TAKING FORM: A REPRESENTATION STANDARD, CONVERSION CODE, AND EXAMPLE CORPUS FOR RECORDING, VISUALIZING, AND STUDYING ANALYSES OF MUSICAL FORM

Mark Gotham
Cornell University
Department of Music

Matthew T. Ireland
University of Cambridge
Sidney Sussex College

ABSTRACT

We report on new specification standards for representing human analyses of musical form which enable musicians to represent their analytical view of a piece either on the score (where an encoded version is available) or on a spreadsheet. Both of these representations are simple, intuitive, and highly human-readable. Further, we provide code for converting between these formats, as well as a nested bracket representation adopted from computational linguistics which, in turn, can be visualised in familiar tree diagrams to provide 'at a glance' introductions to works. Finally, we provide an initial corpus of analyses/annotations in these formats, report on the practicalities of amassing them, and offer tools for automatic comparison of the works in the corpus based on the content and structure of the annotations. We intend for this resource to be useful to computational musicologists, enabling study of form at scale, and also useful pedagogically to all teachers, students, and appreciators of music from whom projects of this kind can be rather disconnected. The code and corpus can be found at `https://github.com/MarkGotham/Taking-Form`

1. INTRODUCTION

Marking up scores is a fundamental part of life for any musician working in a score-based medium including – but not entirely limited to – Western classical music. The specifics vary widely from analytical observations to instrument fingerings, but the wider goals can be understood in the same terms: commenting on the music to clarify, visualise, remind, and reinforce a particular understanding of the music in question or plan for its execution.

Regrettably, these annotations are rarely stored, kept, and shared effectively. For instance, an ensemble may assiduously mark up a set of parts, but at the end of the hire period, publishers require them to erase those annotations. (Admittedly this is routinely flouted.) Some conductors travel with their own sets of parts for this reason; others waste rehearsal time conveying markings that could well have been in the score in the first place.

Those score markings represent one of many forms of siloed knowledge that we cannot easily share and repurpose. The same is true of most musical analysis which is published in articles and monographs which time-pressed musicians for the most part simply do not read, and from which the information is hard to extract automatically.

The digital age presents potential solutions to many of these problems, and computational musicology is beginning to answer the call with solutions that centre on representing musico-analytical information in a structured fashion. For instance, formats designed to enable the representation, study, and re-purposing of Roman-numeral analyses include the 'TAVERN', 'ABC', and 'RomanText' projects among others [5, 13, 19]; Automated extraction of performance data include [6, 7], and 'Tuttitempi'; [1] and even pencil-on-paper performers' markings have become an extricable dataset [1].

A key part of the equation here is the opportunity for musical visualization; prior work in this area sees a close relationship between efforts to create format standards for representing musical structure on the one hand, and visualising those structures on the other. This has included handling note-by-note annotations ab initio (such as Dezrann [8] and Verovio [14]) as well as a range of ad hoc visualizations for other research projects like the 'Ribbon' formal analysis of the Josquin Research Project [16]. [2]

This expands the more established practice of visualizing audio analysis in which Fourier or wavelet transforms visualise spectral content (routine in commercial softwares); beat trackers visualise the micro-timing of performed tempo; [3] and even the recorded waveform itself visualises the dynamic profile (an easily forgotten, but eminently useful resource). [4]

2. FORM

Form is a particularly pertinent candidate for stronger representation. The range of use cases includes computational study and visualizations for education in the widest sense:

Attribution: Mark Gotham, Matthew T. Ireland. "Taking Form: A Representation Standard, Conversion Code, and Example Corpus for Recording, Visualizing, and Studying Analyses of Musical Form", 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 2019.

[1] `https://tuttitempi.com/`
[2] `http://josquin.stanford.edu/`
[3] For instance, see Sonic Visualiser [2]
[4] See [4] for a summary of other relatively recent audio visualizations.

not only of those formally involved in traditional modes of teaching and learning, but anyone seeking to attain or maintain a stronger grasp of the work in question.

