METHODOLOGIES FOR CREATING SYMBOLIC CORPORA OF WESTERN MUSIC BEFORE 1600

Julie E. Cumming
McGill University
julie.cumming@mcgill.ca

Cory McKay
Marianopolis College
cory.mckay@mail.mcgill.ca

Jonathan Stuchbery
McGill University
jonathan.stuchbery@mail.mcgill.ca

Ichiro Fujinaga
McGill University
ichiro.fujinaga@mcgill.ca

ABSTRACT

The creation of a corpus of compositions in symbolic formats is an essential step for any project in systematic research. There are, however, many potential pitfalls, especially in early music, where scores are edited in different ways: variables include clefs, note values, types of barline, and editorial accidentals. Different score editors and optical music recognition software have their own ways of storing and exporting musical data. Choice of software and file formats, and their various parameters, can thus unintentionally bias data, as can decisions on how to interpret potentially ambiguous markings in original sources. This becomes especially problematic when data from different corpora are combined for computational processing, since observed regularities and irregularities may in fact be linked with inconsistent corpus collection methodologies, internal and external, rather than the underlying music.

This paper proposes guidelines, templates, and workflows for the creation of consistent early music corpora, and for detecting encoding biases in existing corpora. We have assembled a corpus of Renaissance duos as a sample implementation, and present machine learning experiments demonstrating how inconsistent or naïve encoding methodologies for corpus collection can distort results.

1. INTRODUCTION

Because creating accurate corpora is extremely labour intensive, early music researchers often draw on symbolic scores already available online. These collections, however, exhibit many different approaches to encoding scores, depending on the choices of the individual who did each encoding, the music editor used, the particular symbolic music file formats used, and the ways in which those files were generated. Even when transcribing music directly into a music editor, it is important to have clear guidelines for many elements of the transcription. A good corpus, therefore, requires a clear set of guidelines and templates for notation and file creation. It also requires a workflow that integrates correction, and consistent processes for generating symbolic files. We describe an effective process for encoding a consistent corpus for research projects on Renaissance music, and use it to create a publicly-available collection of duos. We end with an experiment involving this dataset showing how different or inconsistent encoding methodologies can distort results.

1.1. Related Work

Several collections of symbolic Renaissance scores exist. The Choral Public Domain Library (CPDL) [4] includes large amounts of Renaissance music, but there is no attempt at standardization. The original ELVIS database [5] also aimed for quantity without much curation, but with substantial metadata. The Josquin Research Project (JRP) [21] is carefully curated and extremely consistent. Smaller collections assembled for specific projects, such as [8], [12], [13], [19], [20], and [22], are carefully curated, but each uses a different approach.

2. RESEARCH CORPORA IN RENAISSANCE MUSIC: NOTATIONAL CONSISTENCY

In Renaissance music manuscripts and prints the parts are not aligned in score. Instead they are presented in separate parts (on different parts of the page or in separate partbooks). In order to study this music the parts must be transcribed and combined into a score. Mensuration signs (similar to time signatures) indicate the metrical organization, but the parts have no barlines, and ties are never used. There are multiple different clefs (C clefs on any line; F clefs on three lines; G clef is rare). Performers are expected to add accidentals in specific melodic and contrapuntal situations without explicit accidentals in the score (resulting in debates among performers and editors of early music). Note values are larger than those of common Western notation: between 1450 and 1550 the beat normally falls on the semibreve (whole note).

Modern editors have a wide variety of approaches to transcription, as described in in [3] and [14]. Some try to make the edition look like 18th-century music, while others try to preserve elements of the original notation, and everything in between. There are editions of Renaissance music scores in original clefs and modern clefs; with barlines, without barlines, or with *mensurstriche* (barlines that only appear between the staves). We can find scores with original, halved, quartered, and smaller note values.

