Music Encoding Conference Proceedings

2020

26-29 May, 2020

Tufts University, Boston (USA)

Edited by

Elsa De Luca and Julia Flanders

Keynote I

Articles

Posters

Keynote II

Foreword

Richard Freedman
Haverford College
rfreedma@haverford.edu

Anna E. Kijas
Tufts University
Anna.Kijas@tufts.edu

The 2020 Music Encoding Conference was both the biggest and strangest ever. Originally planned to be held on the campus of Tufts University from May 26 to May 29, 2020, we were obliged by the COVID-19 public health crisis to move everything to the digital realm. Members of the organizing, program and local arrangements committees imagined ways to host workshops, interest groups, town hall meetings online, papers, posters, and keynote talks digitally using Zoom and Slack. Such forums can never replace face-to-face interaction. But the music encoding community–always remarkable for its sense of openness and professionalism–was ready for the challenge. And so our meeting was no less successful than it was curious or large (with over 250 registrants, it was twice the size of any other of our meetings).

We are pleased to offer the collected papers and posters prepared for MEC 2020 as they circulated in advance of the virtual event. We append to those the written versions of the keynote talks by Dr. Timothy Duguid and Dr. Estelle Joubert. Together, these documents represent the rich array of practices and thinking embraced by the music encoding community, from ways of thinking about images, editions, and performance, to new approaches to analysis and encoding.

We are grateful to the many hands who contributed to this work, namely Anna Kijas, Julie-Ann Bryson, Sarah Connell, Julia Flanders, Jessica Fulkerson, Johannes Kepper, Elsa De Luca, Vincent Besson, Margrethe Bue, Joy Calico, Stefan Münnich, Anna Plaksin, David Weigl, and Irmlind Capelle.

The forgotten classroom? Bringing music encoding to a new generation

Timothy Duguid
University of Glasgow
tim.duguid@glasgow.ac.uk

Abstract

Digital methods have begun to make their way into the research practices of music scholars, and most this insurgence can be attributed to the rise of the discipline of music technology. Though music encoding is becoming increasingly prevalent among the research and teaching methodologies of music scholars, evidence gathered from course descriptions and presentations at national meetings of music scholars would indicate that encoding continues to lag other music-based technologies. Drawing from the advancement of music technology and the experiences of digital humanities teaching and scholarship, this paper presents a path for the music encoding community to promote greater integration of encoding and digital methods more broadly into the pedagogical practices of music historians and music theorists.

Introduction

> How do we teach music encoding? Do we profess it? Do we profess to teach it? Or, do we teach (courses like encoding and computer-assisted analysis) so that we might profess (our scholarly understanding of digital musicology as the intersection of musicology and computing)? However seemingly simple the question "what do we do?" may be, we do a disservice to our field and ourselves if we fail to consider the importance of pedagogy when it comes to answering such questions, no matter how commonsensical they might at first appear.[1]

This is a modified quote from Brett Hirsch in which his references to digital humanities have been replaced with references to music. Just as these questions helped frame a budding reemphasis on pedagogy within the digital humanities community in 2012, they are helpful to the music encoding community as it weighs the proper use of music encoding within the classroom. As Hirsch notes, some discussions of pedagogy seem pedestrian and as he says 'commonsensical'. Nevertheless, they are foundational for establishing a pedagogy for music encoding. Indeed, just as current music encoding tools and methodologies had to start from scratch, so also the associated pedagogical strategies for incorporating these digital research methods must start at the most fundamental levels.

And yet, the fundamental nature of these questions belies their complexity. It would be quite bold for any one person to claim to sufficiently answer these questions. After all, as Sean Michael Morris states, "Pedagogy has at its core timelines, mindfulness, and improvisation. Pedagogy concerns itself with the instantaneous, momentary, vital exchange that takes place in order for learning to happen" [2]. Like in improvisation, the pedagogue is constantly adapting to the audience, to the subject, and to the goals of the performance. Although this might appeal to some, the classroom is not a formula by which all students will learn if the instructor follows it. And yet, it is tempting to approach pedagogical practices in this way. Perhaps this is just my background as the son of a carpenter, but one tool will not allow you to build a house. Indeed, I spent many summers as a gofer for my father, crawling in his van to find that one tool among the hundreds that would get a specific job done. So too must pedagogues build and rely upon a set of tools that will facilitate learning depending on the situation. My presentation today is therefore not going to answer the questions outlined

1 Hirsch's original quote is "...do we teach digital humanities? Do we profess it? Do we profess to teach it? Or, do we teach (courses like computer-assisted text analysis and others surveyed in this collection and beyond) so that we might profess (our scholarly understanding of the digital humanities as the intersection of humanities and computing)? However seemingly simple the question 'what do we do?' may be, we do a disservice to our field and ourselves if we fail to consider the importance of pedagogy when it comes to answering such questions, no matter how commonsensical they might at first appear." (emphasis original) [1, pp. 16-17].

at the start, but rather to foster discussions of these questions by presenting a couple of tools to add to our collection of strategies for incorporating music encoding and other digital methods into music classrooms.

Digital pedagogy?

What does "digital pedagogy" mean? Like its parent, digital humanities, this term has been widely discussed across the humanities with little resolution. Simply breaking the term into its constituent parts, Morris describes pedagogy as "...a scholarship unto itself, a study of learning and the many ways it is fueled - in classrooms, in workshops, in studios, in writing centers - wherever learning is poised to occur" [2]. While Brian Croxall and Adeline Koh have likened the digital to that which consists of "electrical elements" [3] I think music scholars require a more precise definition, particularly considering our history with analogue electronic devices such as oscilloscopes, analogue synthesizers, and microphones (just to name a few). So, I turn to the definition from the OED, which states that digital refers to "signals, information, or data: represented by a series of discrete values (commonly the numbers 0 and 1), typically for electronic storage or processing" [4]. We can then infer that "digital pedagogy" is the study of the processes by which learning occurs either in or as a result of the electronic storage or processing of discrete values.

This is an admittedly wide umbrella that may leave many uneasy about the sorts of learning and activities it could include. In many ways, such a broad definition harkens to the unease many digital humanists feel when someone asserts that doing 'digital research' involves simply reading an article online or publishing in an e-Journal. Indeed, at the popularization of Learning Management Systems such as Blackboard and Moodle, many were happily convinced that digital pedagogy simply meant offering a course online. As Morris quips, digital pedagogy "was easy...a mere work of relocation" [2]. In one sense, this view is correct: digital technologies have been used to teach the subject at hand. However, limiting digital pedagogy to posting slides or lecture recordings online hamstrings the types of resources and capacities that the digital affords. Within digital humanities pedagogy, there has been a trend away from the types of sterile and static pedagogical practices that simply transfer existing "analogue" content and methods online and towards more active, student-centered approaches that emphasize collaboration, hacking, process, and construction and that actively bring cutting-edge research into the heart of the classroom. In this regard, Morris's definition of pedagogy is particularly helpful. For within true digital pedagogy there are continuous acts of refinement: learning from the digital approaches that have or have not worked in the past in an effort to improve and enhance the learning experience.

How did we get here?

Pedagogy has long been at the heart of humanities computing. Workshops such as the "Teaching Computers and the Humanities" series sponsored by the Association for Computers and the Humanities, as well as the Computers and Teaching in the Humanities conference provide some early examples. Moreover, the 1980s and 90s saw the establishment of dedicated digital humanities centers such as the Center for Computing in the Humanities at the University of Toronto, the Centre for Computing in the Humanities (now the Department of Digital Humanities) at King's College London, the Institute for Advanced Technology in the Humanities at the University of Virginia, and the Humanities Advanced Technology and Information Institute at the University of Glasgow (now the Department of Information Studies). But, despite the efforts of these and the establishment of initiatives such as the Digital Humanities Summer Institute at the University of Victoria, pedagogy was sidelined in public discourses through much of the first decade of the 2000s. Whether this occurred as a result of funding availability or other external pressures, research methods garnered the collective attentions of both scholars and benefactors. As Hirsch recalls, Donald Bruce's plenary presentation at the 2009 Digital Humanities Summer Institute highlighted this growing imbalance, something Hirsch labels "bracketing", and the community began to take note. By the end of 2011, the Digital Humanities at Oxford Summer School had been started, the first THATCamp Pedagogy had been held, and two roundtable sessions focused on digital pedagogies had been accepted for the 2012 annual meeting of the Modern Languages Association in Seattle [1, pp. 3-5].

Since that time, pedagogy has become a central concern of the digital humanities community. Since 2011, for example, the National Endowment for the Humanities has approved 63 different grants totaling over $8 million that develop digital teaching resources and pedagogical methodologies.[2] In the same period, the Mellon Foundation has invested over $3.8 million across 7 different grants that are similarly focused.[3] The literature on digital pedagogies has also grown significantly since 2011. The volumes of the *Debates in the Digital Humanities* series, and *Hacking the Academy* have devoted numerous chapters to the topic, and journals such as *Digital Humanities Quarterly* and *Digital Scholarship in the Humanities* as well as blogging platforms such as *Hybrid Pedagogy* have devoted significant space to digital pedagogy.