Discussion of form necessarily involves comparisons across the full range of the work in question. Keeping all of this information in mind is a challenge and thus form is a parameter particularly in need of schematic representations: while a harmonic or melodic device can often be reproduced exactly, that is almost never possible with form. This has led to a range of creative representations of large-scale structures with clear, at-a-glance summaries.

This paper reports on an attempt to represent formal analyses such that they can be stored, visualised, and studied effectively. We accommodate any number of internal divisions, and envisage this being used to mark up analytical views of everything not specified in the score: that is divisions right down to the level of hypermetrical / phrase groupings. In principle, this could be continued further, to represent the structure of the measure level and beat groupings. [5] That may well be useful for metrically complex works; we include one example in the 'Miscellany' corpus (see Section 3) but do not explore it further here as those details are usually clear from the notation in the common practice repertoire we are primarily targetting. Instead we focus on representing analytical views of those un-notated, hypermetrical and formal levels above the single measure.

We propose three similarly expressive standards for representing analyses, and we provide code for converting between those representations. *On-score mark-up* (Section 2.2) forms the most natural user interface for musicians accustomed to making such annotations; *Abstract syntax trees* (Section 2.1) present a final visualization format; and the *tabular standard* (Section 2.3) is a flexible intermediary encoding useful for overcoming some of the complexities inherent in annotating a score with labels that denote position in a hierarchical structure. We now begin at the 'extremes', as it were, with the (final) visualization format and (initial) on-score mark-up, before proceeding to 'join the dots', discussing the intermediary standard.

2.1 Abstract Syntax Trees (ASTs)

In dealing with form, we face the specific problem of representing hierarchical information, and thus we respond to an explicit call from [15] for a 'Standard Format' for 'Hierarchical Analyses and Representations', including form. Previous work includes Craig Sapp's automation and visualization of 'Hierarchical Key Analysis' [18], and the 'Variations Audio Timeliner' [6] approaches form specifically (though it not open source). There is also a well established precedent for using 'tree' structures (borrowed from computational linguistics) to represent this hierarchical information in music.

Perhaps the most famous linguistics-music cross-over, 'A Generative Theory of Tonal Music' [11], applied a

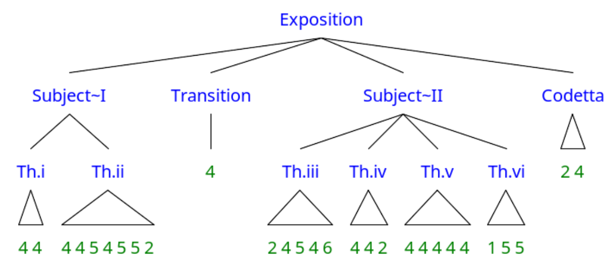

Figure 1. Example tree visualisation: the exposition from the first movement of Schubert's Symphony No.8. The abbreviations are 'Subject' for 'Subject Group', and 'Th.' for 'Theme'.

version of ASTs to discuss metrical and grouping well-formedness and preference at the note level in Schenkerian terms. [17] has applied the same logic to the different 'levels' of harmonic identity, from the chord symbol (such as 'C') through Roman-numerals ('I') to Riemannian functions ('T'). [7]

Figure 1 provides an example of such an AST representation of a passage of musical form: the exposition from the first movement of Schubert's Symphony No.8. The figure includes structural groupings from the highest level down to the level of measure grouping: the numbers of the lowest level refer to the number of measures in each successive phrase grouping.

This paper provides a specification format for producing such visualisation on the basis of well-structured representations of form and phrase in music, as well as code for converting between this representation format, various kind of tree visualizations, and, crucially, encoded scores. With these tools, we can mark observations directly onto scores (which is pedagogically invaluable) and keep them in a structured format for representations and analysis.