Most editors introduce editorial accidentals, but there are multiple possibilities, and few agree on every decision. Editors often also transpose works (for performance by a specific ensemble, or because they believe that the original pitch was higher or lower than the "written" pitch in the original source). The same piece of music edited by different people will look very different (see Figure 1). Transcribing works directly from the original sources is extremely time consuming, however, so if a piece is available in modern transcription, we normally start with that, either by transcribing it or by using an OMR program such as PhotoScore, and then correct it manually.

Figure 1. Contrasting editions of Josquin Desprez, *Missa de beata virgine*, Agnus II, mm. 1–7. Top: original note values with *mensurstriche* from [10]. Middle: halved note values with barlines from [9]. Bottom: our edition, with original note values, barlines, and time signature that matches the measure length.

2.1. Problems Resulting from Inconsistent Notation

When converting published scores into a symbolic corpus for music research (through OMR or transcription with a music editor), or when taking symbolic scores from an online repository, it is essential to make the notation of the scores consistent. Inconsistent notation can cause significant errors in computational analysis, as we show in the experiment described in Section 6 below. For example, when analysing counterpoint we normally sample the score at every minim (half note) in the original notation. If we have one score in original note values, and one in quartered note values, the half note will have a completely different meaning, and the results will not be comparable. The length of a work can also provide information on genre. If the measures are different lengths, because of different editorial decisions, then this data will be incorrect. When looking at issues of mode we normally check *final* and key signature; if a work is transposed, this will distort the data. If the number of beats in a measure does not match the time signature, software such as music21 [7] will not parse the symbolic score correctly.

2.2. Creating and Obtaining Symbolic Scores

The most straightforward way to create a symbolic file is to transcribe the piece into a music editor from images of the original source (Renaissance manuscript or print). While this is time consuming, especially if the original source is difficult to read, or if there are ambiguities in the notation, it results in a file that is very close to the original source.

All the other methods involve working with a modern edition: transcription into a music editor from a modern edition (we do this when the notation of the edition is not suitable for OMR); obtaining symbolic files from online repositories, including the CPDL [4] and the JRP [21]; or using an OMR program such as PhotoScore on a modern edition. Almost all of these files need adjustment with regard to note values, time signatures, editorial accidentals, and pitch level. As we constructed our corpus, we kept finding additional issues that required decisions, which we incorporated into guidelines and templates.

2.3. Our Guidelines for Consistency in Scores of Renaissance Music c. 1450–1550

In order to establish norms it is useful to decide on one source of authority, and to create a clear set of guidelines, as well as a template encapsulating the guidelines. We chose not to follow the standards of a single modern edition. Instead, we stayed as close to the original as possible, given that we are transcribing the pieces into modern notation in score, with barlines. This means that we use the original notated pitch of the work, original note values, and we do not include editorial accidentals, since these are often a subjective decision of a particular editor and there is rarely complete consensus among experts. For ease of reading we use modern clefs: treble clef, transposing treble clef, and bass clef (see Figure 1). We use time signatures and ties; most of our time signatures use the whole note as the beat (2/1 or 3/1). There are no time-signature change unless there is a real change of meter in the piece, and the time signature must match the length of the measure. The traditional final long is transcribed as two breves, tied over the bar. We only include fermatas found in the original source, and use a fermata symbol that does not affect the rhythmic value of the note. In general, correct and consistent encoding is considered more important than the appearance of the score, and more important than graphic features of the modern edition or the original notation, such as ligature brackets, ranges, and original clefs and note shapes.

3. ENCODING EARLY MUSIC

Once researchers have established notational norms for the corpus, they must also establish norms for encoding. When using pieces available on line, or when more than one person is creating symbolic files for the corpus, there are many possible sources of inconsistency: symbolic files in different formats; the use of different music notation software to generate files; different software versions; and different encoding settings for a given piece of software. We created a set of basic principles to address such problems, and incorporated them into our workflow and score editor templates.

