Data Quality Matters: Iterative Corrections on a Corpus of Mendelssohn String Quartets and Implications for MIR Analysis

Jacob Degroot-Maggetti, Timothy de Reuse, Laurent Feisthauer, Samuel Howes, Yaolong Ju, Suzaka Kokubu, Sylvain Margot, Néstor Nápoles López, Finn Upham

DATA QUALITY MATTERS: ITERATIVE CORRECTIONS ON A CORPUS OF MENDELSSOHN STRING QUARTETS AND IMPLICATIONS FOR MIR ANALYSIS

Jacob deGroot-Maggetti[1,2] **Timothy de Reuse**[1,2] **Laurent Feisthauer**[2,3]
Samuel Howes[1,2] **Yaolong Ju**[1,2] **Suzuka Kokubu**[1,2] **Sylvain Margot**[1,2]
Néstor Nápoles López[1,2] **Finn Upham**[1,2]
[1] Schulich School of Music, McGill University, Canada
[2] Computational Tonal Study Group
[3] Department of Information, University of Lille, France
jacob.degroot-maggetti@mail.mcgill.ca, timothy.dereuse@mail.mcgill.ca,
laurent.feisthauer@univ-lille.fr, samuel.howes@mail.mcgill.ca, yaolong.ju@mail.mcgill.ca,
suzuka.kokubu@mail.mcgill.ca, sylvain.margot@mail.mcgill.ca,
nestor.napoleslopez@mail.mcgill.ca, finn.upham@mail.mcgill.ca

ABSTRACT

In this paper, we describe a workflow of successive corrections on Optical Music Recognition (OMR) generated MusicXML files and their respective outputs under Music Information Retrieval (MIR) tasks. The original OMR-generated files of six Mendelssohn String Quartets were initially corrected by individual members of this interdisciplinary group, then reviewed by others to further standardize the quality and music analysis priorities of the team. Four MIR tasks are applied to each round of corrections on this collection: cadence detection, chord labeling, key finding, and monophonic pattern discovery. We measure changes in the outputs of these four MIR tasks from one round of corrections to the next in order to evaluate the impact of corrections. Results show that expert revision is more beneficial to some MIR tasks than to others. The resulting corpus of curated MusicXML files is available as an open-source repository under a Creative Commons Attribution 4.0 International License for further MIR research.

1. INTRODUCTION

Music Information Retrieval (MIR) algorithms that analyze symbolic music require high-quality data to produce accurate results. When building symbolic music corpora for MIR research, manually transcribing data using music notation software is expensive [1].

A faster option might be to use Optical Music Recognition (OMR) software on existing images of printed scores as an initial step. For example, Condit-Schultz et al. [2] worked on automated harmonic analysis of 571 chorales by Johann Sebastian Bach and Michael Praetorius. OMR was used in the process of creating symbolic encodings, with the results reviewed and manually corrected by a human annotator. Cumming et al. [3] created symbolic corpora of Renaissance music using OMR-generated scores as the first step and followed strict guidelines of manual corrections for the retention, addition, or removal of specific notations such as ties and fermatas. Although the performance of OMR applications has been improving over the years [4], extensive manual revisions are still required to ensure data quality and consistency for MIR analysis. This expensive and time-consuming task is especially relevant for OMR-induced errors since small ambiguities can lead to substantial variation in analytical output [5]. *What are the impacts of expert curation on data for MIR analysis tasks?*

We answer this question using files produced in the process of building a symbolic corpus of Mendelssohn string quarters. The OMR-generated passed through three iterations of increasingly-stringent manual corrections without additional annotations. We measured the impact of each round of corrections on four MIR analysis tasks (key finding, chord labeling, melodic pattern discovery, and cadence detection) through the changes between each iteration. Expert analysis of the scores exposes the types of errors to which these tasks are sensitive, demonstrating the need to tune corpus content to the anticipated analyses.

