller rounding errors when represented as decimal fractions, so quarter notes are preferable over longer values like whole bars;
(3) interoperability, inertia, practicality: once someone has used a convention, it is more likely that others will follow.

It bears repeating that we aim for the new functionality to be platform-neutral, and to build on this prior work (both conceptual and computational) wherever possible. We are content to accept the `qstamp` as both a usable *measure* of symbolic time in general, and as a *name* specifically.

We can also build on the the extensive existing work for format parsing. The influential **music21** [7] libraries looms large here. For our purposes, it provides a shared routine for parsing many of the *score* formats discussed above (MusicXML, krn, …) as well as some score-like *analysis* formats such as `.rntxt` [35] (see §4.3). This is useful for extracting measure map information (§4) as well as for modifying that data in-place.[16]

music21 internally represents scores in a tree structure in which every element has a qstamp equivalent called `offset`. In addition to a time-aligned element tree it provides users with several types of `offsetMaps`, which map those timestamps to collections of score objects. Among these, the `measureOffsetMap`, is relevant here as it serves to summarise `stream.Measure` objects in the score with their `offset` (again, from the start).[17] This is conceptually relevant here, though somewhat limited in scope as only some of the information needed can be extracted directly (e.g., number and duration, but not repeat spans).

We decided against full implementation of the measure map within music21 for two main reasons. First, there is now a move to rationalise the code base to the exclusion of new functionality, perhaps partly to focus on the core routines including the multi-format parsing that we benefit from as discussed above.[18]

The second reason, perhaps related, is that music21 does not inter-operate with everything. One wide-spread XML score format that music21 cannot parse is MuseScore's `.mscx`. This is covered by the library **ms3** [20]. which can be used to *extract* aspects of MuseScore 3 and 4 files (including notes and bars) and to store that data in a uniform tabular format. ms3 also offers the corresponding task of *insertion*, though for this we need the bar numbering consistency discussed above.[19]

The extracted table also includes a running count of `<measure>` tags (column `mc`) starting with 1 (so that it corresponds to the numbers that MuseScore displays in the status bar) as well as other

[15] To date (Verovio 3.16.0) ties between notes are not represented or taken into account.

[16] This links to the third itemised point above, as such central libraries play a large role in this standardisation. music21 is the entry point for many musicologists moving into programming. It explicitly encourages that with a user guide that assumes no prior computational knowledge.

[17] For multi-part music this includes lists of the `stream.Measure` objects starting at the same position. For instance, standard and well-formed works for SATB choir or string quartet will have exactly 4 entries in every such list.

[18] See https://groups.google.com/g/music21list/c/HF3tgkMvNWI, §3.

[19] Currently ms3 enforces one specific set of bar numbering rules and, whenever a file is parsed, warns users about any deviations in the encoding.

columns required to fully specify the bars included in a score, including their time signatures, actual durations, global offsets, repeat signs and control flow markers (such as backward and forward jumps). In addition, *ms3* infers from this information an additional column that specifies for each bar the IDs of all bars that can follow, in the order of times that same bar is reached by the control flow. This information can be used to unfold ('flatten') repeats before extraction in order to make the stored tables correspond to a 'playthrough' version, which comes with the additional column `playthrough_mn` which disambiguates the bar numbers by applying the convention with appended lowercase characters (a, b, c, …) to disambiguate several occurrences of the same bar number.

Finally, the web platform **Dezrann** enables users to view, edit, and share music analyses through labels on a score or other representation of music [12]. Labels are referenced by their `qstamp`, but the user interacts on the platform through bars and beats, such as '9-1/4'. Again, this 'Measure Map' study is partly motivated by the need (shared by Dezrann and many platforms) for handling those complex measure numberings reliably.

Building the 'Measure Map' explicitly in relation to music21 (xml, krn, …), ms3 (MuseScore), and Dezrann helps us to ensure strong inter-operability and wide potential use cases from the outset.

4 THE 'MEASURE MAP' (MM)

In light of the above, we propose the 'Measure Map' (hereafter 'MM') for capturing the information about each bar-like unit of a symbolic encoding that is *essential for the alignment* of sources. We begin with the full set of properties and their types, with constraints and short descriptions. Section §4.1 expands on how the fields relate, including the nature of their relative priority. For the most complete information, please see the formal JSON schema describing this specification at https://github.com/measure-map/specification.