Music also has a sustained history in pedagogical practice and research, and it has a similarly long history with technology. I'll not rehearse what are well-known stories such as Johann Sebastian Bach's widely varied education in music performance and composition or Edison's invention of the cylinder phonograph. However, it is interesting that music pedagogy and technology became somewhat estranged in the twentieth century. Reporting on the state of higher education institutions in the United Kingdom in 2007, Carola Boehm traced the history of music technology through five generations of researchers and innovators. The first generation, labelled the Experimenters and Innovators, includes Schaeffer, Stockhausen, Eimert and Cage, among others. Then came the "Commercializers" in the 1970s and 80s such as Boulez, Vercoe, Wishart, and Puckett, who first began to teach music technologies in the classroom and who began to market technologies widely. This gave rise to the third generation, the 'First Lecturers' in the 1990s and 2000s, who seeing the rise in affordable digital audio equipment wanted to provide training for enthusiasts. Boehm's fourth generation was therefore one in formation when she wrote, as it included those who were then graduating from newly constructed degree programs in music technology. Finally, the fifth generation was one she projected would move on to graduate-level education in the 20-teens. Despite this optimism, she still concluded that music technology remained the discipline that "never was" [5]. Boehm has since published a reappraisal of music technology within the U.K., conceiving of a sixth generation in which music technology has been cemented as an academic field, with the fourth and fifth generations having begun to have an impact on the industry [6].

Wanting to compare those findings with current pedagogical practices in the United States, I conducted a survey of more than 60 of the country's leading music schools. After exploring the undergraduate and graduate course catalogues of each of these institutions, I found that all of the institutions offer technology-related courses to their students. Although I have only looked closely at schools in the U.K. and U.S, I daresay that one would find similar results in other countries around the world. One could conclude, therefore, that digital pedagogy is well in-hand throughout music schools today.

However, a closer look at the course descriptions of the same group of U.S. institutions reveals another story. Given the emergence of 'maker culture,' it is unsurprising that many institutions are now offering courses on digital music recording, music synthesis technologies, sound production, music distribution and marketing, and multimedia integration and alignment (including audio in video, film and video games). One might even add music notation software to that mix, particularly given the divergent idiosyncrasies of LilyPond, Finale, Sibelius, MuseScore, etc.[4] When I remove these courses from the list, in other words, looking for course descriptions in which digital humanities-related research methods are mentioned (i.e. optical music recognition, notation encoding; GIS; score-media alignment; metadata generation and curation; network analysis; and computer-aided distant reading of corpora, just to name a few), that list is whittled down to just 20 courses, and that generously includes the courses on notation that purport to include the latest developments in digital music notation that may or may not include music encoding.[5] If those notation courses are removed from the list, the number is cut in half. Across more than 60 of the most reputed music institutions of higher education in the United States, only 10 course descriptions could be found that use these methods. While it should be

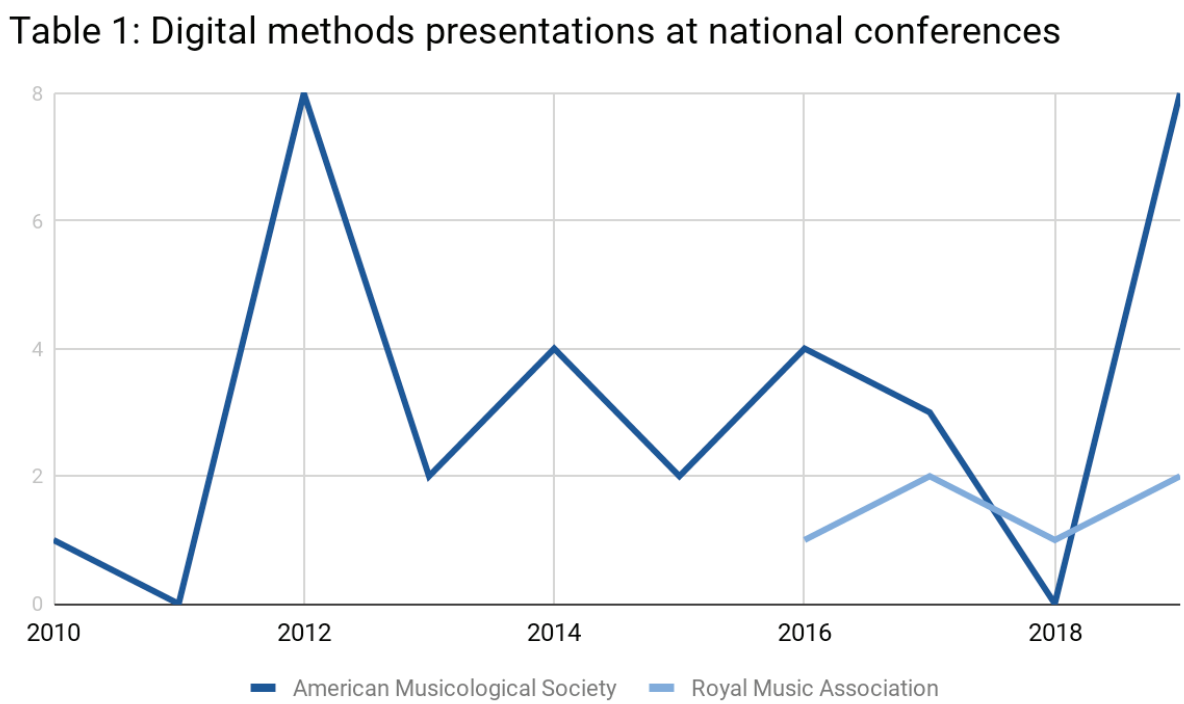
2 National Endowment for the Humanities, "Funded Projects Query Form," neh.gov, https://securegrants.neh.gov/publicquery/main.aspx (accessed 22 May 2020).

3 The Andrew W. Mellon Foundation, "Grants Database," mellon.org, https://mellon.org/grants/grants-database/ (accessed 22 May 2020).

4 Although the issues with these software packages are well documented, Martin Keary's reviews provide some representative examples of this criticism. Martin Keary, "Tantacrul", YouTube channel, https://www.youtube.com/user/martinthekearykid.

5 As a side note, this list of methods excluded courses that utilized image-based collections and repositories. While beneficial to music teaching and research, there is little computational difference between their utilization and that of PDFs or even hard copies of notated music.

noted how infrequently course descriptions are updated and that they cannot be expected to include all that a particular course might cover, this is symptomatic of the state of today's music academy, and particularly in the core areas of music history, literature, and theory.



As another example, consider the annual meetings of the American Musicological Society and the Royal Music Association (Table 1). Looking at the published abstracts for the AMS dating back to 2010 and the RMA back to 2016 (earlier ones are not available on their website), a similar pattern emerges. The AMS has twice featured 8 papers, posters, or roundtables that include digital methods in their abstracts: in 2012 and 2019. However, these two years were significant outliers, as the remainder have featured between 0 and 4 presentations. Even if one accepts that some presentations may have been excluded from these counts because their abstracts do not mention any digital methods, the overall percentage remains paltry considering how large the conference is. For instance, 2019 featured more than 380 different presentations, which means that only 2% included digital methods. The Royal Music Association is not any better, as 2017 and 2019 were the high-water marks, featuring only 2 presentations that mentioned digital methods. This all points to an absence of digital methods from research workflows of historical musicologists, or at least the workflows of those considered within the mainstream of their respective disciplines. Indeed, if faculty are not engaging with these methods in their own research, they are not likely to teach them to their students.

On the contrary, emerging areas such as music recording, sound production, and electroacoustics - those fields commonly included under the umbrella of music technology- have largely adopted digital methods in their research and pedagogical workflows. Even applied musical instruction has begun to incorporate more digital resources, as more and more apps are being built to provide access to sheet music, to record practice or performance, and for immediate analysis for those performances. Sadly, music history, literature, and theory have not been so quick to adopt digital methods in either research or teaching. Indeed, based on my findings regarding course offerings and research paper presentations, musicologists and music theory scholars seem to relegate digital methods to research on twentieth- and twenty-first-century music, in essence where digital media already exists. They are much less likely to employ digital methods for music composed before 1900.

This is not to say that musicologists and theorists are unaware of the developments in these other areas, nor are they ignorant of the goings-on in the digital humanities. A survey conducted by Inskip and Wiering in 2015 would indicate that a lack of freely available digital data is one of the largest barriers to widespread implementation of digital research methods. However, it is also true that specialists in music before the twentieth century are often unaware of the latest technologies and therefore how their research could benefit from digital methods. Additionally, a large number are generally uneasy about computers - after all, they argue, learning how to use Finale and Sibelius was traumatic enough! [7] Regardless of the reasons, students continue to pass through theory, literature and history curricula thinking that the cutting edge in these fields remains closely tied to analogue outputs or digital recordings. Looking at it another way, and a more superficial way, compare the 'toys' of musicologists and theorists with the 'toys' of other music scholars. The former has books, journals, eBooks, recordings, and PDFs along with instruments of varying types. The latter has mixers, synthesizers, loudspeakers, microphones, streaming services, and computer algorithms.

So, what is the music encoding community to do? Over the years, this community has frequently engaged in discussions, both internally and externally with other like-minded groups, strategizing methods to promote music encoding and the various capacities it affords. Any attempt to list these efforts would be incomplete and do a disservice to those not mentioned. However, engaging with these scholarly communities at their annual meetings have had positive effects. In addition, pedagogical efforts such as the digital methods workshops hosted at various conferences and intensive summer schools around the world have provided hands-on opportunities for researchers to learn and interact with music encoding practices. These corporate efforts add to the numerous individual conversations that our members all have had with those in our own respective institutions. Of course, these should all continue, but I would argue that an increased emphasis on incorporating these into undergraduate and graduate-level instruction is a critical step in transforming the discipline. Following Boehm's outline, one could argue that music encoding may only be in its second or third generation, so now is the time to start incorporating it into the classroom.