2. CORPUS CREATION

2.1 Mendelssohn String Quartets

Our corpus consists of the string quartets of Felix Mendelssohn. The Classical string quartet is a particularly relevant genre for computer-assisted analysis of music. In contrast to piano repertoire, where voice leading is often obscured by the limitations of the performer's hands and by what can be notated on the grand staff, a string quartet score preserves the independent parts of four separate instruments, allowing attribution of the role of each part (such as melody, bass, accompaniment, leading and imitative voices, etc.). The Classical aesthetic is characterized by a very clear harmonic, melodic, and formal organization, so it is not surprising that Beethoven's and Mozart's string quartets have already been encoded and annotated [6, 7] for music analytical purposes. Mendelssohn's string quartets are a natural next step; his works have Classical characteristics that place them in the tradition of Beethoven's late quartets [8].

Specifically, we encoded quartets Op. 12, Op. 13, and Op. 44, Nos. i, ii, and iii, all composed between 1827 and 1847, and *Four Pieces for String Quartet*, Op. 81. [1] The initial OMR encodings of these 24 movements were generated from scans of the 1875 edition published by *Breitkopf und Härtel*, available as PDF files on IMSLP. [2]) Although not part of the set studied in this paper, an additional quartet, Op. 80, is incorporated in our final published corpus. It had been previously encoded in MusicXML [3] by user *Musemeister*.

2.2 The CTS Team

An interdisciplinary team of nine people collaborated to create this corpus, with members from music technology, music theory, string performance, and music cognition. Each member brought unique viewpoints, skillsets, and objectives to the project. Whether they were interested in a specific MIR task or the applications of MIR to music theory, cognition, or pedagogy, team members refined encoding objectives together to satisfy their varied interests during the iterative correction and cleaning of OMR-generated symbolic music files.

2.3 From PDF to MusicXML

The first step in building the corpus was to transcribe the PDF files into a symbolic, machine-readable format. We used the commercial OMR software PhotoScore to analyze the original score images in PDF. It first detected the position of each staff on the page and we ensured that these detected positions were correct, adjusting as necessary. We also manually corrected the key and time signatures. The OMR results were then exported to MusicXML because the format is widely supported by music notation software. These files formed the Corrections 0 dataset "C0",

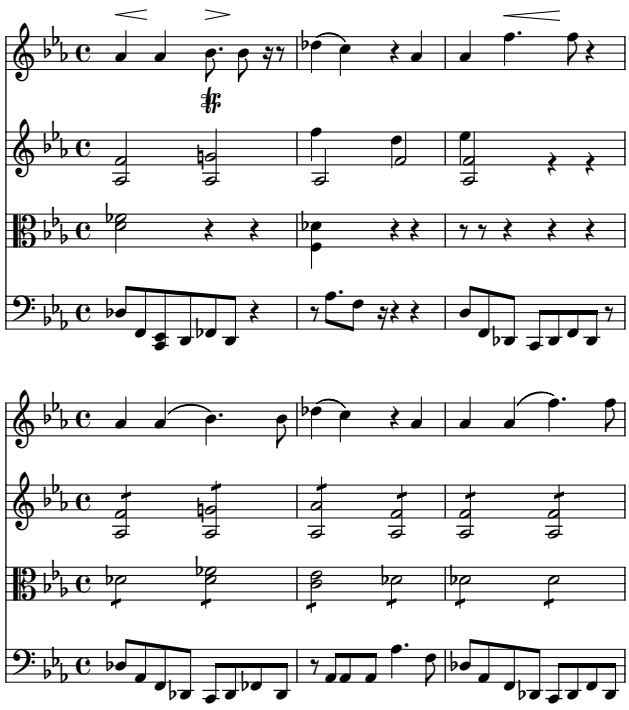

Figure 1. Measures 60-62 of Op. 44, No. iii, Mvt. 4. The upper system is initial OMR output (C0) and the lower system is after three rounds of manual corrections (C3).

with no additional manual corrections of score information. All subsequent corrections were made using MuseScore v2.3.2.