3.1. Encoding Formats

We generate Sibelius, MIDI, Music XML, **kern, and PDF files for use in several different machine learning and music analysis contexts. Although it is arguably desirable to use purely open file formats when possible (e.g., for long-term compatibility), the ubiquity of a format is also an essential consideration, in order to maximize accessibility. We argue that presenting files in a variety of formats, open and closed, allows us to find a good compromise between these two concerns.

Much of the detail about encoding described here is focused on MIDI, which is important because of its ubiquity and because it requires that certain data be specified rather than left ambiguous (e.g., the tempo of a piece cannot be left undefined, as this will implicitly result in the default MIDI tempo being used). Although there are often good musicological reasons for ambiguity, it can cause serious problems for many systematic analysis, search, display, or feature extraction systems, which may use improper defaults or not work at all when faced with certain kinds of ambiguous data. From the specific perspective of computational music processing, MIDI helpfully forces encoders to specify best estimates in cases where there is ambiguity. The most important reason for choosing MIDI, however, is simply that it can be both parsed and produced by almost any software, and follows a universally accepted and open standard. That being said, MIDI has many well-publicized imperfections and limitations, so it is always advisable to distribute datasets in other formats, as we do.

3.2. Basic Principles for Encoding the Corpus

- Use the same software, software version, operating system, and encoding settings
- Use a uniform and short file naming convention, and only allow ASCII characters, as archiving or moving files between computers or network locations can cause problems with long file names or non-ASCII characters
- Encode provenance information directly in the files themselves, in case encapsulating databases, etc. are lost; use rich character sets when permitted
- Be consistent with:
  - Instrument names (e.g., “alto” singer vs. “alto” viola); be sure there are no missing instrument names that default to incorrect instruments
  - Dynamics
  - Tempo
  - Time signatures and meter changes
  - Key signatures
  - Voice segregation
  - Transposing treble clefs
  - Fermatas
  - Playback settings, affecting dynamics, varying tempo, note durations, etc. (disable rubato, swing, and “human playback” settings so that encodings are as rhythmically quantized as possible)
- For MIDI in particular:
  - Use MIDI Type 1
  - Conform to General MIDI instruments
  - Avoid keyboard instruments for non-keyboard parts, as keyboard encodings can sometimes cause individual voices in a polyphonic work to be collapsed into one part
  - Standardize to 960 PPQN (Pulse Per Quarter Note)
  - Set tempo to whole note = 80 BPM (quarter note = 320)
- Avoid:
  - Encoding methodologies that needlessly throw away information
  - Encoding methodologies that permit ambiguity (e.g., in note durations) in cases where automated feature extraction or analysis will be used
  - Format conversions: if they are necessary (e.g., in order to increase accessibility), generate all alternative encodings from a single master file

We dealt with consistency issues by building templates (blank pieces in the notation software with all the correct settings), into which we copied our pieces. These templates are available at [6].

3.3. Choice of Score Editing Software

We chose to use the latest version of Sibelius for compatibility and consistency reasons. It is one of the most widely used score editors, it works well with the PhotoScore OMR software, and it has a scripting language (ManuScript). It is also the only score editor that can be used to create MEI files, using the Sibelius MEI plugin [15]. Although there are certainly important advantages to using open-source software (e.g., MuseScore) when possible, there are no open-source alternatives to Sibelius that offer these essential advantages. That being said, Sibelius did initially cause us problems: the transposing clef often did not encode the voice in the lower octave, even though the “8” below the clef showed in the score. This distorts contrapuntal analysis (e.g., consonant fifths between voices turn into dissonant fourths).