In formulating a strategy for incorporating digital research methods such as music encoding into course curricula, the experiences of colleagues in the digital humanities are instructive. As mentioned earlier, pedagogy was not a significant focus of the digital humanities in the 2000s, and when that began to change in the early 20-teens, the initial assessments of digital humanities pedagogy were that it was widely varied. On the one hand, researchers were simply teaching students based on their own research and methods, which of course vary from project to project and person to person. On the other, it undoubtedly confused many students who were trying to figure out what this "digital humanities" thing was (incidentally, something that practitioners themselves still have difficulty defining). However, the field has begun to coalesce, leading Deborah Garwood and Alex Poole to conclude that "DH pedagogy inspires students and faculty members to critically, openly, collaboratively, collectively and symbiotically to explore existing or to carve out new research and scholarly areas across disciplines" [8, p. 552]. The same could be said for digital pedagogy in music-related studies: it should inspire students and faculty to critically, openly, collaboratively and collectively explore existing scholarship and establish new areas of inquiry that are not necessarily limited by disciplinary boundaries.

Music encoding in the classroom

There are a number of tactics that one could employ in tackling the issue of digital pedagogy. Some, like Claire Battershill and Shawna Ross in their recent monograph *Using Digital Humanities in the Classroom*, discuss the barriers that have been constructed against the incorporation of digital methods in the classroom, categorizing them according to the source: that is as coming from the instructor, students, and colleagues [9, pp. 13-24]. Within the context of a monograph acting as a practical guide to incorporating well-established pedagogical methods into classroom environments, such an organization makes sense. However, music pedagogues are not so fortunate in having tried and tested methods for incorporating digital methods into music classrooms, and particularly music history and music theory classrooms. Therefore, the remaining discussion is going to be more topical, exploring the issues of audience and managing stress and chaos, before concluding with a couple of skills that should be included in digital curricula.

Audience-appropriate content

Modern society is fixated on audiences, customers, and even students. While one might argue that this has its drawbacks, considering one's audience does help to provide helpful perspectives from a pedagogical point of view. Student-centered teaching strategies have become quite popular in the past couple of decades, but a challenge to digital pedagogies appears when the instructor gets so excited about a newly discovered or developed tool or digital method. In their enthusiasm, the instructor forgets why the students are sitting there in that lecture theatre, and the class becomes a lesson in a tangentially related digital tool rather than the original subject. Regardless of whether said instructor is excited about a tool, a digital method, or some minutia of digital humanities theory, Ryan Cordell boldly asserts, "undergraduate students do not care about digital humanities," and he continues "most graduate students...do not come to graduate school primarily invested in becoming 'digital humanists'" [10]. His comments could also be applied to music students: most have intentionally chosen to avoid computer science and mathematics. One could take this one step further. There was a pervasive theory in pedagogical writing around the turn of the century that students were "digital natives" and were therefore more comfortable with and competent in all activities relating to computers. However, as Brandon Locke comments, " Students are often much less adept at creating content that is not tightly mediated by some kind of commercial service with restrictions on form (e.g. Snapchat, Twitter, Facebook)" [11]. Students are therefore just as reticent as other generations when it comes to angle brackets and curly braces. Indeed, despite the "digital natives" moniker that sadly still surfaces in the pedagogical literature, it is important to remember that many music students will not have the inbuilt, innate, or otherwise preexisting familiarity with or comfort with music encoding or code-based analysis tools. Nor do they necessarily want to spend significant time learning how to code and encode.

When developing course content that utilizes digital methods, one should therefore consider the students' skill levels at entry and the desired results once they complete the course.

As an illustration, I point to a course I teach at Glasgow called Music Curation and Analytics, which is offered to upper-level undergraduates in Information Studies. Most of these students are not music students (and one would assume intentionally so, since they are studying information studies and not music). The first year I taught the course, I had them transcribe a piece of music in MuseScore and then export it to musicXML and on to MEI before they then edited the MEI file. The idea was that they would gain experience in understanding each format. Since the students already had a level of XML training, I figured that they would be able to handle the MEI modification. For students who had a background in music, this task was not too onerous, but others really struggled with the transcription in MuseScore - despite me providing a basic introduction to reading Western music notation - because they remained too unfamiliar with music terminology and therefore spent much of the semester trying to transcribe their piece, let alone considering what changes could be made to the MEI. In the second year, I focused less on the specifics of music notation and more on the comparisons between the MusicXML file and the MEI file, describing the differences and what those meant both semantically and in terms of the capabilities of both formats. Students did much better with this approach, given their existing background in XML. Indeed, this latter approach was much more attuned to the course objectives, which were to introduce students to the ways in which music-related information is created, stored, analyzed and otherwise reused.

Managing stress and chaos

Despite this anecdote, some outside this community (and perhaps some within it) might argue that music encoding is too new, and its accompanying toolset too underdeveloped to be presented in the classroom. Those promoting this view might worry that students could be overwhelmed and frustrated by complicated software installations and tools that frequently "break" or do not perform as expected. On the one hand, this risk can be reduced by limiting student expectations of the technology. For instance, MEI rolled-out version 4.0 while I was in the middle of teaching music encoding to a group of masters students. As you may be aware, version 4.0 involved significant changes to the way metadata was captured in the meiHead element, and this impacted some of the validation functionality afforded by plugins to Atom. However, at the beginning of the course, several weeks in advance of the release, I had mentioned that MEI is a community-based standard for encoding music notation and that those standards can change to adapt to meet the needs of the community. The students were therefore much more flexible in their expectations of the technology. Rather than causing significant upheaval in the middle of the class, the update in MEI versions offered us the opportunity to explore the new guidelines and to learn from them together. We were able to discuss the changes and to consider the semantic impacts of those changes. This is a relatively tame example, but there are others in which something may actually fail. Indeed, Katherine Harris goes so far as to insist that students *will* break digital tools [12, p. 21]. As Lisa Sprio notes, however, "…the digital humanities community recognizes the value of failure in the pursuit of innovation…since it indicates that the experiment was likely high risk and means that we collectively learn from failure rather than reproducing it (assuming the failure is documented)" [13]. Indeed, students should not be completely shielded from unsuccessful results. Rather, they should be trained in ways to document them and to learn from them.

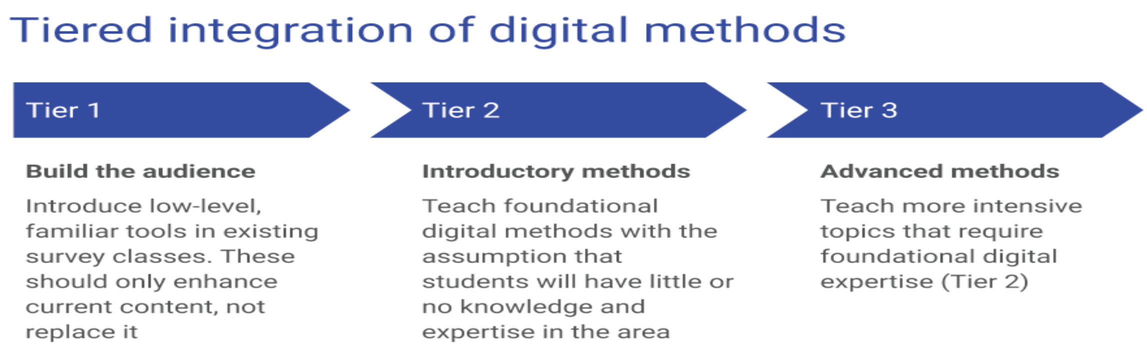


Figure 1: Proposed integration of digital methods into music curricula

Beyond turning these challenges and even failures into positive learning experiences, the music encoding community can recommend systemic controls that effectively would limit students' potential exposure to frustrating results until they have reached a point at which they can either troubleshoot them or can properly contextualize their experience. The music encoding community therefore needs a coordinated progressive strategy for introducing digital methods into music history, literature and theory curricula (as that suggested in Figure 1). Of course, tiered approaches to curricula are nothing new to music pedagogues who teach a broad range of courses from music appreciation to advanced Schenkerian analysis. However, the same pedagogues may not have considered that a similar approach is required for digital methods. Given the general reticence that many music students have towards computers, digital pedagogues need to start with some simple digital discovery before throwing students into the world of angle brackets and curly braces. That is, show them the utility and capabilities that digital methods afford. This is the step that many instructors missed in the early days of the digital humanities. In the early days of the digital humanities, instructors rushed to create survey courses, forgetting that students first needed to be shown why DH was important and how it could positively benefit their studies and research. As Cordell notes [10], students and colleagues are more receptive to digital methods when they were integrated into a course that they already deemed relevant to their studies. Indeed, this is what Adeline Koh also describes, as she encourages instructors to employ the tools with which students are most familiar (i.e. Google Maps, Wikipedia, etc.) before delving into more complicated elements [14]. Music teaching should therefore start with simple tools that are integrated into survey curricula to provide data-intensive illustrations of the overarching concepts that are being taught. At this level, it is critical that the expertise and training for the digital resource should be minimal, so it does not overshadow the subject-specific training. Jonathan Howell provides a good illustration of the balance required at this level. He describes how he created a linguistics course that relied heavily on R, but that his students struggled to keep up with both the programming requirements of the course and the linguistics content. Before offering the course a second time, he built a web application that allowed his students to take advantage of analytical tools offered by R without requiring them to know how to code in R. The result was a much better student experience that recognized the benefits of digital approaches within the context of linguistic research [15]. Resources such as the Verovio Online Editor and jSymbolic could be incorporated in this same way because they do not require significant coding expertise at the outset. However, music pedagogy would benefit from more of these types of low-level digital tools that allow students to start familiarizing themselves with digital methods.