2.4 OMR Corrections

Starting from these initial OMR-generated music files (C0) we applied three successive stages of manual corrections: "C1", "C2", and "C3". The goal of each stage was to improve the accuracy of the previous stage(s) and to ensure that all information necessary for the MIR algorithms was included. The original C0 files included score elements both unlikely to be used by existing symbolic MIR algorithms and deemed by the team's music theorists to be less essential for the specific analytical approaches that we chose. Many of these score elements—for example, hairpin dynamics—were ignored or removed during the rounds of corrections (for a full list, refer to the supplementary materials).

The focus of C1 was accuracy of pitch and rhythm, while elements such as dynamic markings and articulations were largely ignored in the interest of time. As work progressed, it became clear which errors were most common in the OMR output. For example, there were many misaligned or missing notes in passages with higher note density (see Figure 1). The OMR software frequently encoded ties as slurs and *vice versa*. Despite their visual similarity, these curved lines produce different rhythmic values of notes, with consequences for our MIR analysis algorithms. Mendelssohn's scores also included many detailed performance instructions, prompting lengthy discussions about what information should be preserved in the final dataset

[1] These four independent pieces were gathered up in one opus and published after Mendelssohn's death.

[2] Downloadable at `https://bit.ly/2zzS0Bk`.

[3] Downloadable at `https://bit.ly/3dRC9wZ`.

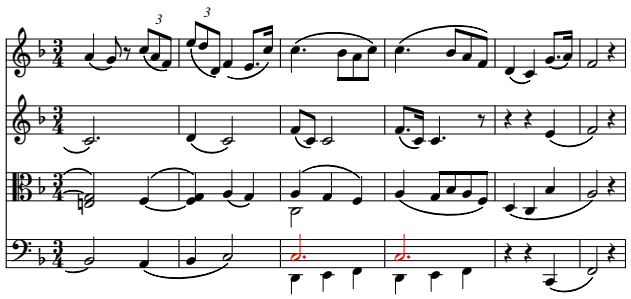

Figure 2. Measures 14-19 of Op. 13, Mvt. 2.

and what should be left out. With a more rigorous protocol in place, we reviewed each other's work to produce C2. The main purpose of this phase of review was to ensure that no errors had been missed in the previous round of revision. Again, during C2, details of the score came to light that necessitated further discussion, and the varied perspectives of the multidisciplinary team informed decisions about how to proceed. For example, double-stops, where more than one string is played at once, posed a recurring challenge. While chords on a staff line can be encoded in most analysis systems, Mendelssohn wrote many passages with moving lines against held notes (see Figure 2). This texture is easily misinterpreted by algorithms unfamiliar with the particularities of string music. Ultimately, note accuracy and articulation of onsets (namely, ties, slurs, and staccatos) were prioritized, while most indications of dynamics and ornamentation were excluded: a trade-off between comprehensiveness and machine interpretability.

Finally, the last round of corrections (C3) was a review to align all the encodings according to the conclusions of discussions during C2 and to standardize formatting and metadata. A full account of the score elements preserved in the final dataset is included in the supplementary materials. For consistency, a single person reviewed all the movements in preparing C3. Further discussion is provided in Section 4.1.

2.5 Differences between Correction Rounds

Besides a qualitative report on the amount of corrections we made in each round, quantitative measurement of the scale of changes is possible on these digital files. As the ideal MusicXML tool [9] is not publicly available with open source code, we used the more generic `SequenceMatcher.quick_ratio()` from Python's `difflib` library to produce percent differences. The median and range of similarity scores between C0 and C1, C1 and C2, and C2 and C3 are shown in Table 1.

Interpreting these numbers directly is difficult as MusicXML files include a plethora of elements beyond the focus of our corrections. Still, it is reassuring to see the median difference between successive corrections decrease by an order of magnitude each round. If all file modifications had the same impact on the MIR analyses, their outputs would show a similar pattern of decreasing impact.