4. WORKFLOW

In the process of developing our corpus we developed a workflow for file creation, including both manual and scripted processes that allowed us to avoid inconsistent file production. This workflow can be used by other researchers who want to create consistent corpora, and is available in more detail at [6]. It can be summarized briefly as follows:

- Create or collect symbolic files
- Copy the corrected symbolic files into the template

- Correct the files in Sibelius, following the guidelines in Section 2.3
- Check the files for problems (by looking at the PDFs, and comparing the files to original sources), and correct them manually when necessary
- Save the verified result as a "master file"
- Once all the desired master files for the corpus are assembled, generate all files in all alternative formats at the same time using a script
- Check MIDI files for consistency using jSymbolic [18] (which reveals inconsistent settings, including meter changes, dynamics, and tempo settings)

5. THE JOSQUIN / LA RUE DUOS CORPUS

We used the workflow and templates introduced above to create a corpus devoted to studying differences in the music of two leading Renaissance composers, Josquin Desprez (c. 1450–1521) and Pierre de la Rue (c. 1452–1518). These two composers are particularly interesting because it is difficult to tell their music apart, even for experts. They are almost exact contemporaries, and there are ten compositions attributed to both composers in different 16th-century sources. Past attempts to describe differences in style are often frustratingly vague, as in this discussion of why a La Rue Mass is not by Josquin: "the rhythmic motion and continuous repetition of the main melodic motif in mm. 45–66 lack the vitality characteristic of Josquin" [11].

Our corpus consists of duos (two-voice sections) from Masses by these two composers. It is important to compare works in the same genre, since different genres can result in different styles, even for the same composer. Also, composers and improvisers in the Renaissance began by learning to work in two voices; this is the purest form of Renaissance counterpoint. For this study we included only duos from Masses securely attributed to the composers (i.e., there is consensus that the Masses are not by another composer). For Josquin, we used the "secure" categories established by Jesse Rodin in the JRP [21]; for La Rue we used the assessments in the La Rue edition [17].

Most of the symbolic files in the corpus came from the JRP [21]. We searched the Masses for duo sections surrounded by double bars (separate sections of longer Mass movements). We downloaded the Music XML files for the relevant movements, opened them in Sibelius, and extracted the duos. Some additional movements were transcribed from the La Rue edition, restoring the original note values.

Our final corpus, titled the JLSDD (Josquin La Rue Secure Duos Dataset), after systematic cleaning, correction, and format translation, consists of 33 secure Josquin duos and 44 secure La Rue duos, each available as Sibelius, Music XML, MIDI, MEI, **kern, and PDF files at [6]. They are distributed with pre-extracted jSymbolic [10] features, and the Sibelius templates used to build the corpus may also be downloaded from [6].

6. EXPERIMENTS: JOSQUIN VS. LA RUE

We performed a series of machine learning-based composer attribution experiments in order to gain empirical insight into the effects of different encoding methodologies. For related studies on systematic composer classification, see [1], [2], and [16].

6.1. Datasets Used

All of the experiments described here made use of the 33 secure Josquin duos and 44 secure La Rue duos introduced in Section 5. We generated three different experimental MIDI datasets from this corpus:

- *Original:* All 77 secure Josquin and La Rue duos, generated from the Sibelius files, as they existed before systematic standards were used to correct, annotate, and encode them. These duos used a variety of General MIDI instrument patches, varying amounts of rubato added by Sibelius, varying amounts of dynamic variation added by Sibelius and inconsistent approaches to metrical annotation (e.g., time signatures of 4/4 and 8/4 vs. 2/1). Notably, these differences were distributed across the music of both composers, and were not meaningfully correlated with either of them.
- *Clean:* All 77 secure Josquin and La Rue duos, generated from the Sibelius files after systematic standardization had been applied. The files were all encoded using General MIDI Patch 53 (voice), all had a tempo of 80 whole-note beats per minute, all had time signatures based on whole-note beats and none had added rubato or dynamics. These are, in effect, the clean release version of the duos corpus described in Section 5.
- *Simulated:* The 33 secure Josquin duos, generated from the Original Sibelius files using systematic settings that differed from the settings used when generating the Clean dataset. This was done in order to allow us to simulate the effects of combining datasets acquired from different sources, where different encoding standards were used. In this case, all files were encoded using General MIDI 1 (piano), a tempo of 120 whole-note beats per minute, no rubato added, and no dynamics added. The choice of a piano patch had the additional effect of causing Sibelius to encode the notes from both voices into a single MIDI channel and track, thereby losing the explicit voice segregation found in the Original and Clean datasets.