There are, of course, limitations to digital tools, as Locke argues, "Tool-based literacy limits sustainability, cross-platform work, and understanding of the impact of media upon the message" [11]. It is therefore important for curricula to build on the initial introductions that occur in the first tier with both surveys of digital methods and more focused digital training to provide much-needed critical skills to evaluate those digital methods. Although it is not a degree-based curriculum, I would argue that the offerings of the Digital Humanities Summer Institute (DHSI) are a helpful exemplar. Begun in 2000, DHSI provides intensive training in the digital humanities. It offers over 50 different one-week courses over a two-week period in June that cover a broad range of topics relating to DH research and pedagogical practices. Much like other digital humanities summer schools, DHSI operates on the assumption that its students have already encountered digital methods within their coursework, research, or teaching. This digital first contact has the DHSI student itching to learn more, but that person may not have any level of technical expertise. DHSI therefore offers a number of "Foundations" courses that provide entry-level surveys of digital methods and training in courses such as TEI, DH technologies, introductory computation, digitization, and even music encoding.[6] I would argue that these types of courses are the logical second step in a tiered digital curriculum. For degree-based music instruction, this could include introductions to music encoding in which students actually start encoding music using various standards. It could also include basic introductions to computational analysis of musical content. The key is that these courses should effectively build from the ground up, that is, they should start with the assumption that students have little or no expertise in that particular area.

The third and final step in this tiered approach involves offering much more advanced courses in digital methods that require a certain level of expertise at the outset. These courses may explore the areas of computer learning, analytical methods in python or R, or even combinations of digital methods, and often these courses are much more focused in terms of their musical remit. For instance, one could envision a course on computational stylistic analyses of Stravinsky's oeuvre.[7]

Skills development

Having outlined this hierarchy, it is important to consider what topics are fundamental to the discipline as it moves towards digital research methodologies, and which are less crucial. Given the widely varied and changing state of digital methodologies in music, I would not pretend to offer such a hierarchy on my own here. That said, I would suggest two important skill sets that should be included.

Digital literacy

Despite the increased use of digital pedagogies, Locke comments, "there should be reason for concern that students are often taking part in digital information and media transmission, but are not currently trained in the literacies and affordances of the technology they use" [11]. Indeed, it is almost cliché that every course today claims to instill in students critical thinking skills, but this can be very difficult to achieve in a single course. I would argue that if music teachers continue to make these claims, particularly for history and theory curricula, there needs to be a reevaluation of how students in the digital age can be trained in critical thinking so that it approaches what Locke and others would label digital literacy. Although students are accustomed to taking surveys and to providing reviews of their meals and shopping experiences, it can be difficult to encourage them to think outside their own experience and particularly about the strengths and weaknesses of those digital methods and the resulting limitations of the data they produce. I would argue that there are four components to digital critical evaluation. To illustrate the first two, permit me a brief excursus.

6 For a list of courses, see "Course Offerings," *Digital Humanities Summer Institute* (DHSI), https://dhsi.org/course-offerings/ (accessed 22 May 2018).

7 A similar hierarchical structuring of instruction is proposed by [10].

Nestled in the hills of Western Pennsylvania, is a small city called Beaver Falls. Known as the hometown of American Football Hall-of-Famer Joe Namath and the setting of the 1980s TV show Alf, Beaver Falls is also home to a small liberal arts school called Geneva College. As an alumnus of Geneva, I could regale you with some of its historical claims to fame, which include participating in the Underground Railroad during the American Civil War, as well as claiming to have played the first men's college basketball game in 1893. However, my reason for mentioning Geneva in this context is not for one of these claims to fame but rather for what some might consider to be a mundane architectural feature: a bridge at the edge of campus that crosses some 50 feet (15.25 meters) above the Beaver River connecting Beaver Falls to the small township of Eastvale. This was the site of an interesting experiment that did not result in a discipline-changing discovery but rather an experiment that epitomizes the learning experience.

A personal friend and Geneva alumnus told me of one of his experiences as a student there. During one of his summer vacations, he worked as a lab assistant for one of the chemistry professors. This meant that he and another student were tasked with preparing the labs for the upcoming autumn term. They cleaned the labs and their equipment; took inventory; and disposed of, ordered and received new equipment and supplies. One day, he and the other lab assistant came across a substantial container of sodium that needed to be disposed of. This was back in the 1960s, and what else were two college students to do with a bucket of sodium? Of course, let's take it down to the Eastvale Bridge and heave it over the side to see what happens! According to my friend, the result was quite spectacular, resulting in a jet of water that shot up onto the bridge and the vehicles crossing it.

Looking back on the situation, said alumnus admitted that it was probably not the safest or smartest thing to do. However, it illustrates two elements that I think are critical to education: knowledge and play. The two students knew of sodium's reactivity with water, and they were willing (admittedly unadvisedly) to apply that knowledge to "see what happens". And, given the fact that my friend told the story with a smirk on his face some forty years later would indicate that he has never forgotten about the violent reaction that can occur when sodium comes into contact with water. I would therefore argue that first and foremost, students need to have the requisite subject knowledge to be able to contextualize information. Then students should be afforded the opportunity to apply that knowledge while playing with specific digital tools. This approach to digital pedagogy is well established across the sciences and humanities, as is chronicled by Jentery Sayers [16]. Despite the benefits of allowing students the space to play with digital tools and methods, Nuria Garcia, *et al* caution that the digital sandboxes established for classrooms need to have boundaries, arguing "The goal in the college classroom should not be to allow for open-ended digital play and exploration of the kind that professional humanities scholars are motivated to undertake, because as one learner noted, ,the amount of information can truly be overwhelming, and a large part of the success of this exercise seems to lie in not only how to use the [digital] tools to the best advantage, but in...avoiding dead-ends" [17].

Even if students are afforded the space to tinker with digital tools, they often lack the ability to understand the raw data they are gathering, particularly if it is quantitative data. As Jonathan Howell argues, "...quantitative literacy ought not to be regarded by the instructor in a non-STEM field as an add-on to existing course content, but ideally as an integral part of teaching students how to be a historian/anthropologist/classicist/etc" [18, p. 16]. The past 3-4 months have provided an instructive illustration of the dangers of quantitative illiteracy if one is willing to look. The COVID-19 outbreak has provided an unparalleled (I refuse to use the word "unprecedented", given its overuse and abuse lately) deluge of quantitative data for public consumption. There have been daily updates of test rates, positive test results, negative test results, hospital admission statistics, ICU admission statistics, daily deaths with COVID-19 listed as a potential cause, deaths of people who had previously tested positive for COVID-19, care home deaths, and now "R-numbers." Despite all this raw data, it has been painfully obvious that many (including the media and politicians) are ill-equipped to parse the numbers and to understand what the numbers mean and what they do not mean. Similarly, as quantitative analyses become increasingly present in musical analysis, it is important for the field to consider how it can teach students how to value these analytical techniques and the data they generate, evaluating the assumptions inherent in the methods and tools and thereby critically evaluating the conclusions that result.

Moreover, focusing solely on digital and quantitative methods provides students with a limited scope and therefore hampers their ability to critically evaluate those methods. As suggested by Paul Fyfe the combination of analogue and digital methodologies gives students the requisite space for critical observation. In a class on *Pride and Prejudice*, Fyfe comments, „Unplugging the search engine can help students perceive the limitations as well as the possibilities of what makes these engines run: pattern matching, which by itself is a far cry from reading at any distance. It sharpens students' attention to forms of analysis that explore the analog and digital domains along a continuum. It helps students to interrogate the various kinds of readings they can do therein. And it reveals all of those kinds of readings as actively constituting critical interpretations" [19]. Critical evaluation of digital tools, resources, and methods - even such as music encoding - require students first to have discipline-specific knowledge of music. They then should be trained in how to encode that music before they are given space to play around with various approaches to encoding music. Whether or not quantitative methods have been used, the students need training to illuminate the strengths and weaknesses of the encoding techniques they have employed. Finally students need to be able to compare these digital methods with analogue versions of the same.

Collaboration

In addition to digital literacy, digital pedagogies in music should include skills in collaboration. This may be an area of discomfort for many music theory scholars and musicologists, who, as noted by Kris Shaffer, prefer working in isolation [20]. However, one of the hallmarks of the digital humanities has been the promotion of collaborative research. Digital humanists freely recognize that no one person possesses the requisite skills and knowledge to produce a high-quality digital resource. Students should therefore be confronted with this reality: they may not be able to master all things musical while also trying to master all things digital. They should therefore be encouraged to specialize and then to collaborate with those with complementary specialties.

Even so, as Rebecca Frost Davis asked, "..but how do you *teach* collaboration?". This question has been problematic in DH pedagogy, particularly in terms of assigning credit in assessments. Recognizing the potential inequity of assigning all group participants the same grade regardless of their contribution level, some have innovated systems of assessing each person according to their contribution to the group's final output.

While I do not pretend to have solved the issue, I have found one method that works with my Music Curation and Analytics students while avoiding some common pitfalls. From the beginning, I was confronted with the reality that most of my students do not know how to read Western music notation and that I did not have the time to provide significant training in this while also covering aspects of encoding and curating notation data. Two other facts were also clear to me as I was planning this course. First, students rarely invest the amount of time outside of class that the University recommends they do (for humanities, 9 hours of prep for every hour spent in class). Second, students are often frustrated by graded group projects because of the inequalities that often surface. My solution was to have a scheduled session at the beginning of each week during which students have structured time to prepare for the week's lecture. During that period, they were given a brief introduction to the week's topic, and then they were asked to "play" together in groups, trying to accomplish some set tasks that are unassessed. The following day we discussed their group work during the lecture. This was then followed by a lab period in which the students were individually assigned an assessed task that builds on that week's group activity and lecture. During the first week's group session, I told the students that they could form their own groups, but I made sure that each group had at least one person who could read music. For the tasks relating to music notation (i.e. using MuseScore to transcribe a piece of music or encoding a piece to MEI), the person who could read music was asked to assist those who could not. This approach to group work was largely successful, as by the end of the semester the students were working well together not only on the group activities but also on their individual assignments. In fact, several of the students remarked that the group session helped them to better understand both lecture content and to be better prepared for the assessments.