Comparison Pair	Percent Difference median [min, max]
C0 to C1	10.0% [2.8%, 21.8%]
C1 to C2	1.3% [0.0%, 7.3%]
C2 to C3	0.2% [0%, 1.3%]

Table 1. Percent difference for each comparison pair. The results are medians across all 24 movements, with maximum and minimum values indicated in brackets.

3. MIR ALGORITHMS

Four symbolic MIR algorithms were applied to each version of the Mendelssohn String Quartet Corpus. These algorithms were chosen because they were either designed or extensively used by members of the CTS team. Without ground truth annotations to assess the *accuracy* achieved by each MIR tasks, the corrections were evaluated through their *perceivability* to the algorithms in output *changes* between successive versions. [4] For two of the tasks that produce sequences of annotations, results from different versions of the same movement had to be aligned to one another before comparison; this procedure is detailed in the supplementary materials. In total, 96 evaluations per task were performed as each analysis algorithm was applied to all four versions of the 24 movements in the corpus.

3.1 Key Analysis

A recent key-finding algorithm [10] provided two predictions: global key per movement and local key per onset slice. We ran the algorithm using the default parameters provided in the implementation. Between C0 and C1, predictions of global key changed in 3 of the 24 files tested. There was no change in prediction between C1, C2, and C3. Predictions of local key changed substantially between C0 and C1 across all files, and changed much less between C1 and C2, and between C2 and C3, as shown in Table 2.

Comparison pair	Changes in local key annotations median(%) [min(%), max(%)]
C0 to C1	46.8% [9.9%, 71.1%]
C1 to C2	0.4% [0.0%, 9.8%]
C2 to C3	0.0% [0.0%, 3.0%]

Table 2. Percent differences in local key annotations for each comparison pair. The results are medians across all the 24 movements with minimum and maximum values indicated in brackets.

3.2 Chord Labeling

The automatic chord labeling model [11] was applied to each stage of the dataset, predicting chords for every onset slice of the piece.

[4] E.g. comparing the outputs of a key-finding algorithm applied to C0 and C1.

Comparison Pair	Changes in chord labels median(%) [min(%), max(%)]
C0 to C1	69.1% [17.5%, 96.7%]
C1 to C2	0.7% [0.0%, 41.1%]
C2 to C3	0.0% [0.0%, 12.5%]

Table 3. Percent difference in chord annotations for each comparison pair, shown as medians across all the 24 movements with minimum and maximum values indicated in brackets.

Each chord is labelled according to its root (C, F♯, B♭, etc.) and its quality (major, minor, fully diminished seventh, dominant seventh, etc.), with no mention of its inversion or its harmonic function (Roman numeral analysis).

The results are shown in (Table 3). We can see differences between C0 and C1 were substantial (median 69.1%), while the percent change between C1 and C2 was much smaller (median 0.7%). The majority of the movements showed no change in chord labels between C2 and C3 (median 0.0%). Such results indicate that chord labelling is not sensitive to local differences.

3.3 Monophonic Pattern Discovery

The SIARCT-C Algorithm [12] was used on each version of each movement to discover sets of repeating patterns. A "pattern" here refers to a set of excerpts of a piece that are all nearly identical in pitch and rhythm under transposition. While the algorithm is capable of operating on polyphonic music, here we focus on finding monophonic patterns between voices. To this end, each MusicXML file was transformed into point sets of (onset time in quarter notes, morphetic pitch) pairs; for example, the first measure of the first violin's part in Figure 1 is notated as the sequence `(69, 0) (69, 1) (71, 2) (71, 2.75)`. Dynamics, articulatons, and durations are discarded. Some algorithmic pattern discovery methods do use this kind of information, such as the Automatic Timespan Tree Analyzer [13], but the majority use only rhythmic and pitch-related data, partially due to the computational complexity of the problem. The four voices in each file were concatenated into one sequence for the purpose of evaluation.