In the specification format proposed, each mark-up annotation (labelled on a score) maps directly to a node in the AST. The level (distance from the root node) in the AST must be explicitly encoded within the annotation.

2.2 On-score mark-up

There are many ways to encode this information. This paper reports on a single standard for doing so which is highly flexible both in terms of the types and terms of divisions used, and also the input method. We begin with perhaps the most user-friendly method: marking formal divisions directly onto encoded scores. All that is required is a music notation software which will export to the interoperable standard musicXML. Most notation software support this, including both commercial and free/open-source options like MuseScore.

To enter a marking, simply create a 'stave text' as described in the following instructions which combine the

[5] Our computational framework could be extended to calculate and visualise any arbitrary summary statistic, measuring any lowest level divisions.

[6] variations.sourceforge.net/vat/

[7] For more on datasets for and visualisations of GTTM and Schenkerian reductions see [10, 12] and the work of Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo, including [9] and Masatoshi Hamanaka's XML markups of musical examples from (and using the tree structure representation of) GTTM [11] `http://gttm.jp/gttm/`

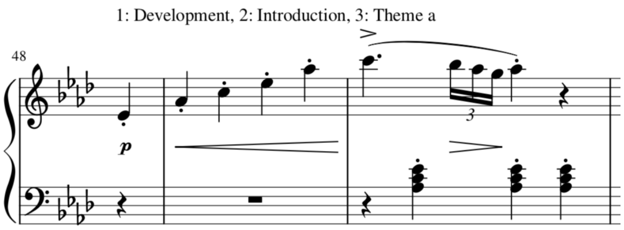

Figure 2. Example file markup in the first movement of Beethoven's Piano Sonata op.2 no.1.

definition of the minimal well-formedness constraints with 'how to' style instructions. Figure 2 illustrates what this looks like on a score.

- First, click on any note at the right measure and beat position for the marking you want to enter.
- Enter a new textual marking at that position (most software provide CMD+T (Mac) / CNTRL+T (Windows) as a shortcut for this).
- Begin each on-score mark-up with an initial character to specify a 'level number' for the marking in question, followed by a colon. For instance, 'Exposition', 'Development' and 'Recapitulation' will all be on the top level of division (number 1) in sonata form movements, so '1: Exposition'. All other levels continue the divisions from here, (e.g. '2: First Subject Group').
- Where you wish to indicate a division, but have no name for the span in question, use the correct level number and a placeholder text like '4: X'.
- In practice, many entries like '1: Exposition' and '2: First Subject Group' will begin at the same time. To indicate these multiple, simultaneous level entries, insert one text entry with all the component parts divided by a comma (','). For instance, many sonatas will begin with the long string: '1: Exposition, 2: First Subject Group, 3: Theme a, 4: Sentence, 5: Presentation, 6: Basic Idea'.

All entries are relative to each other, so level numbers are given by finding the directly relevant parallel. These will fall into a few basic types of comparison:

- A span and its first phrase-division will generally be used together, with the division at +1 level. For a phrase, we might have '4: Period, 5: Antecedent'.
- The next division of that phrase ('Consequent') is at the same level of the first division ('Antecedent'), thus '5: Consequent'.
- The next phrase outside of this span returns to the initial level, so '4: Period'.
- When we eventually get to larger structural boundary then we have to find an entry at a comparable level, which will involve proportionately wider view. For the 'Recapitulation' we're starting again from the top level (so '1: Recapitulation'), while for the second subject group it's level 2 ('2: Second Subject Group').

Note that while the numbers assigned to the top levels will be consistent within and even across pieces ('Exposition', 'Development' and 'Recapitulation' will almost always appear as a set at level 1), those at lower levels will vary depending on the number of level divisions above them. For instance, the same phrase grouping structure may appear at multiple levels across a piece: at lower levels in a long and richly-structured exposition, but at nominally 'higher' levels in a short Coda with fewer divisions (see Figure 1).