ID `string`
: Any unique string to identify this bar object. The next array may use the **ID** to refer to other bars.

count `integer, minimum = 1`
: Position of this bar object in the MM, using natural numbers starting with 1.

qstamp `number, minimum = 0`
: The symbolic time to have elapsed since the beginning of the source, measured in quarter notes.

number `integer, minimum = 0`
: A number assigned to this bar based on a set of conventions.

name `string`
: A label for the bar, typically used for distinguishing between bars with the same **number**, e.g. `19a` and `19b` for first and second ending as infig.1).

time_signature `string`
: The time signature label is usually in the form `<int>/<int>` (e.g., '3/8'), but can be any text (e.g., 'C', 'common time', 'unmeasured' 'cadenza') as long as **actual_length** is specified.

nominal_length `number or null, minimum = 0`
: The default quarter length duration that corresponds to the given **time_signature**. Can take the value `null` if the time signature has no corresponding duration.

actual_length `number, minimum > 0`
: The actual duration of the bar, in quarter notes. This is usually the same as the **nominal_length**, though it can be shorter (e.g., for anacruses) or longer (e.g., cadenzas).

start_repeat `boolean or number, default = false`
: Typically boolean type, with `true` indicating a start repeat (‖:) at the beginning of the bar. Alternative usage permits encoding the **qstamp** of a within-bar repeat mark.

end_repeat `boolean or number, default = false`
: Like **start_repeat**, but for (bar-)*end* repeat marks (:‖).

next `array[integer] or array[string]`
: The **ID** strings or **count** integers that correspond to all bars that can follow this one, in order of performance.

The minimal requirement for a MM is a sequence of at least two objects (first and last bar). Obvious choices for representing such a sequence are a table where rows correspond to objects and columns to properties, or JSON format. In our example corpus (§4.3) we include MMs in JSON format with the file extension `.mm.json`.

4.1 Expansion and compression

The same MM can take two different forms. In *expanded* form, the MM includes as many objects as the described source has bar-like units. The *compressed* form can be derived by omitting entries that can be generated unambiguously by applying a set of rules on their predecessor. A compressed MM can be expanded using this same set of rules which we include as part of the specification online.

As a motivating example, consider that a compressed MM can be *manually created* from a score for the final movement of Amy Beach's *Symphony in E minor, (the 'Gaelic')* as succinctly as:

```
[{'time_signature': '2/2'}, {'number': 563}]
```

Although this movement includes markings that users may *wish to* include in extended MMs—such as changes of tempo (e.g., *poco più mosso*), rehearsal marks, and the striking use of *ritmo di tre battute*—none of this is *essential for alignment.* As there are no split bars, repeats, etc., this very reduced form can be expanded to the complete representation of all 563 bars, fully described with the specified properties, by applying the specifications defaults:

count: Numbers 1 through 563.
ID: String conversion of **count**.
qstamp: Cumulative sum of **actual_length**s.
actual_length: Defaults to **nominal_length**.
nominal_length: Defaults to **time_signature** converted to quarter length (here 4),
time_signature: Defaults to the previous entry's value.
number: Defaults to **count** because **actual_length** equals **nominal_length** throughout.
name: String conversion of **number**.
start_/end_repeat: Default to `false`.
next: Single-element arrays containing the respective subsequent **count**, except for the final entry.

The compressed form this saves considerable disc space compared with the expanded/verbose equivalent. Moreover, as well as being easy to create mantually, the compressed MM offers a benefit for human-readability: it serves as a succinct 'warning system' by revealing potential alignment issues at a glance. Specifically, any

MM entry beyond the minimal start/end pair indicates moments of interest. We move now from this identification of *potential* issues in *one* source to the diagnosis of *actual* discrepancies *between* two or more versions of the same material.