Conclusion

Imagine a situation in which a music theory instructor is teaching about chord progressions, and asserts that an Authentic Cadence is the most common and most authoritative way to end a piece of tonal music. Immediately a student shouts, “Prove it!” I daresay the vast majority of instructors today would not be able to prove it, even though they might be able to point to some important examples, While complete proof might be outside our grasp (particularly considering how little music throughout history has been preserved), it is well within the realm of possibility that said instructor could run a quick script on a large corpus of music and show said student that an Authentic Cadence is indeed most prevalent. At the same time, however, said instructor could simultaneously discover that a VI-I cadence is also common in a certain group of pieces, which then could provide an avenue of investigation for both the instructor and the class. However fantastical this story may seem, situations like this arise on a regular basis within digital humanities classrooms around the world, even if on a smaller scale. With training and a strategic approach to digital methods implementation, the same could be true for music classrooms.

Some historical musicologists or music theory scholars might recoil at what has been presented here as too statistical or at least too unsettling and computer dependent. After all, much of what I have advocated here requires a reconsideration of the ways in which we approach music history, literature, and theory instruction, even at the most fundamental levels. And yet, the music encoding community offers a supportive atmosphere for those who want to incorporate encoding into their research workflows. As this community continues to grow and as music encoding continues to become more prevalent in research methodologies, we must consider the future and particularly how these methodologies can be passed on to the next generation of researchers. So, while communities such as ours may not be able to realize a change in music history or music theory curricula by ourselves, we can encourage those respective communities to update and expand their methodologies. Indeed, we can continue to promote the latest innovations in digital methodologies at national meetings and focused workshops, and thereby continue to highlight the benefits of employing digital methods within those respective fields. We can also start developing hierarchies of digital pedagogy as guides to both professional societies and individual departments for incorporating digital methodologies into their curricula. Finally, as you “go out” to your institutions (I am speaking in the digital sense since we remain in our homes for this conference), consider how you could either start incorporating music encoding and digital methods into your classes or alternatively how you might encourage your colleagues to do so. Indeed, by promoting best practices in both research and teaching as a collective, we can, like Boehm, look ahead to our own fourth, fifth, and sixth generations of music encoders and the exciting innovations that will accompany them.

Works cited

[1] Hirsch, Brett D. “</Parentheses>: Digital Humanities and the Place of Pedagogy” in *Digital Humanities Pedagogy: Practices, Principles and Politics,* ed. Brett D. Hirsch. Open Book Publishers, 2012.

[2] Morris, Michael. “Decoding Digital Pedagogy, pt. 1: Beyond the LMS”. Posted on *Hybrid Pedagogy* (5 March 2013), https://hybridpedagogy.org/decoding-digital-pedagogy-pt-1-beyond-the-lms/

[3] Croxall, Brian, and Adeline Koh. “Digital Pedagogy?”. Posted on *A Digital Pedagogy Unconference at #MLA13* (2013), https://www.briancroxall.net/digitalpedagogy/what-is-digital-pedagogy/

[4] “digital, n. and adj.”. *Oxford English Dictionary Online.* Oxford University Press (March 2020), https://www.oed.com/view/Entry/52611?redirectedFrom=digital

[5] Boehm, Carola. “The discipline that never was: Current developments in Music Technology in higher education in Britain” *Journal of Music, Technology and Education* 1, no. 3 (2007), 7-21, doi: 10.1386/jmte.1.1.7_1.

[6] Boehm, Carola, Russ Hepworth-Sawyer, Nick Hughes, and Dawid Ziemba. “The discipline that ‚became‘: Developments in Music Technology in British higher education between 2007 and 2018” *Journal of Music, Technology & Education* 11, no. 3 (2018), 251-67, doi: 10.1386/jmte.11.3.251_1.

[7] Inskip, Charles, Frans Wiering. “In their own words: using text analysis to identify musicologists’ attitudes towards technology” in *Proceedings of the 16th International Society for Music Information Retrieval Conference* (ISMIR 2015), 1-7. https://www.researchgate.net/publication/291096095_In_their_own_words_using_text_analysis_to_identify_musicologists‘_attitudes_towards_technology#fullTextFileContent

[8] Garwood, Deborah A., Alex H. Poole. “Pedagogy and public-funded research: an exploratory study of skills in digital humanities projects” *Journal of Documentation* 75, no. 3 (2019), 550-76, https://doi.org/10.1108/JD-06-2018-0094.

[9] Battershill, Claire, and Shawna Ross. *Using Digital Humanities in the Classroom.* London: Bloomsbury, 2017.

[10] Cordell, Ryan. “How not to Teach Digital Humanities” in *Debates in the Digital Humanities*, ed. Matthew K. Gold. Minneapolis: University of Minnesota, 2016, https://dhdebates.gc.cuny.edu/read/untitled/section/31326090-9c70-4c0a-b2b7-74361582977e#ch36

[11] Locke, Brandon T. "Digital Humanities Pedagogy as Essential Liberal Education: A Framework for Curriculum Development" *Digital Humanities Quarterly* 11, no. 3 (2017), http://www.digitalhumanities.org/dhq/vol/11/3/000303/000303.html

[12] Katherine D. Harris, "Play, Collaborate, Break, Build, Share: ‚Screwing Around' in Digital Pedagogy, The Debate to Define Digital Humanities...Again" *Polymath: An Interdisciplinary Arts and Sciences Journal* 3, no. 3 (Summer 2013), 1-26.

[13] Spiro, Lisa. "‚This is Why we Fight': Defining the Values of the Digital Humanities" in *Debates in the Digital Humanities*, ed. Matthew K. Gold. Minneapolis: University of Minnesota, 2016, https://dhdebates.gc.cuny.edu/read/40de72d8-f153-43fa-836b-a41d241e949c/section/9e014167-c688-43ab-8b12-0f6746095335#ch03

[14] Koh, Adeline. "Introducing Digital Humanities Work to Undergraduates: An Overview". Posted on *Hybrid Pedagogy* (13 August 2014), https://hybridpedagogy.org/introducing-digital-humanities-work-undergraduates-overview/

[15] Howell, Jonathan. "When Technology is too Hot, too Cold or Just Right" *Emerging Learning Design Journal* 5 (2017), 9-18, https://digitalcommons.montclair.edu/cgi/viewcontent.cgi?article=1020&context=eldj

[16] Sayers, Jentery. "Tinker-Centrick Pedagogy in Literature and Language Classrooms" in *Collaborative Approaches to the Digital in English Studies*, ed. Laura McGrath. Logan, UT: Computers and Composition Digital Press, 2011, https://ccdigitalpress.org/book/cad/Ch10_Sayers.pdf

[17] Alonso Garcia, Nuria, Alison Caplan, and Brad Mering. "A Pedagogy for Computer-Assisted Literary Analysis: Introducing GALGO (Golden Age Literature Glossary Online)" *Digital Humanities Quarterly* 11, no. 3 (2017), http://www.digitalhumanities.org/dhq/vol/11/3/000323/000323.html

[18] Howell, Jonathan. "When Technology is too Hot, too Cold or Just Right" *Emerging Learning Design Journal* 5 (2017), 9-18, https://digitalcommons.montclair.edu/cgi/viewcontent.cgi?article=1020&context=eldj.

[19] Fyfe, Paul. "Digital Pedagogy Unplugged" *Digital Humanities Quarterly* 5, no. 3 (2011), https://www.digitalhumanities.org/dhq/vol/5/3/000106/000106.html

[20] Shaffer, Kris P. "A Proposal for Open Peer Review" *Music Theory Online* 20, no. 1 (February 2014), https://mtosmt.org/issues/mto.14.20.1/mto.14.20.1.shaffer.php

IIIF-based lyric and neume editor for square-notation manuscripts

Juliette Regimbal
McGill University
juliette.regimbal@mail.mcgill.ca

Gabriel Vigliensoni
McGill University
gabriel.vigliensonimartin@mcgill.ca

Caitlin Hutnyk
McGill University
caitlin.hutnyk@mail.mcgill.ca

Ichiro Fujinaga
McGill University
ichiro.fujinaga@mcgill.ca

Abstract

In this paper we introduce a set of improvements to Neon, an online square-notation music editor based on the International Image Interoperability Framework (IIIF) and the Music Encoding Initiative (MEI) file format. The enhancements extend the functionality of Neon to the editing of lyrics and single-session editing of entire manuscripts and lyric editing. We describe a scheme for managing and processing the information necessary for visualizing and editing full manuscripts. A method of concurrently editing the position and content of lyrics is also discussed. We expect these will provide a better user experience when correcting the output of automated optical music recognition workflows.

Introduction

Neon is a web-based music editor for square notation designed for correcting the output of optical music recognition (OMR) workflows [1]. The project went through many iterations since its original release and currently uses MEI (Music Encoding Initiative) and Verovio[1] as its underlying format and technology [2]. In this paper, we present the latest advances in the application. The main improvements are: (i) the use of the International Image Interoperability Framework (IIIF)[2] to source the images and (ii) the ability to display and edit text in the page. Also, we propose a method of relating IIIF Manifests to MEI files.

These refinements to Neon are motivated by actual musical needs. Using IIIF allows users to view the entire manuscript, as opposed to the previous page-by-page editing approach of Neon, which lacked surrounding context. Being able to edit text is crucial for chant in square notation because, like all neume notations, the music is composed to be sung. As a result there is a direct mapping between the neumes and syllables. Since the MEI Neume module is capable of expressing the link between these elements in a hierarchical fashion [3], we can visualize and edit text effectively using MEI.