6.2. Feature Extraction

Features were extracted from each of the Original, Clean, and Simulated datasets using the newest version (2.2) of the open-source jSymbolic software [18]. jSymbolic extracts 246 unique features from symbolic music files, including a number of multidimensional features, for a total of 1497 values. These features can be loosely grouped into the following categories: pitch statistics; melodic features; chords and vertical intervals; rhythm; instrumentation; texture; and dynamics. jSymbolic was chosen because it includes far more features than any other musical symbolic feature extraction software, and its extensive documentation and relatively easy-to-use interface make it particularly accessible to musicological researchers who may have less experience with MIR software.

Two sets of features were extracted for each experiment:

- *All Features:* All features implemented by jSymbolic that can be extracted from MIDI files.
- *Safe Features:* A subset of the All Features group that consists of just 173 of jSymbolic's 246 implemented features. These features omit all features associated with tempo, dynamics, instrumentation, and meter, among other things. The intention of these features is that they can be used even when datasets are in fact systematically biased based on encoding methodology (since the features that would be sensitive to these biases are not extracted). All features known to be associated with these qualities were left out, and then a further feature / class correlation check (see below) was performed in order to make sure no bias-sensitive features remained. The Safe Features are a good fit with Renaissance music, in which tempo, dynamics, and instrumentation are not indicated in the musical sources, and are left to the discretion of the performers.

We further analyzed the Clean and Original datasets by calculating the Pearson correlation coefficient between each feature in each dataset we experimented with and the composer class (Josquin or La Rue). For all features with high correlations, we manually checked to see whether the strong correlation was due to an actual meaningful musical difference or to bias introduced by the encoding methodology. For example, all the Clean pieces had a tempo of 80 BPM, and all the Simulated pieces had a tempo of 120 BPM. Thus the tempo feature alone was perfectly correlated to the class when the Simulated Josquin pieces were compared to the Clean La Rue pieces, and thus tempo even by itself perfectly distinguished Josquin from La Rue. Of course, this is in fact due to the arbitrarily chosen tempos assigned when encoding each of these two datasets, so the perfect classification performance of tempo in this example is clearly due solely to an encoding methodology inappropriately correlated to class.

6.3. Machine Learning Methodology

The features extracted from the Original, Clean, and Simulated datasets were used in several supervised 10-fold cross-validation experiments performed using the open-source Weka machine learning software [24]. In particular, Weka's SMO support vector machine implementation was used with default hyper-parameter settings. This particular configuration was chosen because it is a relatively quick-and-easy approach to use, while still being quite effective, and thus simulates what musicological researchers with only casual expertise in machine learning might do relatively easily.

6.4. Experimental Results and Analysis

Table 1 shows the classification accuracies for each dataset, averaged across cross-validation folds. In some cases, the pieces compared for each of the two composers come from the same dataset (Original, Clean, and Simulated), in order to explore the internal effectiveness of the encoding methodology used in that dataset. In other cases, the music for one composer was drawn from a different dataset than the music for the other composer, in order to simulate what one might encounter if one were to perform experiments using music that had been encoded using different methodologies.

We can see in Row 1 that the SMO algorithm was able to use the jSymbolic features to correctly distinguish between the Josquin and La Rue duos 87.0% of the time when the Clean dataset was used. This is quite impressive, given how similar the two composers are, and we can be confident that this result is not inflated by encoding bias (because of the systematically consistent way that the Clean data was encoded, and because the features were manually examined to provide additional assurance that no unanticipated bias slipped through).