4.2 Formats, conversion, diagnosis, resolution

The MM code base provides functionality for:

- producing MMs directly from everything parseable by music21 (MusicXML, krn, …) and the ms3 parser (mscx, mscz);
- validating MMs against a formal schema;
- integrating MMs into the Dezrann workflow;
- compression / expansion of MMs (as described in §4.1);
- diagnosis of differences between two MMs, yielding human- and machine-readable instructions for adjusting one to fit.
- implementation of the diagnosis instructions to make in-place changes on one score to align and fit with the another (currently music21-application only).

The diagnosis process involves numerous checks, from diverging bar numbers in otherwise congruent MMs, to MMs with mismatched numbers of entries. These divergences are typically simple enough when they appear alone; complexity arises where two or more co-occur in the same work. The appropriate solution then depends on inferring the causes for the discrepancies and resolving them in the appropriate order. The basic adjustment operations that our algorithm (currently) suggests are:

- **re-number**, to fit one or more of the standards given above (this can also run separately, e.g, before any comparison);
- **adopt**, e.g., a repeat mark or the `actual_length` of an anacrusis, mapping from one MM to the other;
- **split/merge** objects so that the *actual_length*s match up;
- **insert/remove** material to match the preferred source, e.g., repeated bars.

If no plausible best guess can be deduced, this is a strong indicator of a serious mismatch *in the music*. That being said, the algorithm is subject to ongoing parametrisation and we encourage critical feedback and contributions from the community.

4.3 A corpus demonstration

In demonstrating the new format and its usefulness we provide curated measure maps for several corpora, including:

The Annotated Mozart Sonatas [19], a dataset that calls for alignment and cross-evaluation with alternative annotations e.g., [1] and additional types of analysis (such as the texture annotations provided in [5]).

A subset of the OpenScore Lieder corpus [15] consisting of c.250 songs (from 1,300 total) by female composers (Hensel, Schumann, …) and with harmonic analyses from 'When in Rome' [14].

371 Bach chorales A dataset influential enough to exist in multiple formats and versions. As a test case, we identified and converted three sets of scores, and one set of analyses.

We demonstrate the utility of MMs first by using them to align and include the Mozart and Lieder corpora on Dezrann (http://dezrann.net/corpora) complete with scores and annotations (harmonic, textural, and structural) as available from the sources.

	K^0	K^1	K^2	M^1	M^2	C^2
perfect match	109	111	0	159	0	0
split/merge	124	124	124	127	127	129
re-number	0	124	124	6	6	115
adopt	126	2	111	73	232	112
mismatch	8	8	8	6	6	5
sum	**370**	**369**	**369**	**371**	**371**	**361**

Table 1: A comparison of MMs for the `.rntxt` chord analysis files against six sets of corresponding score encodings. *K*, *M*, and *C* stand for the original score formats (kern, MuseScore and Capella). Superscript numbers designate the original format (0), conversion to MusicXML (1), and conversion to MuseScore 4 (2).**

The Bach chorale settings have a long history in both traditional and computational musicology. We have aligned three datasets encoding these scores, each comprising all or most of the 371 chorales originally compiled by J.S. Bach's son, Carl Philipp Emmanuel. The encoded datasets were created independently in three different formats (**kern, MuseScore, Capella), using slightly different numbering systems, providing a usefully challenging test case.

Aligning the scores first involves non-trivial piece-level metadata comparisons (in this case using Riemenschneider's system) and serves multiple purposes. First, aligning different versions of the same corpus makes them interoperable and enables cross-validation and error detection. As discussed, certainty that the different score encoding files intend to describe the same music is a prerequisite for meaningful cross-corpus application, including MM comparison.

Second, establishing which files correspond to each other between the datasets provides us with a rich resource for testing and tuning our MM code base. Each of the three datasets are present in at least two different conversion formats, resulting in over 2,000 score files. Comparing the corresponding MMs gives considerable insight into what can go wrong when converting between formats (§3.2), and also highlights differences that arise from different print editions encoded (§2).

Third, the dataset lets us demonstrate one of the most frequent use cases, which is computationally evaluating a set of analytic labels against the scores they describe. For this example we use the harmonic analyses of all 371 chorales that were created by Dmitri Tymoczko and colleagues in RomanText files [35] and are included in the When-in-Rome meta-corpus [14]. These harmonic analysis files can be stored individually ("off-score") and parsed with `music21`. This has allowed us to create one MM per file, to be compared against those we have created for six different sets of scores.