Section 1

Section 1.1

Musical works are best described in their original sources. For square-notation music these are manuscripts. High-quality images are necessary for processes like OMR, which rely on computers to create digital encodings, but are also necessary for human editors that cannot physically access the original source. Using images presents its own problems; an image representing a page can be well over 100 MB and ordering these images requires additional information. These characteristics result in a large payload to transfer for even one image where much of it is unused as humans need less detail to recognize musical elements than computers.

1 https://www.verovio.org/
2 https://iiif.io/

IIIF addresses this problem through its Image API (Application Programming Interface). It allows parts of an image to be requested at various sizes [4]. These sizes permit a IIIF viewer to request images of varying levels of detail based on how much a user zooms in. This can reduce download times for images while maintaining a consistent quality of user experience. The IIIF Presentation API provides information about the overall document including page order.

We integrated Diva.js into Neon to display entire manuscripts with square notation. Diva.js[3] is a IIIF-compliant document viewer written in JavaScript [5]. It is particularly suited for the purposes of viewing and scrolling through large manuscripts as it loads parts of images as needed. A metadata file discussed in the next section is used to associate MEI documents to their corresponding pages. After each document is associated with its corresponding image, it can be rendered and overlaid on the source page displayed by Diva.js.

Section 1.2

One significant challenge in manuscript viewing and correction is the mapping of the MEI encoding the musical content of a page to the image source for that page. This mapping must be determined quickly to reduce loading time and be usable with multiple sets of MEI files. Three approaches were considered to create these mappings: in the MEI files, in the IIIF Presentation Manifest, or in an additional metadata document. Ultimately the metadata document method was selected.

The source description field of an MEI document can include information about the source image used to produce the encoding. Determining which MEI document corresponds to which page is trivial. However this approach requires all MEI documents to be loaded and processed before any data can be conveyed, adding latency to the correction or viewing process.

The IIIF Presentation Manifest provides a means of adding annotations to documents that could be used to associate an MEI document to its corresponding image [4]. Since this information is provided in a document that must be downloaded for a viewer anyway, the additional loading time is minimal. However, the IIIF Manifest can only be changed by the organization hosting it and restricts the ability of people to add new sets of MEI files to a source or change existing files in any way that would require changes to the manifest.

Using a separate metadata file for these associations proved to be the most suitable approach. This method forms a "Neon Manifest"[4] containing the IIIF Manifest, defining the source and its pages, and annotations between MEI documents and their corresponding pages. These annotations are stored as a JSON-LD[5] file. Different metadata files can exist to represent the results of different OMR processes or different editors.

The implementation of this separate manifest provides an efficient way to use a IIIF viewer such as Diva.js in an editor for square-notation manuscripts.

Section 2

As almost all square-notation music contains lyrics, a lack of support for viewing or editing that information makes a square-notation editor incomplete. Since Neon operates as part of OMR, an ability to interact with both the text itself as well as its location on a page is essential. With lyric alignment approaches now being available for OMR [6] it is possible to include text and position information about lyrics in MEI.

There are two main considerations for syllable text editing: how to encode it in MEI and how to display the information to the user. In the MEI 4.0 Neume Schema, text is segmented syllable-by-syllable. The text for each syllable is included in the <syl> element, which is part of a <syllable> element that also contains the neumes. A <zone> element associated with the <syl> element describes where the text would appear in the page. This facsimile information is already used to encode the layout of other musical elements in Neon. Neume editing can result in the neumes being grouped into one <syllable> element from many or split into many <syllable> elements from one. In these cases the text and location information of <syl> elements must be modified as well to reflect these changes in the encoding and permit manual editing.

3 https://ddmal.music.mcgill.ca/diva.js/

4 https://github.com/DDMAL/Neon/wiki/Neon-Manifest

5 https://json-ld.org/

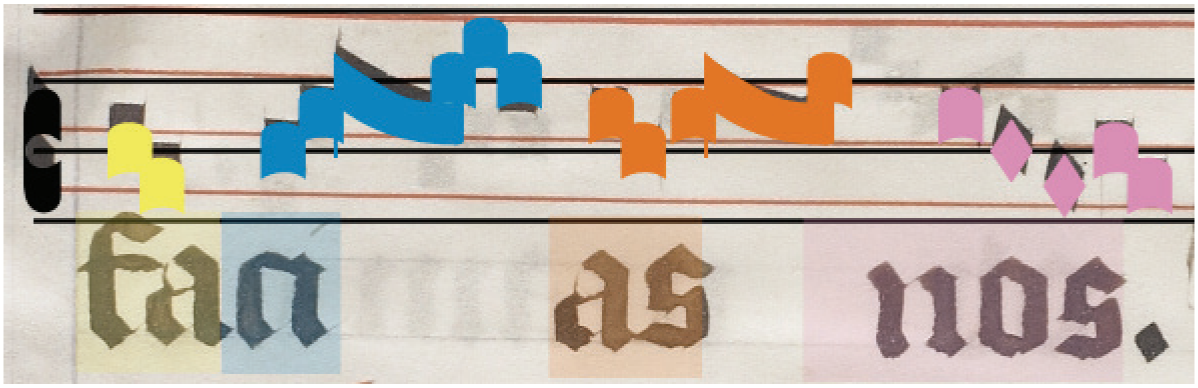


Figure 1: Figure 1: An example of lyric bounding boxes being visualized in Neon. Each syllable, including both neumes and text, is highlighted in a different color. If correction of the linking between syllables and neumes is needed, Neon provides a functionality to perform this mapping.

Since <syl> elements must exist in a syllable and using multiple <syl> elements per syllable is redundant, it is guaranteed in Neon that each <syllable> element will have exactly one <syl> child.

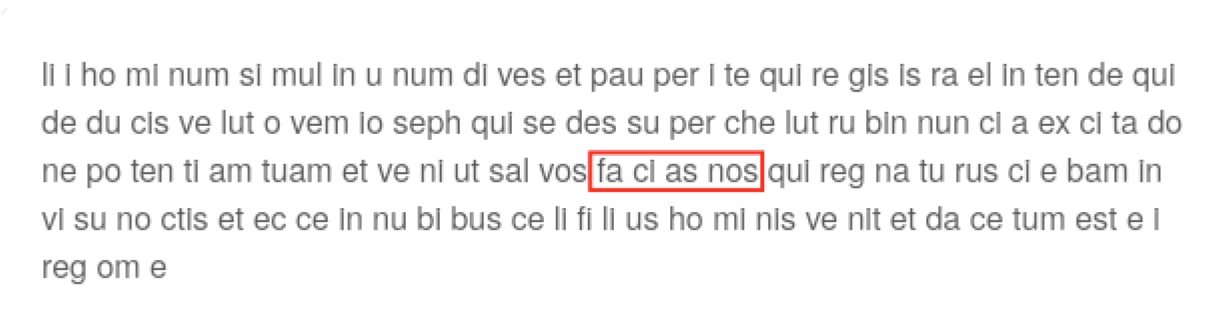


Figure 2: Figure 2: The text content of a page displayed in Neon. The syllables shown in Figure 1 are contained in a red box.

To facilitate editing of neumes, the source image is overlaid with the Verovio rendering of musical symbols. With lyrics, this method does not provide similar benefits, because the appearance of letters vary from source to source and spacing between letters is inconsistent even within a page. Neon displays the text in a separate window beside the image. Partially-transparent bounding boxes are overlaid as shown in Figure 1 while the text of each syllable is displayed as in Figure 2. The bounding box corresponds to the information in the <zone> element associated with the <syl> element.

Conclusion

Square-notation music was written in multi-page manuscripts and represent pitches that are sung with lyrics. Support for multiple pages is provided using a IIIF Manifest file to supply information about the source images. With a separate manifest file relating specific MEI documents to manuscript pages, Diva.js is used to render the MEI over source images across multiple pages. Lyric visualizing and editing features are added. These convey the position and context of text in MEI and permit the user to edit lyrics while concurrently correcting neumes. The implications of editing neumes in MEI on lyrics are considered and resolved in the new version of Neon. Together, these features provide a more complete user experience.

Works cited

[1] Burlet, Gregory, Alastair Porter, Andrew Hankinson, and Ichiro Fujinaga. “Neon. js: Neume Editor Online” in *Proceedings of the International Society for Music Information Retrieval Conference* (ISMIR 2012), 121–26.

[2] Regimbal, Juliette, Zoé McLennan, Gabriel Vigliensoni, Andrew Tran, and Ichiro Fujinaga. “Neon2: A Verovio-based square-notation editor” presented at the Music Encoding Conference, Vienna, Austria, May 29-June 1, 2019.

[3] De Luca, Elsa, et al. “Cantus Ultimus’ MEI Neume Module and its Interoperability Across Chant Notations” presented at the Music Encoding Conference, Vienna, Austria, May 29-June 1, 2019.

[4] Snydman, Stuart, Robert Sanderson, and Tom Cramer. “The International Image Interoperability Framework (IIIF): A community & technology approach for web-based images” in *IS&T Archiving Conference 2015* (ARCHIVING 2015), 16–21.

[5] Hankinson, Andrew, Wendy Liu, Laurent Pugin, and Ichiro Fujinaga. “Diva. js: A Continuous Document Viewing Interface” Code4Lib Journal, no. 14 (2011).

[6] de Reuse, Timothy, and Ichiro Fujinaga. “Robust transcript alignment on medieval chant manuscripts” in *Proceedings of the 2nd International Workshop on Reading Music Systems* (WoRMS 2019), 21-26.