In Rows 1 and 2 we can see that the Clean data performed 2.6% better than the Original data (87.0% vs. 84.4%). We can be confident that neither of these results are artificially inflated by encoding methods correlated with composers, as manual verification to guard against this was performed here as well. There are, notably, some important differences in how different pieces were encoded in the Original data; these differences are just not correlated with the composer. So, rather than causing classification to improve artificially, these encoding differences could instead deflate classification performance by injecting noise into the features. However, it should be noted that the difference in performance between Rows 1 and 2 is not large enough to be statistically significant (with a p-value of 0.05).

In Rows 4, 5, 9, and 10 we can see that classification results were grossly inflated to 100% when the Simulated data for Josquin was mixed with either the Clean or Original data for La Rue. This is because there were elements associated with instrumentation, tempo, meter, and dynamics that were strongly based on the encoding methods used rather than the underlying music, and these encodings were correlated with the composers. This confirms that, if one is not careful to avoid bias when encoding data, then one can achieve results that seem impressive but are in fact meaningless.

We can see that the Clean / Clean and Original / Original results are quite the same for the All Features (Rows 1 and 2) and Safe Features (Rows 6 and 7) groups. This makes sense, since the Safe Features omit all features that could be biased by the encoding differences in the Clean and Original groups, and the Clean group has no internal bias based on encoding source, while the Original group has no correlation between the different encoding methodologies used and the particular composers.

Row	Feature Set	Josquin Dataset	La Rue Dataset	CA (%)
1	All	Clean	Clean	87.0
2	All	Original	Original	84.4
3	All	Clean	Original	98.7
4	All	Simulated	Clean	100.0
5	All	Simulated	Original	100.0
6	Safe	Clean	Clean	87.0
7	Safe	Original	Original	84.4
8	Safe	Clean	Original	87.0
9	Safe	Simulated	Clean	100.0
10	Safe	Simulated	Original	100.0

Table 1. Classification accuracies (CA) averaged across 10 folds for each of the 2-class composer attribution experiments. Each experiment is performed once with all 246 unique features ("All Features") and once with a reduced set of 173 features chosen to be less vulnerable to encoding bias ("Safe Features"). All experiments include the same 33 secure Josquin duos and 44 secure La Rue duos, but the encodings for each vary ("Original," "Clean," or "Simulated").

There is a difference, however, between the All Features and Safe Features performance for the Clean Josquin vs. Original La Rue experiments: the 98.7% achieved by the All Features group (Row 3) was clearly inflated, but the 87% achieved by the Safe Features group (Row 8) was not (in fact, it was identical to the best real results found in the Clean Josquin vs. Clean La Rue experiment). This is because Clean Josquin vs. Original La Rue does include some differences in tempo, meter, instrumentation, rubato and dynamics that are correlated with composer in this case (Clean Josquin is uniform in these parameters, but Original La Rue is not). The All Features set is sensitive to these differences, and thus produces inflated results, but the Safe Features set filters out these problems by ignoring the composer-correlated biased quantities.

It is also notable that both the Simulated Josquin vs. Clean La Rue (Row 9) and Simulated Josquin vs. Original La Rue (Row 10) results were clearly inflated (both 100%), even for the Safe Features. This is because the Simulated encoding compressed the two distinct voices in each duo into a single voice (as a side effect of using a piano patch rather than a voice patch); although no notes were lost in this process, many features that rely on voice segregation were affected. The Safe Features did not omit such voice-linked features, so they were affected by the encoding bias. This serves as a good reminder that even "safe" features may not always be as safe as one thinks, and that cleanly and consistently encoded data is always better when available.

Of course, a reduced set of "safe" features can still be useful when one has no choice but to use data from different sources that have used different encoding methodologies. We could, for example, have made an "Extra Safe Features" group that also avoided features linked to voice segregation. The problem with being too cautious in this way, however, is that one risks omitting features that do in fact reveal musically meaningful insights. For example, examination of the feature values shows that Josquin and La Rue used voice crossing to different extents, so features related to voice crossing distinguish the two composers meaningfully; if one omits all voice-related features out of fear of biased results, then such insights will never be revealed. "Safe" feature sets must always strike a balance between security against encoding bias on the one hand and openness to musically meaningful information on the other.