1 shows preliminary results of this comparison, showing which of the operations listed in Section §4.2 may be needed to adjust each of scores to the analysis files (or vice-versa). While the MMs pertaining to scores originally encoded in **kern or MuseScore match their corresponding analysis files perfectly in roughly 30–40% of cases, after converting the three datasets to MuseScore, *none* of the 1,101 MMs match perfectly. However, the table suggests that the vast majority of sources may be adjusted, even automatically,

to fit (the few exceptions are in the row labelled "mismatch"). That leaves us with a very small fraction of scores to be opened and checked manually in order to determine and potentialy correct the source of the discrepancy.

5 OUTLOOK

As we have emphasised throughout, the issues at stake here are complex and it is almost certainly impossible for a single deterministic system to catch all discrepancies, in any combination. In that spirit, we close with a few examples of approximately where the boundary currently falls, alongside wider comments on the outlook and prospects for future adoption and development.

5.1 Bar variation between parts

Some music features different barring (and thus deliberately different MMs) between the constituent parts. This is reasonably common both before and after the common practice.[20] It also occurs (occasionally) within the common practice, most famously in the dance scene at the end of the first act of Mozart's *Don Giovanni* which is simultaneously 'in' 3/4, 2/4, and 3/8.

Again, we note the distinction between advice for *visual presentation*, and what is needed 'under the hood', in *logical representation*. Elaine Gould [16] advises that 'Bar numbers should not be used in music in which individual performers have different numbers of bars or where barlines do not coincide [...] Instead, use rehearsal marks at points where players coordinate.' This is advice for how the music should look when *printed* (or at least rendered). The *logical* encoding clearly does need some kind of bar numbering, probably with either hidden barlines or separate bar allocations for each part. The MM approach can *encode* either approach. While the assumption up to this point is that parts align and that any of those parts can be used to produce a MM that is representative of all, it is perfectly possible to produce separate MMs for each part. Diagnosis tools may identify this, but we do not prioritise or test for this use case.

5.2 Change of time signature mid-measure

Figure 4 shows an extract from Beethoven's *Piano Sonata*, Op.109 in which there is a change of time signature mid-bar. Most notation editors cannot support this within-syntax. Note how this relates to but differs from cadenzas which can often be handled with a distinction between nominal and actual duration.

In this case, MM functionality would need a barline for the change, producing a new 2/4 bar that happens to be complete (though that is not required). The barline can then be hidden (when it comes to the print-visual implementation) and the numbering re-assigned.

5.3 Nested repeats-within-repeats

While honouring the editorial traditions we have inherited and being sensitive to reasonable variety, it is as well to note that not everything has to be perpetuated: some more marginal ideas may not have become widespread for a reason.

[20] Early examples include the isorhythmic motets of the 14th century; post-common practice cases are common, and earlier than one might think: see, for example, the 4/2-against-3/4 in the 2nd movement of Ravel's piano Trio, and many works by Ives.

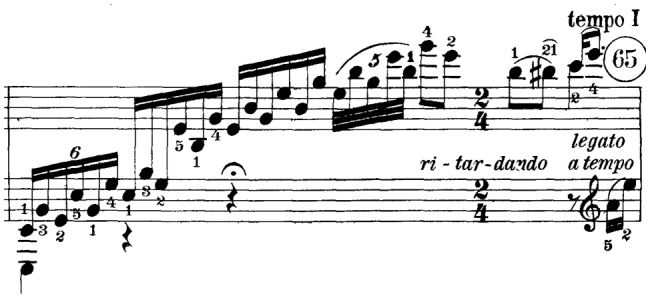

Figure 4: A mid-bar change of time signature ('3/4' to '2/4') in Beethoven's Op.109 piano sonata.

For instance, in some contexts there exist nested repeats-within-repeats. Once again, this practice is more common in 20th-century music (notably in Broadway and Minimalist scores), but can also be found earlier. Here the finale (movement 6) of Beethoven's Quartet No.13 (Op.130) provides a neat example. The Artaria (first edition) and Breitkopf und Härtel use large and small repeat marks; the Schott parts use repeats that are indistinguishable from one another; the Universal and Eulenberg use a *dal segno* alternative (D.C.D.S.); and the Peters edition parts write out the small repeat section so there is only the large first and second ending.