This system has been designed to minimise the potential for confusion. Among the alternatives we considered and rejected was a system of multiple comparative marks with '+' for breaking up the existing span, '-' backing up one level, '=' for the same level and '*' for reverting to the highest level. This may appear to be an improvement until you try to indicate the start of a development, for instance, and have to work out the number of successive '-'s needed to reach the right level.

Further work could explore the possibility of encoding similar annotations within a formal context-free grammar. In that case, the user would specify the start of the scope of each annotation, allowing a top-down parser to infer the AST structure without the user having to explicitly specify the depth of each annotation. Informal user trials suggested this recursive mark-up format to be less intuitive and less flexible than the format proposed above, and so we have not explored it further in this initial work.

In any case, the process is necessarily somewhat complicated. To keep everything in order, some users may prefer to work directly on spreadsheets (for which instructions follow in the next section) or at least to keep an informal spreadsheet on the go at the same time.

2.3 Tabular standard

See `https://www.github.com/MarkGotham/Taking-Form` for code to extract these score markings and set them out in a tabular format that is also (perhaps even more) human-readable. Equally, some users will prefer to work directly on spreadsheets. In our experience, the tabular format provides a faster and more flexible user interface for adding, deleting, and especially for modifying annotations.

In the tabular standard:

- Generally, there is one row per measure. Exceptions include the relatively common case of 1st / 2nd time bars (for which use measure 'numbers' 112a / 112b etc.) and the extremely rare case where two structural boundaries fall within the same measure (use multiple rows).
- Each text entry indicates the start of a span. Again, we suggest using text like 'Exposition' where a

```
Sentence     Presentation    Basic Idea
                             Basic Idea
             Continuation    Fragmentation
                             Cadence
```

Figure 3. An example of the tabular representation format, setting out a generic division of a sentence (level N) into presentation and continuation phases (level N+1) which, in turn, divide into two 'Basic Idea's in the first case, and 'Fragmentation' and 'Cadence' in the latter (all at level N+2).

name is appropriate, and 'X' for any moment that begins a new span, but is not so easily labelled.

- That marking entry remains in effect until a change in the same column (e.g. replacing 'Exposition' with 'Development' or one 'Period' with another 'Period', for instance).
- The columns go from larger to smaller units and users may employ as many columns as needed to get from the large formal architecture (e.g. expositions of c.50 bars) down to few-bar groupings. We make no assumptions here. While a '4' grouping will often divide into 2+2, it might be better expressed as an undivided 4 (and occasionally as an asymmetric 3+1). Assuming that '4' indicates 2+2 would leave us without a description for the undivided 4.
- Again, the final levels of grouping will appear in different columns of the document, depending on the number of intervening levels.
- The 'beat' column is available for registering exactly where a new theme starts, accommodating anacruses for instance. When inputting directly to tabular, the user may wish to indicate beats with negative numbers to indicate both the basic measure range and the exact beat position at a glance. If in doubt, we advise using the downbeat ('beat 1'), especially if the accompaniment part begins there (the accompaniment is a part of the span too, after all). The code for converting to bracket representation involves a simplification that eliminates beat information.

The representation standard (in both the on-score and tabular entry systems) also accepts 'equal division', which may be useful for sections which need two labels for exactly the same span. For instance, a section might be listed as 'A' but also consist entirely of a 'Compound Period'. Similarly, the 'First theme' may very well be a 'Sentence'. Assign these designations consecutive level numbers (columns) even though they represent the same span.

Finally, while the standard accommodates any naming system, we recommend naming themes successively, and not restarting the count for the second subject group (continuing instead with 'Theme C', for instance) in order to enable succinct and unambiguous reference back to these themes later on.

```
[Exposition
    [Subject~I
        [Th.i 4 4]
        [Th.ii 4 4 5 4 5 5 2]
    ]
    [Transition 4]
    [Subject~II
        [Th.iii 2 4 5 4 6]
        [Th.iv 4 4 2]
        [Th.v 4 4 4 4 4]
        [Th.vi 1 5 5]
    ]
    [Codetta 2 4]
]
```

Figure 4. An example of the tabular representation format, setting out form, theme, and phrase groupings in the exposition of Schubert 8/i (corresponding to Figure 1).