We searched only for patterns that were at least eight notes long that occurred at least five times within each movement, allowing for a small amount of variation. These parameters were chosen as a compromise in light of the number of movements we had to analyze and the running time of the algorithm; searches for short patterns take significantly longer than searches for long patterns. We illustrate the effect of iterative corrections by their impact on descriptive statistics of these results: the number of unique patterns detected, the coverage of these patterns over all notes in the music, and the median cardinality (i.e., number of instances) of each pattern discovered. [5]

Table 4 shows how these statistics change between versions. Many more patterns were found in C1 than in C0 (median 85%, maximum 2100%), with a small amount of gain and loss from C1 and C2, and no change in total from C2 to C3. Coverage also grew substantially from C0 to C1, including more than twice the number of notes after this first round of corrections for more than half the movements. In contrast, the median cardinality did not change as drastically for those patterns detected in these different versions, and no apparent changes occurred during the last round of corrections.

Comp. Pair	Magnitude Increase, median [min, max] (%)		
	Num. Patterns	Coverage	Cardinality
C0 to C1	85% [16%, 2100%]	110% [21%, 1100%]	8.3% [-22%, 29%]
C1 to C2	0.0% [-5.1%, 7.1%]	0.041% [0.85%, 9.8%]	0.0% [-3.4%, 3.5%]
C2 to C3	0.0% [0.0%, 0.0%]	0.0% [0.0%, 0.0%]	0.0% [0.0%, 0.0%]

Table 4. Median magnitude increase in three statistics taken on the sets of discovered patterns over the course of the corrections. Minimal and maximal values for change are shown in brackets.

Comparison pair	Change in PACs detected	New PACs detected	PACs lost
C0 to C1	154.5% 22 to 56	177.3% 39	22.7% 5
C1 to C2	-3.6% 56 to 54	1.8% 1	5.4% 3
C2 to C3	1.9% 54 to 55	3.7% 2	1.9% 1

Table 5. Change in the number of PACs detected. 'New PACs detected' report the number of PACs detected in the latter that were not in the former. 'Lost PACs' is the number of PACs that were in the former but not the latter. As the number of PACs detected is quite small, both relative changes and exact numbers are given.

3.4 Cadence Detection

Finally, the cadence detection algorithm introduced by Bigo et al. [14] was used to detect perfect authentic cadences (PACs) [6] throughout the corpus. Each beat was evaluated as a potential point of cadential arrival using a Support Vector Machine. As there are only few cadences within single movements, Table 5 reports the results of

[5] Comparing sets of discovered patterns is difficult because of their highly heterogeneous structure, with individual patterns spanning a wide range of cardinalities and number of notes per instance. While it is possible to devise a more direct evaluation based on similarities between individual patterns, we used an approach based on descriptive statistics for the sake of brevity and interpretability.

[6] Too few cadences of other types, such as half cadences, were successfully detected to interpret sensibly in this context.

this evaluation as a count of cadences detected rather than percent change. As expected, the model benefited greatly from the initial round of corrections: twice as many cadences were identified in C1 files as in C0 files. Additional rounds of reviews had little impact on the total number of PACs detected. Close investigation of the differences between detected cadences in C1, C2 and C3 revealed that some changes in the algorithm's output were due to corrections in notated pitch.

4. DISCUSSION

The different rounds of corrections prompted a wide range of considerations for the group, which are discussed below. Proofreaders with different kinds of expertise, whether in MIR, music theory, string musicianship, or the use of the chosen music notation software, communicated various concerns and discoveries relating to their respective tasks. Finally, special situations are discussed, in which music theoretical and analytical considerations collide with MIR objectives in notable ways.