Clearly this practice exists, but just as clearly it is problematic. We do not categorically exclude this practice outright, but we do think that certain conditions must be met to consider supporting it. Most basically, there needs to be some systematic distinction between the outer and the inner repeats. A tag like that which distinguishes different types of barline would suffice, but no standard score format currently includes such a provision, and neither do we.

5.4 Recommendations for good practice

We conclude with suggestions for good practice in the creation and updating of corpora. For a **primary** repository we recommend including MMs for all files (e.g., accompanying every distinct score file). Alternatively, include an easy to use script for generating those MMs with the corpus-specific file paths.

For a **secondary** repository (e.g., analyses referring directly to a pre-existing score collection) there are several workable approaches. These include: full MMs and difference diagnosis files throughout or only where there are *differences* to report. Most important as always, is clear documentation for which of these approaches has been taken. If in doubt we recommend the 'more is more' approach of full MMs and diagnosis files.

Finally, while we provide for numerous existing formats (as discussed above), there is of course a limit. For corpora in any *encoded format* that is not currently supported, we respectfully recommend writing a converter. For *non-encoded* formats (e.g., physical copies of manuscripts) we encourage manual MMs. As strange as it may seem for all this focus on digital libraries, the MMs for most works are eminently simple enough to produce by hand (as demonstrated by the above case for Amy Beach). And interoperability is arguably most successful when it unites not only digital environments, but also the real world.

ACKNOWLEDGMENTS

We thank all contributors to this wide-ranging project. Initial work on this project (authors Gotham and Dykeaylen) was conducted in the context of Gotham's Affiliated Lectureship at the Department of Computer Science and Technology, University of Cambridge. We thank Alan Blackwell and others for their role in organising and supporting this. Like all users of Dezrann, we thank Charles Ballester and especially the long-term developer Emmanuel ('Manu') Leguy. Authors Giraud and Couturier's work is supported by a French ANR CollabScore ANR-20-CE27-0014 grant and by Région Hauts-de-France. Authors Hentschel and Rohrmeier's research is supported by the Swiss National Science Foundation within the project "Distant Listening – The Development of Harmony over Three Centuries (1700–2000)" (Grant no. 182811) and is being conducted at the Latour Chair in Digital and Cognitive Musicology, generously funded by Mr. Claude Latour.

REFERENCES

[1] Jenine Brown, Daphne Tan, and Michelle Lin. [n. d.]. An Analytical Dataset of Approaches to V in Mozart. 16, 2 ([n. d.]), 341–350. https://doi.org/10.18061/emr.v16i2.8511

[2] Donald Byrd and Eric Isaacson. 2003. A music representation requirement specification for academia. *Computer Music Journal* 27, 4 (2003), 43–57.

[3] Carlos Eduardo Cancino-Chacón, Silvan David Peter, Emmanouil Karystinaios, Francesco Foscarin, Maarten Grachten, and Gerhard Widmer. 2022. Partitura: A Python Package for Symbolic Music Processing. In *Proceedings of the Music Encoding Conference (MEC2022)*. Halifax, Canada.

[4] Barry Cooper. 2017. Rehearsal Letters, Rhythmic Modes and Structural Issues in Beethoven's Grosse Fuge. *Nineteenth-Century Music Review* 14, 2 (2017), 177–193. https://doi.org/10.1017/S1479409816000069 Publisher: Cambridge University Press.

[5] Louis Couturier, Louis Bigo, and Florence Levé. 2022. A Dataset of Texture Annotations in Mozart Piano Sonatas. In *International Society for Music Information Retrieval Conference (ISMIR 2022)*.

[6] William H. Cummings. 1904. Bar-Lines. *The Musical Times* 45, 739 (1904), 574–575.

[7] Michael Scott Cuthbert. 2010. Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data. In *11th International Society for Music Information Retrieval Conference*, J. Stephen Downie and Remco C Veltkamp (Eds.). Utrecht, Netherlands, 637–642.