2.4 Nested Bracket Notation

The linguistics standard for representing syntax trees is a system of nested brackets. Figure 4 sets out an example of this applied to musical form. In this format, the first entry in a bracket is the parent node, labelling the overall span, and each subsequent entry is a child node (an internal division). These child nodes may divide further with additional brackets recursively.

Our specification requires only:

- Matched brackets: each open '[' must be paired with a closing ']'.

There are no further, formal constraints.

Our converter outputs this as a simple string, which is all that's required to meet the syntax criteria for the various existing code libraries for visualising nested bracket notation with trees. However, once again, some users may wish to begin directly at this level, in which case we would advise one additional constraint for readability:

- Add a new level of indentation to the bracket file to indicate the structural level in question.

Figure 4 reflects this more visually graspable version.

3. CORPUS

We complement this new standard and code base with a corpus of examples. The preparation of that corpus warrants comment as a potential model for other projects of this kind.

The analyses began in the classroom, with students each assigned their own movement(s) to work on. The instructors then marked those student analyses, providing a 'corrected' version based on the student work as well as written feedback, drawing their attention to important moments and considerations. This process thus created a corpus of student-initiated, instructor-corrected analyses which went through a final round of finessing and proof-reading to ensure grammatical accuracy and also consistency before release.

Future projects could extend this to a final round explicitly consolidating one scholar's analyses for additional internal consistency and quality control. In any case, the process benefits from involving a large number of people in suitable capacities and in a time-effective manner. Students get high-quality feedback from highly invested researcher-teachers, as well as direct integration into research projects. Researcher-teachers, in turn, benefit from knowing that their marking time is more than doubly valuable: they are providing great feedback and building a corpus at the same time.

The corpus combines tabular and nested-bracket representations to illustrate the range outlined above. The tabular representations consist of:

- Beethoven's 'first period' piano sonatas (nos1–15): 15 sonatas, 54 movements. [8]
- The first movements of all the Mozart piano sonatas (15 sonata movements).

The nested-bracket format demonstrates that format alternative, as well as other formal representations of the music. Specifically, while the tabular representations represent measures by number, these bracket representations register the number of measures within the spans in question directly (the conversion code accommodates both).

We are keen to stress that the standard supports any analyses and terminologies consistent with the minimal formal constraints itemised above. As discussed, we have elected in this corpus to name the themes alphabetically here to minimise confusion with the numbered levels, but again, this format supports any naming system the user cares to apply or devise. We provide a more miscellaneous set of repertoire and formal ideas in the corpus' 'Miscellany' folder to this effect. For instance, the analysis of the first movement from Bach's Brandenburg Concerto No.6 uses this template to set out the form in terms of changes to the canon distances.

Yet more importantly, we emphatically do not intend the analyses themselves to be in any way definitive, but rather a proof of concept, and a first offering. We welcome multiple analyses of the same work to focus our disciplinary discussion of what gives rise to formal labels and to the shifting balance of priorities. That said, creating more analyses of different works is probably a keener priority at the outset of this field. [9]

4. APPLICATIONS

The existence of tree structure representations based on formal analyse enables a number of applications in the visualisation of musical form and the comparison of form between scores. Both the visualization and the automated comparison help to focus questions of divergence between two analyses.

In this project, we have primarily focused on off-score visualizations to enable at-a-glance overviews and the development of global intuitions for structure. Equally, these formal observations could be returned to scores for different kinds of visual clarification. For instance, they enable the automated marking-up of phrase-ending barlines more strongly in scores, which is a popular annotation type among conductors. Moreover, we can toggle these annotations on and off at will: this is an important, and fundamentally digital-age flexibility.