6.5. Summary of Experimental Results

Using consistently and systematically encoded music can potentially play an essential role in:

- Avoiding inflated performances due to encoding biases correlated with class
- Avoiding deflated performance due to feature noise not correlated with class

Using "safe" features chosen to minimize sensitivity to encoding bias is a viable approach if one has no choice but to use data encoded in different ways, but it is inferior to using uniformly encoded data because:

- Overly cautious safe features may eliminate features that would reveal musically meaningful insights
- Insufficiently cautious safe features may admit unanticipated biases into the feature values if one does not perform careful checks to avoid this

7. CONCLUSIONS

We have established that notational consistency and encoding consistency are essential to reliable computer-aided research on Renaissance music. Our experience assembling corpora with a small team of people (including undergraduates, graduate students, post-docs, and professors) showed that establishing clear guidelines and creating templates enabled us to reach the desired level of consistency; that consistency then allows us to conduct compelling research. Our corpus, templates and workflow are available online at [6]. If other scholars adopt the same conventions for their corpora, large and small, and make them available, we will be on the path to large-scale research into Renaissance music; a composite corpus that is both varied and consistent.

8. ACKNOWLEDGEMENTS

This work was supported by the Social Sciences and Humanities Research Council of Canada and the Fonds de Recherche du Québec - Société et Culture. We would also like to thank Laura Beauchamp, Nathaniel Condit-Schultz, Néstor Nápoles López, and Ian Lorenz for their help with multiple aspects of this paper.

9. REFERENCES

[1] M. W. Beauvois, "A Statistical Analysis of the Chansons of Arnold and Hugo de Lantins," Early Music, Vol. 45, No. 4, pp. 527–543, 2017, https://doi.org/10.1093/em/cax108. [Accessed: Jun. 9, 2018].

[2] A. Brinkman, D. Shanahan and C. Sapp, "Musical Stylometry, Machine Learning and Attribution Studies: A Semi-Supervised Approach to the Works of Josquin," *Proc. of the Biennial Int. Conf. on Music Perception and Cognition*, pp. 91–97, 2016.

[3] J. Caldwell, *Editing Early Music*, Clarendon Press, Oxford, 1995.

[4] *Choral Public Domain Library*, 2018. [Online]. Available: http://www.cpdl.org/wiki/. [Accessed: Jun. 7, 2018].

[5] J. E. Cumming et al. *ELVIS Database*, 2016. [Online]. Available: https://database.elvisproject.ca/. [Accessed: Jun. 7, 2018].

[6] J. E. Cumming, C. McKay, J. Stuchbery, and I. Fujinaga, JLSDD (Josquin La Rue Secure Duos Dataset) *GitHub.com*, 2018. [Online]. Available: https://github.com/ELVIS-Project/mass-duos-corpus-josquin-larue/tree/Methodologies-for-Creating-Symbolic-Music-Corpora. [Accessed: Jun. 7, 2018].

[7] M. S. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," *Proc. of ISMIR*, pp. 637–642, Utrecht, Netherlands, 2010. [Online.] http://web.mit.edu/music21/. [Accessed: Jun. 12, 2018.}

[8] K. Desmond et al., *Measuring Polyphony: Digital Encodings of Late Medieval Music*, 2018. [Online]. Available: http://measuringpolyphony.org/. [Accessed: Jun. 7, 2018].

[9] J. Desprez, "27 Duos by Josquin Desprez or not, edited and adapted for instruments, especially recorders and keyboard instruments or harp," A. den Teuling, Ed. Assen, NL, 2014, p. 11. *IMSLP/Petrucci Music Library: Free Public Domain Sheet Music*. [Online]. Available: http://imslp.org/wiki/27_Duos_(Josquin_Desprez). [Accessed: Jun. 7, 2019].