[8] Jacob DeGroot-Maggetti, Timothy R De Reuse, Laurent Feisthauer, Samuel Howes, Yaolong Ju, Suzuka Kokubu, Sylvain Margot, Néstor Nápoles López, and Finn Upham. 2020. Data Quality Matters: Iterative Corrections on a Corpus of Mendelssohn String Quartets and Implications for MIR Analysis. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*. Zenodo. https://doi.org/10.5281/ZENODO.4245459

[9] Francesco Foscarin, Emmanouil Karystinaios, Silvan David Peter, Carlos Cancino-Chacón, Maarten Grachten, and Gerhard Widmer. 2021. The Match File Format: Encoding Alignments between Scores and Performances. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*.

[10] Francesco Foscarin, Philippe Rigaux, and Virginie Thion. 2021. Data Quality Assessment in Digital Score Libraries: The GioQoso Project. *International Journal on Digital Libraries* 22, 2 (June 2021), 159–173. https://doi.org/10.1007/s00799-021-00299-7

[11] Wolf Frobenius. 2012. Tactus. In *Handwörterbuch der musikalischen Terminologie: Im Auftrag der Akademie der Wissenschaften und der Literatur, Mainz*, Hans Heinrich Eggebrecht, Albrecht Riethmüller, and Markus Bandur (Eds.). Franz Steiner Verlag, Stuttgart.

[12] Louis Garczynski, Mathieu Giraud, Emmanuel Leguy, and Philippe Rigaux. 2022. Modeling and Editing Cross-Modal Synchronization on a Label Web Canvas. In *Music Encoding Conference (MEC 2022)*. Halifax, Canada. https://hal.science/hal-03583179

[13] Michael Good. 2001. MusicXML for Notation and Analysis. In *The Virtual Score: Representation, Retrieval, Restoration*, Walter B. Hewlett and Eleanor Selfridge-Field (Eds.). Computing in Musicology, Vol. 12. The MIT Press, 113–124. https://doi.org/10.7551/mitpress/2058.001.0001

[14] Mark Gotham, Gianluca Micchi, Néstor Nápoles-López, and Malcolm Sailor. 2023. When in Rome: a meta-corpus of functional harmony. *Transactions of the International Society for Music Information Retrieval* (forthcoming 2023).

[15] Mark Robert Haigh Gotham and Peter Jonas. 2022. The OpenScore Lieder Corpus. In *Music Encoding Conference Proceedings 2021*, Stefan Münnich and David Rizo (Eds.). Humanities Commons, 131–136. https://doi.org/10.17613/1my2-dm23

[16] Elaine Gould. 2011. *Behind Bars: The Definitive Guide to Music Notation.* Faber Music, London.

[17] Andrew Hankinson, Perry Roland, and Ichiro Fujinaga. 2011. The Music Encoding Initiative as a Document-Encoding Framework. In *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR*, Anssi Klapuri and Colby Leider (Eds.). University of Miami, 293–298.

[18] Johannes Hentschel, Andrew McLeod, Yannis Rammos, and Martin Rohrmeier. 2023. Introducing DiMCAT for Processing and Analyzing Notated Music on a Very Large Scale. In *Proceedings of the 24th International Society for Music Information Retrieval Conference.*

[19] Johannes Hentschel, Markus Neuwirth, and Martin Rohrmeier. 2021. The Annotated Mozart Sonatas: Score, Harmony, and Cadence. *Transactions of the International Society for Music Information Retrieval* 4, 1 (2021), 67–80. https://doi.org/10.5334/tismir.63

[20] Johannes Hentschel and Martin Rohrmeier. [n. d.]. Ms3: A Parser for MuseScore Files, Serving as Data Factory for Annotated Music Corpora. 8, 88 ([n. d.]), 5195. https://doi.org/10.21105/joss.05195

[21] David Huron. 1988. Error Categories, Detection, and Reduction in a Musical Database. *Computers and the Humanities* 22, 4 (1988), 253–264. https://doi.org/10.1007/BF00118601

[22] David Huron. 1994. *The Humdrum Toolkit: Reference Manual.* Center for Computer Assisted Research in the Humanities, Menlo Park, CA.