Retrieving Music Semantics from Optical Music Recognition by Machine Translation

Martha E. Thomae
McGill University
martha.thomaeelias@mail.mcgill.ca

Antonio Ríos-Vila
University of Alicante
arios@dlsi.ua.es

Jorge Calvo-Zaragoza
University of Alicante
jcalvo@dlsi.ua.es

David Rizo
University of Alicante
drizo@dlsi.ua.es

José M. Iñesta
University of Alicante
inesta@dlsi.ua.es

Abstract

In this paper, we apply machine translation techniques to solve one of the central problems in the field of optical music recognition: extracting the semantics of a sequence of music characters. So far, this problem has been approached through heuristics and grammars, which are not generalizable solutions. We borrowed the seq2seq model and the attention mechanism from machine translation to address this issue. Given its example-based learning, the model proposed is meant to apply to different notations provided there is enough training data. The model was tested on the PrIMuS dataset of common Western music notation incipits. Its performance was satisfactory for the vast majority of examples, flawlessly extracting the musical meaning of 85% of the incipits in the test set—mapping correctly series of accidentals into key signatures, pairs of digits into time signatures, combinations of digits and rests into multi-measure rests, detecting implicit accidentals, etc.

Introduction

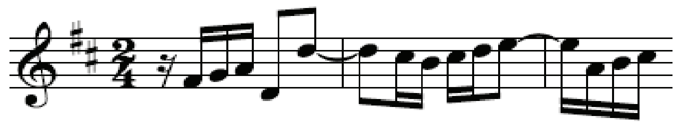
We present a machine learning-based approach to retrieve the semantics of a sequence of (graphic) music symbols, which constitutes a central problem in the field of Optical Music Recognition (OMR). OMR is the process of converting the digital image of a score into a symbolic file encoding the music content of that score. The traditional OMR workflow consists of four stages: preprocessing, symbol recognition, music reconstruction, and music encoding. The third stage, music reconstruction, must retrieve the actual musical meaning of the graphical symbols recognized in the previous stage. So far, the models proposed to solve this problem are based on rules [1] or grammars [2, 3], which prevents their use in other contexts (e.g., in notation systems other than the one for which they were implemented). For high scalability, we propose a machine learning-based approach which learns the semantics of a particular notation system by providing the model with enough training examples.

We use an encoding introduced by [4] to represent the graphical and semantic information obtained by the second and third stages of the OMR workflow, respectively. This encoding provides an intermediate representation that, during the music encoding stage of the OMR process, becomes a well-established music format, such as MusicXML, MEI, or **kern.

Background

Agnostic and Semantic Encodings of Sequences

In MEC'2017, [5] presented the concept of agnostic and semantic sequential representations of a music score. The agnostic encoding represents the output of the music symbol recognition stage of the OMR, where we only have the graphical information about the symbols (their shapes and positions) and no musical meaning. The agnostic representation is a sequential encoding of the graphical symbols in the score (Figure 1b). Each token in the sequence encodes two types of information: the label of the symbol (e.g., C clef, quarter note, half note, sharp) and its vertical position within the staff (e.g., third line, fourth space). On the other hand, the semantic representation is a sequential encoding of symbols in a score, which includes their musical meaning (Figure 1c). Translating an agnostic sequence into a semantic one involves several tasks, including re-interpreting a series of accidentals into a key signature and parsing the position of the notes in the staff into pitch values.



(a) Music excerpt.

```
clef.G-L2, accidental.sharp-L5, accidental.sharp-S3, digit.2-L4, digit.4-L2, rest.sixteenth-L3,
note.beamedRight2-S1, note.beamedBoth2-L2, note.beamedLeft2-S2, note.beamedRight1-S0,
note.beamedLeft1-L4, slur.start-L4, barline-L1, slur.end-L4, note.beamedRight1-L4,
note.beamedBoth2-S3, note.beamedLeft2-L3, note.beamedRight2-S3, note.beamedBoth2-L4,
note.beamedLeft1-S4, slur.start-S4, barline-L1, slur.end-S4, note.beamedRight2-S4,
note.beamedBoth2-S2, note.beamedBoth2-L3, note.beamedLeft2-S3
```

(b) Agnostic encoding of the music excerpt.

```
clef-G2, keySignature-DM, timeSignature-2/4, rest-sixteenth, note-F#4_sixteenth,
note-G4_sixteenth, note-A4_sixteenth, note-D4_eighth, note-D5_eighth, tie, barline,
note-D5_eighth, note-C#5_sixteenth, note-B4_sixteenth, note-C#5_sixteenth,
note-D5_sixteenth, note-E5_eighth, tie, barline, note-E5_sixteenth,
note-A4_sixteenth, note-B4_sixteenth, note-C#5_sixteenth
```

(c) Semantic encoding of the music excerpt

Figure 1: Example of the agnostic and semantic encoding of a musical excerpt [4].

In [4], it was shown that sequential encodings are suitable for converting a digital image into either an agnostic or a semantic representation without human-encoded rules, with more robust results in the agnostic [6]. In this paper, we implement a machine translator that takes the agnostic representation of a sequence of notes in the staff and generates its corresponding semantic representation, in order to take advantage of the performance of the agnostic case for OMR.

Translation Model Description

The main task of the model is to translate an agnostic sequence into its corresponding semantic sequence. We used a “seq2seq” model, first introduced by [7] for machine translation. A seq2seq model consists of two parts, an encoder that maps the input sequence (in this case, the agnostic sequence) onto a fixed-dimension vector, and a decoder that builds the target sequence (here, the semantic sequence) from that vector. We added an attention mechanism, which has been used to improve the translation results by selectively focusing on parts of the input sentence during translation [8]. The attention mechanism allows us to visualize which tokens (graphical symbols) of the agnostic sequence affect the translated tokens of the semantic sequence. In other words, it can show us what the model is paying attention to when translating (Figure 2).

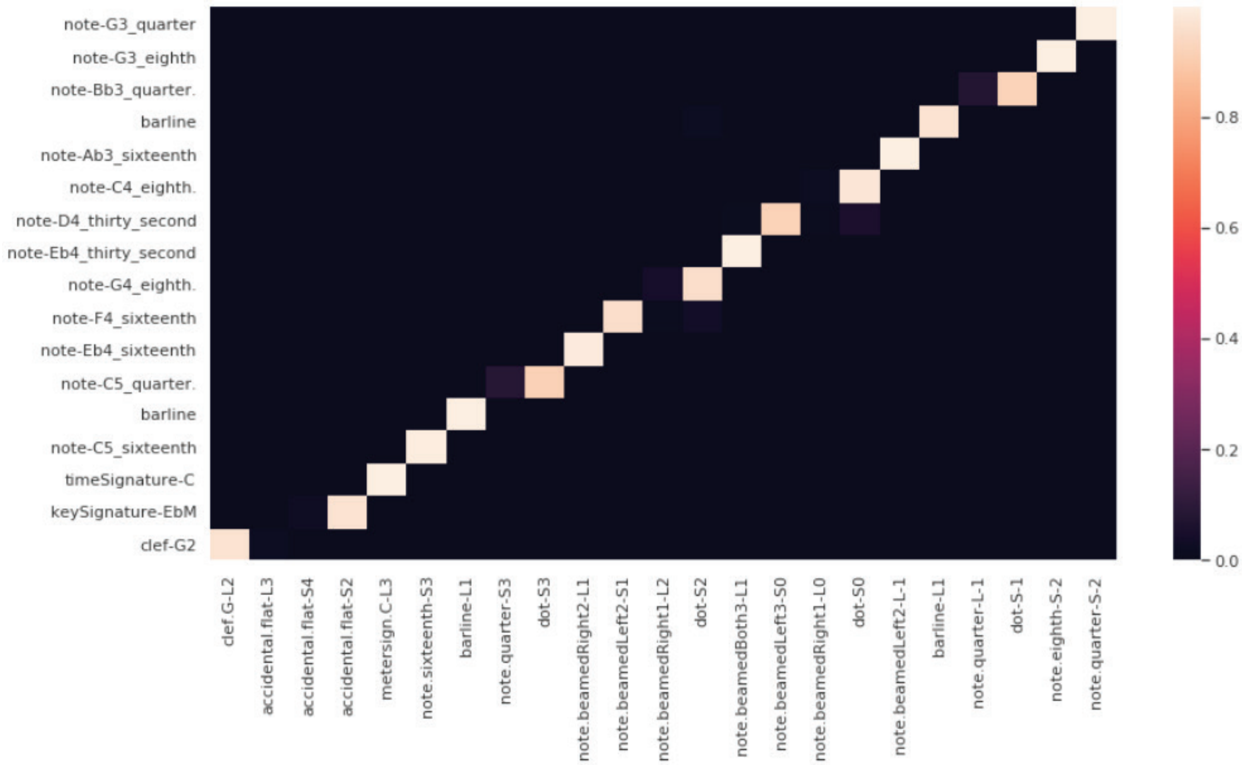
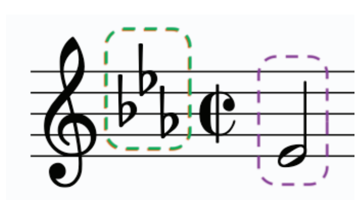


Figure 2: Attention matrix of the model when translating an agnostic sequence (horizontal axis) into a semantic sequence (vertical axis).

Experiment and discussion

We tested this model's performance on the Printed Images of Music Staves (PrIMuS) dataset [4]. The PrIMus dataset consists of 87,678 music incipits from RISM encoded in a variety of formats, including the agnostic and semantic representations mentioned above. We used an 80:10:10 split for training, validation, and testing.