4.1 MIR Significance of Correction Rounds

Using OMR to create datasets of symbolic music is an attractive proposition. Our results suggest that there is significant variation in the quality of the output between files when using software like PhotoScore. No task evaluated here was able to totally overcome the errors introduced by OMR, with all of the results seeing some amount of change after the first round of corrections, and the degree of change in this initial round varied widely between tasks. Global key estimations changed for only 3 of the 24 movements, while the discovered patterns on the raw OMR output bear little resemblance to those discovered after just one round of corrections. However, these findings cannot be extrapolated directly to other algorithms that perform the same tasks; different machine-learning methods may cause models to become more sensitive to some errors and less sensitive to others. For the specific algorithms applied here, we may consider this as evidence that the underlying symbolic-musical structures they use to make judgments are affected by errors in the OMR process to different magnitudes. For most tasks, though, initial correction is necessary when using OMR to create datasets, given the current capabilities of commercially-available OMR software.

For subsequent rounds of corrections, the sizes of changes shrink dramatically but still vary between tasks. In particular, the discovered patterns barely change at all after the first round of corrections; this is likely due to the fact that the algorithm uses only onset times and morphetic pitch, thus ignoring some pitch changes with harmonic consequences.

4.2 Experience of Doing Corrections

For the members of our team with solid experience in copying music, correcting OMR required an average time of 30 to 45 minutes per printed page depending on the variety and amount of errors. To review Mendelssohn's complete string quartets (C0 to C1) thus took approximately 75 to 110 hours. Even though the standardization step (C1 to C2) in itself was much shorter (5 to 15 minutes per printed page, or an approximate total of 25 hours), discussions concerning what should be kept and what should be ignored lasted over a month. Finally, checking the consistency represented 30 additional hours (C2 to C3). While the dataset at C3 was standardized to meet the requirements of our analysis tasks, one might wonder whether investing this amount of time was necessary.

Different movements, and different passages within individual movements, required vastly different degrees of effort to correct. In some sections, only corrections to articulations and accidentals were needed, whereas other sections needed to be completely rewritten. The first round of corrections (C0 to C1) was the most difficult, involving many decisions about which elements of the score to preserve. Some time-consuming corrections had to be rolled back after standardization protocols had been decided on. This round of corrections was particularly difficult for proofreaders who had never used MuseScore. Certain features of the program, such as the addition of key signatures, introduced multiple additional errors when used incorrectly, while some features that might have saved time, such as batch addition of articulation marks, went unused through much of the correction process. There were a few musical situations that tended to produce predictable errors in the OMR encoding. Errors frequently arose when the OMR software missed or misplaced rests, and proofreaders quickly learned that passages with higher note density required much more effort to correct.

4.3 Musical Considerations

Figure 1 gives a general sense of the differences between the initial (C0) and final (C3) stages of the correction process. These differences fall into two broad categories: "pitch-rhythm" differences in the vertical (pitch) or horizontal (rhythm) placement of notes and "notational" differences in articulations, ornaments, and other non-pitch elements of the score. Pitch-rhythm differences had a large effect on the outcomes of analysis tasks. Pitches that were incorrect (Vla., m. 61), misaligned (Vln. 1, m. 62), missing (Vla., m. 62), or extraneous (Vln. 2, m. 61; Vc., m. 60) affected all four analysis tasks. Notational differences in tremolos (Vln. 2 & Vla., mm. 60-62), and slurs (Vln. 1, m. 60) had a smaller effect on the outcomes of analysis tasks, but could be disruptive in MIR tasks that make use of recurring notational cues, especially for pattern finding or cadence detection. For example, Mendelssohn often uses slurs, staccatos, and dynamic markings such as hairpins to highlight recurring motives. When these motives are transposed or altered non-uniformly, as in the tonal answer of a fugue or the development section of a sonata-form movement, they may become undetectable by pattern-finding algorithms that rely exclusively on pitch and rhythm. An analyst relying on these results may be led to misinterpret larger tonal, hypermetric, and formal structures if, for example, the algorithm fails to detect a main theme at the

beginning of a returning section. Algorithms that also consider articulations and dynamic markings might perform better in situations like these. One other complication is that PhotoScore sometimes generates hidden rests (Vla. m. 61), slowing the correction process by making the score less readable to humans.