[23] David Huron. 1998. *Music Research Using Humdrum. A User's Guide.*

[24] David Huron. 1999. *The New Empiricism: Systematic Musicology in a Postmodern Age.* Music and Mind: Foundations of Cognitive Musicology 3. The 1999 Ernest Bloch Lecture, University of California, Berkely.

[25] David Lewis, Elisabete Shibata, Mark Saccomano, Lisa Rosendahl, Johannes Kepper, Andrew Hankinson, Christine Siegert, and Kevin Page. 2022. A Model for Annotating Musical Versions and Arrangements across Multiple Documents and Media. In *Proceedings of the 9th International Conference on Digital Libraries for Musicology (DLfM '22)*. Association for Computing Machinery, New York, NY, USA, 10–18. https://doi.org/10.1145/3543882.3543891

[26] Edward E. Lowinsky. 1960. Early Scores in Manuscript. *Journal of the American Musicological Society* 13, 1/3 (1960), 126–173. https://doi.org/10.2307/830252 jstor:830252

[27] Han-Wen Nienhuys and Jan Nieuwenhuizen. 2003. LilyPond, a System for Automated Music Engraving. (2003).

[28] Polina Proutskova, Daniel Wolff, György Fazekas, Klaus Frieler, Frank Höger, Olga Velichkina, Gabriel Solis, Tillman Weyde, Martin Pfleiderer, Hèlène Camille Crayencour, Geoffroy Peeters, and Simon Dixon. 2022. The Jazz Ontology: A Semantic Model and Large-Scale RDF Repositories for Jazz. *Journal of Web Semantics* 74 (Oct. 2022), 100735. https://doi.org/10.1016/j.websem.2022.100735

[29] Laurent Pugin, Rodolfo Zitellini, and Perry Roland. 2014. Verovio: A Library for Engraving MEI Music Notation into SVG. In *Pugin, Laurent; Zitellini, Rodolfo; Roland, Perry (2014). Verovio: A Library for Engraving MEI Music Notation into SVG. In: 15th International Society for Music Information Retrieval Conference (ISMIR 2014). Taipei. 27-31 Oktober 2014.* Taipei, 107–112.

[30] Perry Roland. 2002. The Music Encoding Initiative (MEI). In *Proceedings of the International Conference on Musical Applications Using XML.* 55–59.

[31] Eleanor Selfridge-Field. 1997. Introduction. Describing Musical Information. In *Beyond MIDI: The Handbook of Musical Codes*, Eleanor Selfridge-Field (Ed.). MIT Press, Cambridge, Mass, 3–38.

[32] Daniel Shanahan, John Ashley Burgoyne, and Ian Quinn (Eds.). 2022. *The Oxford Handbook of Music and Corpus Studies* (first ed.). Oxford University Press. https://doi.org/10.1093/oxfordhb/9780190945442.001.0001

[33] Leland Smith. 1997. SCORE. In *Beyond MIDI: The Handbook of Musical Codes*, Eleanor Selfridge-Field (Ed.). MIT Press, Cambridge, Mass, 252–280.

[34] Kurt Stone. 1980. *Music Notation in the Twentieth Century: A Practical Guidebook* (1st ed ed.). W. W. Norton, New York.

[35] Dmitri Tymoczko, Mark Gotham, Michael Scott Cuthbert, and Christopher Ariza. 2019. The RomanText Format: A Flexible and Standard Method for Representing Roman Numeral Analyses. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, Arthur Flexer, Geoffroy Peeters, Julián Urbano, and Anja Volk (Eds.). 123–129. http://archives.ismir.net/ismir2019/paper/000012.pdf

[36] David M. Weigl, Werner Goebl, Alex Hofmann, Tim Crawford, Federico Zubani, Cynthia C. S. Liem, and Alastair Porter. 2020. Read/Write Digital Libraries for Musicology. In *7th International Conference on Digital Libraries for Musicology (DLfM 2020)*. Association for Computing Machinery, New York, NY, USA, 48–52. https://doi.org/10.1145/3424911.3425519

[37] Benjamin B. M. Yu. 1984. Mental Study in Piano Teaching. 33, 6 (1984), 12–14. https://www.proquest.com/docview/1294820902/citation/CD9AE0C5FFF843B7PQ/1