Turning to automated comparisons, a distance metric between trees can be defined, which in turn enables clustering of scores with similar form. That process is discussed in the following section.

4.1 Tree edit distance and clustering

Zhang and Shasha [20] provide an algorithm for calculating the edit distance between two trees, t_1 and t_2. This metric is based on the minimum number of *insert*, *delete* and *update* operations (each of which have an associated cost for a given node) required to transform t_1 into t_2. We adopt their approach but allow the user to specify a function to describe the cost of each operation for a particular node, to enable the distance metric to be tuned for a particular corpus. In the examples that follow, the cost of inserting, deleting or updating a node is weighted more highly for nodes closer to the root of the tree. The cost function should also use the semantics of each node: the default comparator function in the code provided weights operations on a placeholder node '[X]' lower than the equivalent operation on any other node. Thus inserting a placeholder node '[X]' at level 6 in the tree is less costly than inserting a 'Coda' at level 1 in the tree, for example.

The concept of tree edit distance has been applied to a range of problems such as automated melody recognition, plagiarism detection in software, and finding similarities between sequences of RNA and DNA. We have explored two applications of tree edit distance, involving comparisons between two analyses, and between N analyses. We set these out in turn.

Between two analyses. A summary of the differences in tree structure up to some depth ($l_{\max}$) can be collapsed into strings in a simple language. We base our notation on a language proposed by Cunningham for summarising control structures in programming languages. [10] Figure 5 illustrates how this simplified version of nested-bracket notation (without labels) can be used to visualise differences in form up to a given depth ($l_{\max}$) in the tree.

In our simplified language: '||' denotes the divisions between nodes at level 1 in the tree; '.' represents a structure at level 5, or $l_{\max}$ if $l_{\max} < 5$; '{' and '}' denote the start and end (respectively) of the span of a node at level 2 (if level $2 < l_{\max}$); '`[`' and '`]`' denote the start and end (respectively) of the span of a node at level 3 (if

[8] Of these 15 sonatas, 9 are in 4 movements, 6 are in 3.

[9] [3]'s BPS-FH dataset of first movements from Beethoven Piano Sonatas. BPS-FH focusses on harmony but also includes some formal designations, and thus a balance between the potential benefits of cross-referencing and of extending the collective corpus.

[10] `http://c2.com/doc/SignatureSurvey/`

ANALYSIS 1 {..}{...}{...}|| {..}{....}{..}
ANALYSIS 2 {..}{.. .}{...}||'{..}{....}{..}

Figure 5. Two analyses encoded as strings in a simple language. This enables edit distance-based comparison of their musical form up to level 3 in the abstract syntax tree.

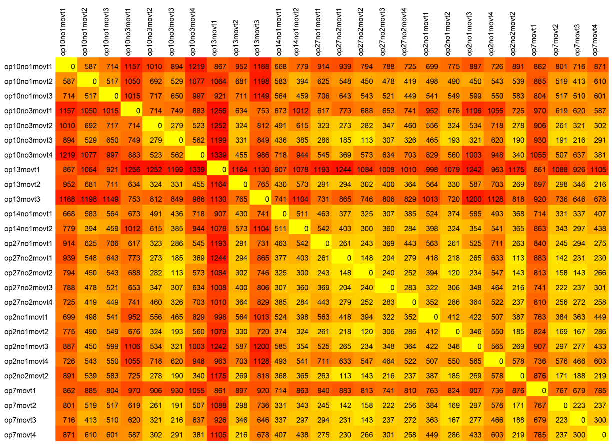

Figure 6. Heat map of tree edit distance between a selection of works in the corpus provided. The colour scale ranges from yellow (most similar/lowest tree edit distance) to red (most different/highest tree edit distance).

level 3 $< l_{\max}$); '(' and ')' denote the start and end (respectively) of the span of a node at level 4 (if level 4 $< l_{\max}$); ' denotes an anacrusis or longer introductory passage.