Extraneous or incorrect clefs were found in four of 24 movements during the final round of corrections (C3). In the fourth movement of Op. 44 No. iii, an incorrect (French violin) clef in the first violin transposed the part up by a major third. This error persisted through several rounds of correction because it was obscured by system breaks. Incorrect clefs had a large effect on key-finding, chord-labeling, and cadence-detection tasks, but not on the transposition-invariant pattern discovery task.

Another recurring issue concerns multiple voices within a single staff (i.e., "double-stops" on a string instrument). While this is possible to encode in MusicXML format, it is often unreadable to the software used for analysis tasks. In these cases, the encoder must decide which voice to keep and which to discard, which can result in the loss of information. In Op. 13, Mvt. 2, mm. 16-17 (see Figure 2), the cello plays both a held C (in red) along with a moving line D-E-F. Without the moving line, the C becomes a pedal at the bottom of the texture, changing the harmonic and cadential sense of the music. In Op. 13, Mvt. 3, mm. 143-53, the cello and violin II have melodic lines below a harmonic pedal resulting in two-note chords for each of them. Choosing one voice or the other is difficult: keeping the pedal tones preserves the harmony, allowing for the detection of cadences and chords; keeping the melodic motives allows for the discovery of more patterns throughout the movement. Deciding what to keep depends on the type of analysis to be carried out.

4.4 Implications for OMR

The initial errors in OMR files disproportionately impacted these tasks: a 10% change in the MusicXML files produced a 47% change in local key judgments and a 69% change in chord labels. This proportion of incorrect results underlines how this commercial software struggled with these scores. One cause was missing notes: runs of sixteenths and eighths typical of this genre of music were often dropped, when the dense or complicated stemmings were incorrectly interpreted. The numerous articulation and ornamentation markings were often misinterpreted as notes, suggesting poor recognition of shapes within staves. There was also confusion between parts in the four staff systems, with notes and text annotations packed more tightly in vertical arrangements than in other genres and publication styles. Software tuned to this era and style of work would hopefully reduce the amount of information lost. However, given the variety in performance quality across these scores, human supervision is highly recommended.

5. CONCLUSION

This project is a case study in how human corrections on OMR can influence MIR analysis results. The range of outcomes across these analyses suggests that the value of human correction time depends on the MIR task. If some noise in the results is permissible and one is only interested in large-scale qualities like global key, the raw OMR files may suffice, but anything closer to the notes would benefit from some review and correction. Without human intervention, half of the outputs for local key detection and chord labeling were corrupted, while monophonic pattern discovery and cadence detection missed substantial portions of the relevant material in most pieces.

The second and third rounds of corrections had progressively smaller effects on the aggregate results of these analyses, as expected, but there are instances when these smaller changes were crucial for the type of analysis at hand. Passing the symbolic music files between multiple reviewers minimized the impact of human error. Some of the changes in these later rounds were motivated by new understanding of the music, music encoding limitations, and what could be used by our MIR algorithms.

This project is not representative of all symbolic music corpus-building with OMR. PhotoScore was not necessarily the best OMR processor for string quartet music printed in 1875. The sensitivity of these MIR analysis tasks on changes in symbolic music information is also specific to their implementations; a study of monophonic patterns that included articulation or dynamics would tell a different story from that above. However, the novel comparison process across iterations of corrections highlights the importance of expert musical care in the developing of symbolic music corpora, as well as the need for explicit acknowledgement of the types of score information preserved therein.

Discussions between team members about the potential relevance of ornamentation and articulation to each analytical objective resulted in a set of files that contained more information than could be used by the algorithms applied here. At the same time, important layers like dynamics were removed because of the difficulty of producing machine-interpretable encodings. We hope to see that the retained layers of performance information are used in future work with this collection of symbolic scores, and that symbolic music encoding and analysis tools continue to progress towards capturing a richer range of musical information. The final version of this Mendelssohn String Quartet Corpus, a pedagogical, scholarly, and artistic resource for musicians, composers, and music researchers alike, can be downloaded from: `https://github.com/DDMAL/felix_quartets_got_annotated`.