[10] J. Desprez, *Missa de Beata Virgine: zu 4 und 5 Stimmen*, 2. Aufl., ed. Friedrich Blume, Das Chorwerk, Heft 42. Wolfenbüttel: Möseler Verlag, 1951, p. 48. [Online]. Available: http://ks.petruccimusiclibrary.org/files/imglnks/usimg/5/56/IMSLP48537-PMLP102712-Das_Chorwerk_042_-_Desprez,_Josquin_-_Missa_De_Beata_Virgine.pdf. [Accessed: Jun. 7, 2018].

[11] W. Elders, *New Josquin Edition*, vol. 4, *Masses based on Gregorian chants 2: Critical Commentary*, p. 102, Koninklijke Vereniging voor Nederlandse Muziekgeschiedenis, Amsterdam, 2000.

[12] R. Freedman, D. Fiala, R. Viglianti, and V. Besson. *Citations: The Renaissance Imitation Mass (CRIM)*. [Online]. Available: https://sites.google.com/haverford.edu/crim-project/home; https://www.dropbox.com/sh/lyka868ojjkgz12/AADa3dYzGTfqB8YMU48jbIuUa?dl=0; http://159.65.177.99:8000/pieces/. [Accessed: Jun. 7, 2018].

[13] R. Freedman and P. Vendrix, *The Lost Voices Project, 2014.* [Online]. Available: http://digitalduchemin.org/; https://www.dropbox.com/sh/f2z4iyks2fk9y1a/AAD5qJXwlYQdVC-kuPgBv3Mha?dl=0; http://digitalduchemin.org/mei/DC0407.xml. [Accessed: Jun. 7, 2018].

[14] J. Grier, *The Critical Editing of Music: History, Method, and Practice*, Cambridge Univ. Press, 1996.

[15] A. Hankinson, "Sibelius MEI Plugin," *GitHub.com*, 2017. [Online]. Available: https://github.com/music-encoding/sibmei. [Accessed: Jun. 7, 2018].

[16] D. Herremans, D. Martens, and K. Sörensen, "Composer Classification Models for Music-Theory Building," in *Computational Music Analysis*, D. Meredith, Ed. Cham: Springer International Publishing, 2016, pp. 369–392. [Online]. Available: https://www.researchgate.net/profile/Dorien_Herremans/publication/283321533_Composer_Classification_Models_for_Music-Theory_Building/links/5633449c08ae242468db84a9/Composer-Classification-Models-for-Music-Theory-Building.pdf. [Accessed: Jun. 7, 2018].

[17] P. de La Rue, *Opera Omnia*, vol. 7, *Mass Dubia*, ed. N. Davison, J. E. Kreider, T. H. Keahey, American Institute of Musicology, Neuhausen, 1998.

[18] C. McKay et al., "jSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research," *Proc. of the Int. Soc. For Music Information Retrieval Conf.*, accepted for publication, 2018.

[19] E. Parada-Cabaleiro, A. Batliner, A. Baird, and B. W. Schuller, "The SEILS Dataset: Symbolically Encoded Scores in Modern-Early Notation for Computational Musicology," *Proc. of the 18th ISMIR*, pp. 575-481, Souzhou, China, 2017. “The SEILS Dataset,” *GitHub.com*, 2017. [Online]. Available: https://github.com/SEILSdataset/SEILSdataset. [Accessed: Jun. 7, 2018].

[20] E. Ricciardi and C. S. Sapp, *Tasso in Music Project*. [Online]. Available: http://www.tassomusic.org/. [Accessed: Jun. 7, 2018].

[21] J. Rodin and C. S. Sapp, *Josquin Research Project*. [Online]. Available: http://josquin.stanford.edu/; http://josquin.stanford.edu/about/attribution/. [Accessed: Jun. 7, 2018].

[22] P. Vendrix et al. *Gesualdo Online*. [Online]. Available: https://ricercar.gesualdo-online.cesr.univ-tours.fr/. [Accessed: Jun. 7, 2018].

[23] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufman, New York, 2011.