The process of calculating the tree edit distance allows correct alignment of the two strings. Our current analysis code highlights in green those nodes that are present in one analysis but not the other, and in red those nodes for which differences exist at a level greater than $l_{\max}$.

Between N analyses. Expanding this principle further, tree edit distance can be used to compare the form of all N scores in a corpus. Pairwise similarities in form can thus be identified. Figure 6 shows the tree edit distance between a selection of works in the corpus provided, visualised as a heat map.

5. SUMMARY AND OUTLOOK

This paper reports on a new specification standard for representing musical form, a corpus of example movements, and a set of code for converting between a range of representation and visualisation formats, handling score annotation to tabular or nested bracket representation, and tabular to nested bracket representation.

Among the possible improvements to this model, we envisage developments to both the user-interface and the representation structure. We consider the current opportunity to work on free and open source notation software to be a radical improvement to the student experience, enabling them to work directly on 'the music', listening back to it at will, and recording their observations in a digital format. That said, we are still limited to the availability of encoded scores and this is one musical area in which working with PDFs would be helpful, and could be practical.

As projects such as PeachNote exemplify, while OMR cannot yet reliably generate performance-ready scores, it is generally robust enough to support use cases like score-to-audio matching. Thus, as long as we still have many more PDFs than encoded scores available, it would be worth considering an alternative user-interface which offers a drag-and-drop system, for placing annotation labels onto the score. Here, we only need OMR robust enough to extract measure lines accurately and pair them with the annotation. Using pre-made (rather than free-text) labels would also solve the issue of ambiguous user-entries, though it would also limit the range of possible answers.

Turning to the representation standard, one priority for improvement is the inclusion of ambiguity. At present, we cannot admit multiple variant readings in a single file (though nothing stops analysts from producing multiple variant files). Formal judgments can arise through identification of melodic and / or harmonic closure, textural and / or dynamic changes, and much more besides. When forced into a single reading, we have to make difficult decisions between these competing priorities.

Furthermore, this ambiguity motivates a second direction for computational research based on this kind of corpus: while we can clearly pursue questions exploring proportions and the like, taking the analyses at face value, we can also explore which score elements correlate most strongly with formal designations, where analyses (dis)agree, and what gives rise to these shifts in priority.

This paper began with an overview of some recently implemented representation and visualization formats for other musical parameters; a key frontier in this field will be the combination of those standards. Given the lack of a consensus over a primary successor to XML for score encodings (MNX, MEI, ...), this much more sparsely populated field of analytical representations may be doubly slow in determining its future direction. That said, representations of harmony and form by reference to their measure and beat positions are easy to combine, perhaps representing harmonies as leaf nodes (assuming they change at least as frequently as formal sections).

Finally, we also motivated this research with its potential breadth of appeal to musicians outside the scholarly, computational community. This should be a key consideration for determining future research priorities. Integrating diverse groups in the production of corpora (also featured in this text and project) may be one way to road test new ideas for ease and popularity of use at the outset. We invite other researchers building corpora to experiment with variants on this process and whatever the exact standards they use, to focus on building up a large and versatile meta-corpus for the benefit of all.

6. ACKNOWLEDGEMENTS

This research was supported by a grant from Cornell University's Active Learning Initiative (ALI). We wish to thank the ALI, and all Cornell students who took part.

7. REFERENCES

[1] Eamonn Bell and Laurent Pugin. Approaches to handwritten conductor annotation extraction in musical scores. In *Proceedings of the 3rd International Workshop on Digital Libraries for Musicology*, DLfM 2016, pages 33–36, New York, NY, USA, 2016. ACM.

[2] C. Cannam, C. Landone, and M. Sandler. Sonic Visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the ACM Multimedia 2010 International Conference*, pages 1467–1468, Firenze, Italy, October 2010.