While OMR can be a helpful tool for corpus building, such projects still require human expertise in both the music represented and in its intended uses. For MIR-related research, some tasks benefit from manual review more than others.

6. ACKNOWLEDGEMENTS

We would like to acknowledge the contributions of our many collaborators on the Single Interface for Music Score Searching and Analysis (SIMSSA) project, especially Ichiro Fujinaga. The authors' names are ordered alphabetically, and each author shares an equal contribution to this paper.

7. REFERENCES

[1] M. Gotham, P. Jonas, B. Bower, W. Bosworth, D. Rootham, and L. VanHandel, "Scores of scores: An openscore project to encode and share sheet music," in *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, ser. DLfM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 87–95. [Online]. Available: https://doi.org/10.1145/3273024.3273026

[2] N. Condit-Schultz, Y. Ju, and I. Fujinaga, "A flexible approach to automated harmonic analysis: Multiple annotations of chorales by Bach and Prætorius," in *Proc. of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 66–73.

[3] J. E. Cumming, C. McKay, J. Stuchbery, and I. Fujinaga, "Methodologies for creating symbolic corpora of Western music before 1600," in *Proc. of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 491–498.

[4] J. Calvo-Zaragoza, J. Hajic Jr, and A. Pacha, "Understanding optical music recognition," *arXiv preprint arXiv:1908.03608*, 2019.

[5] N. Nápoles López, G. Vigliensoni, and I. Fujinaga, "Encoding matters," in *Proc. of the 5th International Conference on Digital Libraries for Musicology*, Paris, France, 2018, pp. 69–73. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3273024.3273027

[6] M. Neuwirth, D. Harasim, F. C. Moss, and M. Rohrmeier, "The Annotated Beethoven Corpus (ABC): A dataset of harmonic analyses of all Beethoven string quartets," *Frontiers in Digital Humanities*, vol. 5, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fdigh.2018.00016

[7] N. Zaslaw, "Review: Digital Mozart Edition (DME)," *Journal of the American Musicological Society*, vol. 71, no. 2, pp. 572–586, 08 2018. [Online]. Available: https://doi.org/10.1525/jams.2018.71.2.572

[8] R. L. Todd, "Mendelssohn(-Bartholdy), (Jacob Ludwig) Felix," *Grove Music Online*, 2000, available at https://doi.org/10.1093/gmo/9781561592630.article.51795. Accessed March 20th, 2020.

[9] F. Foscarin, F. Jacquemard, and R. Fournier-S'niehotta, "A diff procedure for music score files," in *6th International Conference on Digital Libraries for Musicology*, 2019, pp. 58–64.

[10] N. Nápoles López, C. Arthur, and I. Fujinaga, "Key-finding based on a hidden markov model and key profiles," in *Proc. of the 6th International Conference on Digital Libraries for Musicology*, New York, NY, 2019, pp. 33–37.

[11] Y. Ju, S. Howes, C. McKay, N. Condit-Schultz, J. Calvo-Zaragoza, and I. Fujinaga, "An interactive workflow for generating chord labels for homorhythmic music in symbolic formats," in *Proc. of the 20th International Society for Music Information Retrieval Conference*, Delft, Netherlands, 2019, pp. 862–869.

[12] T. Collins, A. Arzt, S. Flossmann, and G. Widmer, "SIARCT-CFP: Improving precision and the discovery of inexact musical patterns in point-set representations," in *Proc. of the 14th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2013, pp. 549–554. [Online]. Available: http://www.cp.jku.at/research/papers/collins_etal_ismir_2013.pdf

[13] M. Hamanaka, K. Hirata, and S. Tojo, "ATTA: Automatic time-span tree analyzer based on extended GTTM," in *Proceedings of the 6th International Society for Music Information Retrieval Conference*, London, UK, 2005, pp. 358–365.

[14] L. Bigo, L. Feisthauer, M. Giraud, and F. Levé, "Relevance of musical features for cadence detection," in *Proc. of the International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 355–361. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01801060