[3] Tsung-Ping Chen and Li Su. Functional harmony recognition of symbolic music data with multi-task recurrent neural networks. In Emilia Gómez, Xiao Hu, Eric Humphrey, and Emmanouil Benetos, editors, *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 90–97, 2018.

[4] Nicholas Cook. Methods for analysing recordings. In Nicholas Cook, Eric Clarke, Daniel Leech-Wilkinson, and John Rink, editors, *The Cambridge Companion to Recorded Music*, Cambridge Companions to Music, pages 221–245. Cambridge University Press, 2009.

[5] Johanna Devaney, Claire Arthur, Nathaniel Condit-Schultz, and Kirsten Nisula. Theme and Variation Encodings with Roman Numerals (TAVERN): A new data set for symbolic music analysis. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, pages 728–734, 2015.

[6] Johanna Devaney and Hubert Léveillé Gauvin. Representing and linking music performance data with score information. In *Proceedings of the 3rd International workshop on Digital Libraries for Musicology, DLfM 2016, New York, USA, August 12, 2016*, pages 1–8, 2016.

[7] Johanna Devaney and Hubert Léveillé Gauvin. Encoding music performance data in humdrum and MEI. *Int. J. on Digital Libraries*, 20(1):81–91, 2019.

[8] Mathieu Giraud, Richard Groult, and Emmanuel Leguy. Dezrann, a web framework to share music analysis. In Sandeep Bhagwati and Jean Bresson, editors, *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, pages 104–110, Montreal, Canada, 2018. Concordia University.

[9] Nami Iino, Mayumi Shimada, Takuichi Nishimura, Hideaki Takeda, and Masatoshi Hamanaka. Proposal of an annotation method for integrating musical technique knowledge using a GTTM time-span tree. In Ioannis Kompatsiaris, Benoit Huet, Vasileios Mezaris, Cathal Gurrin, Wen-Huang Cheng, and Stefanos Vrochidis, editors, *MultiMedia Modeling - 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8-11, 2019, Proceedings, Part I*, volume 11295 of *Lecture Notes in Computer Science*, pages 616–627. Springer, 2019.

[10] Phillip B. Kirlin. A data set for computational studies of Schenkerian analysis. In *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*, pages 213–218, 2014.

[11] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. The MIT Press, Cambridge, MA, 1983.

[12] Alan Marsden, Satoshi Tojo, and Keiji Hirata. No longer 'somewhat arbitrary': Calculating salience in GTTM-style reduction. In *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, DLfM '18, pages 26–33, New York, NY, USA, 2018. ACM.

[13] Markus Neuwirth, Daniel Harasim, Fabian C. Moss, and Martin Rohrmeier. The Annotated Beethoven Corpus (ABC): A dataset of harmonic analyses of all Beethoven string quartets. *Frontiers in Digital Humanities*, 5:16, 2018.

[14] Laurent Pugin, Rodolfo Zitellini, and Perry Roland. Verovio: A library for engraving MEI music notation into SVG. In Hsin-Min Wang, Yi-Hsuan Yang, and Jin Ha Lee, editors, *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*, pages 107–112, 2014.

[15] David Rizo and Alan Marsden. A standard format proposal for hierarchical analyses and representations. In *Proceedings of the 3rd International Workshop on Digital Libraries for Musicology*, DLfM 2016, pages 25–32, New York, NY, USA, 2016. ACM.

[16] Jesse Rodin, Craig Sapp, and Clare Bokulich. Josquin research project [digital resource], 2010.

[17] Martin Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, 2011.

[18] Craig Stuart Sapp. Visual hierarchical key analysis. *Computers in Entertainment (CIE)*, 3(4):1–19, October 2005.

[19] Dmitri Tymoczko, Mark Gotham, Michael Scott Cuthbert, and Christopher Ariza. 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*, 2019.

[20] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems *SIAM journal on computing*, 18(6):1245–1262, 1989.