We evaluated the model using the edit distance, which measures the number of operations (in terms of insertion, deletion, and substitution of tokens) needed for two strings to match. Given an agnostic sentence, the edit distance was computed between the corresponding semantic sequence in the dataset and the translated sequence obtained. The model flawlessly extracted the musical meaning of 85% of the agnostic sentences in the test set, correctly identifying key signatures, time signatures, multi-measure rests, dotted notes, and notes affected by notated or implied accidentals (see Figures 3 and 4).

(a) Beginning of one of the incipits in the test set.

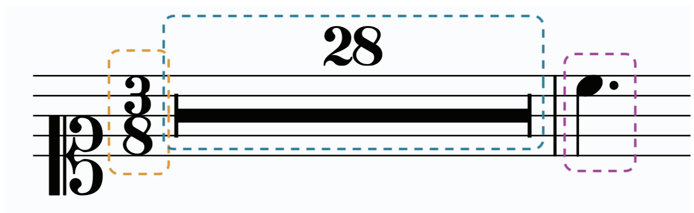
clef.G-L2 accidental.flat-L3 accidental.flat-S4 accidental.flat-S2 metersign.C/-L3 note.half-L1

(b) Agnostic encoding of the music excerpt.

clef-G2 keySignature-EbM timeSignature-C/ note-Eb4_half

(c) Semantic encoding generated by the model.

Figure 3: Example of the translation of key signatures (green) and implicit accidentals (purple) by the model.



(a) Beginning of one of the incipits in the test set.

clef.C-L1 digit.3-L4 digit.8-L2 digit.2-S5 digit.8-S5 multirest-L3
barline-L1 note.quarter-S4 dot-S4

(b) Agnostic encoding of the music excerpt.

clef-C1 timeSignature-3/8 multirest-28 barline note-C5_quarter.

(c) Semantic encoding generated by the model.

Figure 4: Example of the translation of time signatures (yellow), multi-measure rests (blue), and dotted notes (purple) by the model.

According to the edit distance values obtained, for 7% of the test sentences, only one error was made in the translation. One example of this is the sequence shown in Figure 2, where the last dotted note (coming from the agnostic tokens “note.quarter-L-1 dot-S-1”) is wrongly translated into a Bb instead of Ab. As seen in the attention matrix of Figure 2, when translating dotted notes, the translator pays more attention to the dot token than to the preceding note token. Similar to dotted notes, the model also pays considerably more attention to the last accident in a series of accidentals at the moment of parsing the key signature.

As can be seen from Figure 6, most error-free sentences lie on the average-sentence-length region (the 18–33 interval with the highest data concentration in Figure 5). Analyzing some of the examples with the highest edit distance values, some of the patterns found are the presence of a clef change, after which the translator’s performance consistently drops for all following tokens; and long sentences (of more than 35 tokens, lying in the right end of the distribution shown in Figure 5).

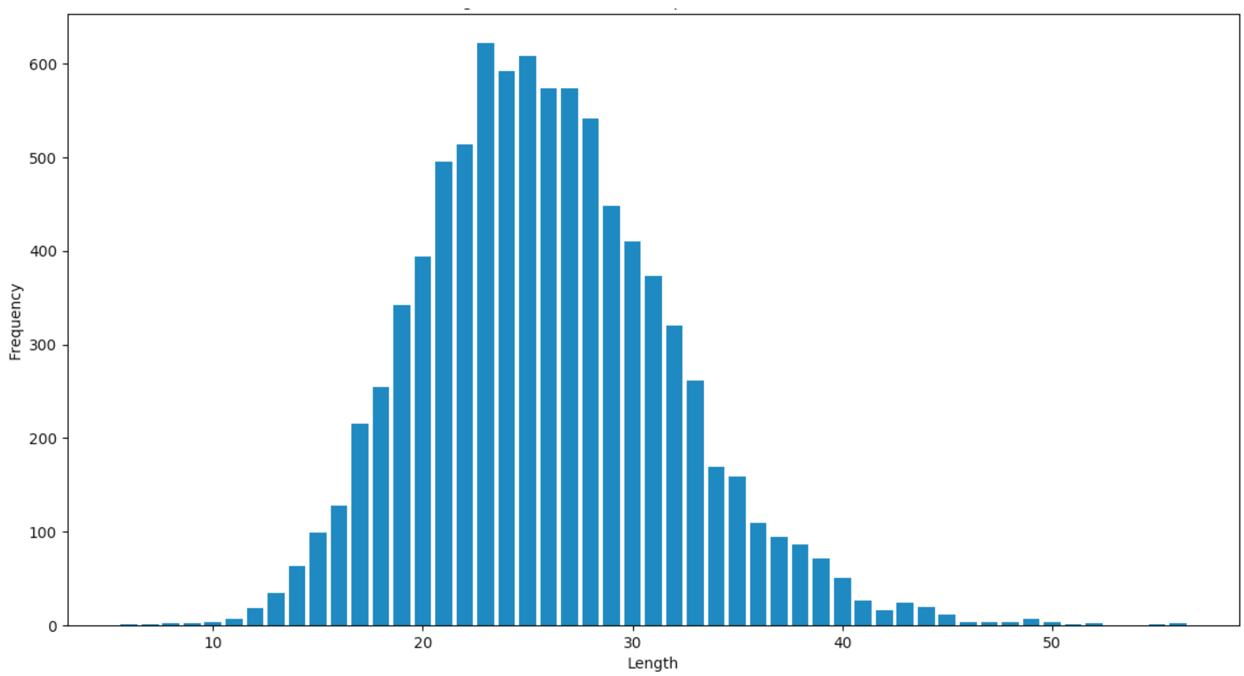


Figure 5: Length of the semantic sequences in the test set.

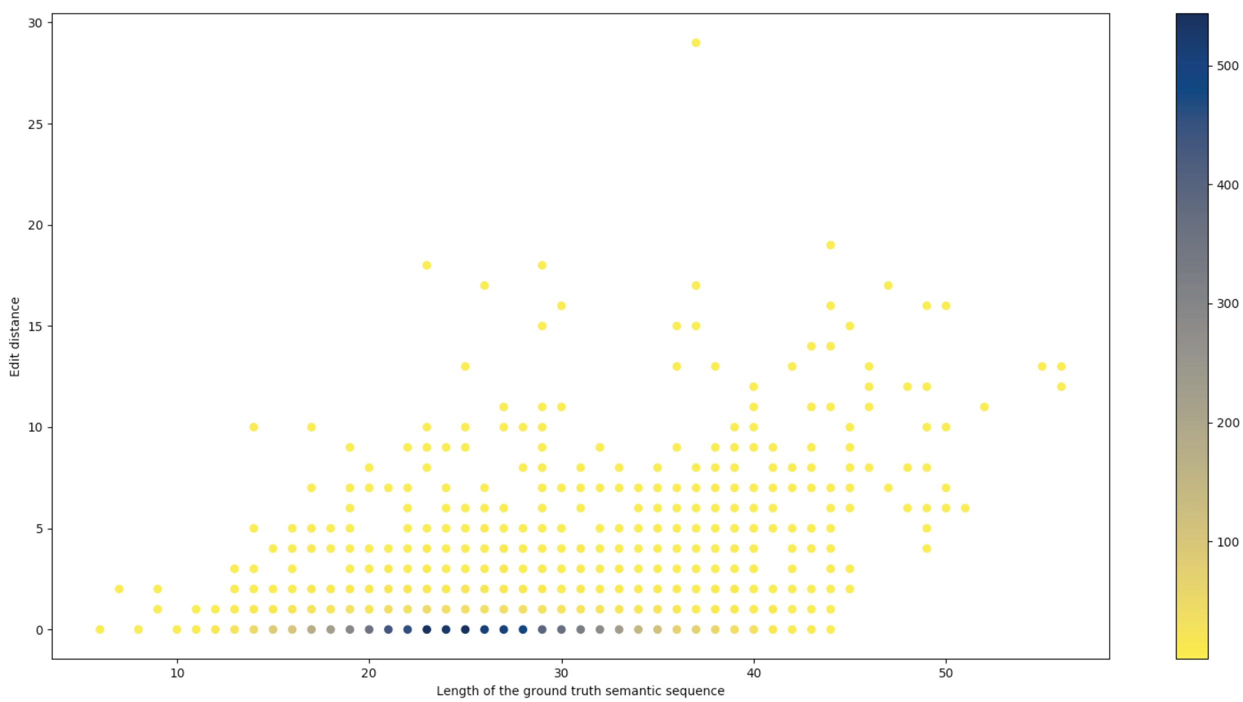


Figure 6: Color density plot of the edit distance of all sentences in the test set. The color bar indicates the frequency of a particular (sentence length, edit distance) pair.

Conclusion

Given its example-based learning, the model we propose is meant to apply to different notation systems provided there is enough training data. The performance in the PrIMuS dataset was satisfactory for the vast majority of examples. However, we plan to improve the attention mechanism to enhance its performance before tackling notation systems with more complex semantics (e.g., mensural notation). Other future work includes the substitution of the semantic representation by **kern, a well-established music encoding format that also encodes the music symbols sequentially for each staff. The advantages of **kern over the semantic encoding are that the former allows for rendering the encoded sequence in Verovio, and that there is technology already available to obtain more complex formats (e.g., MEI or MusicXML) from **kern [9].

Acknowledgements

This work is supported by the Spanish Ministry HISPAMUS project TIN2017-86576-R, partially funded by the EU, and by CIRMMT's Inter-Centre Research Exchange Funding and McGill's Graduate Mobility Award.

Works cited

[1] Rossant, Florence, and Isabelle Bloch. "Robust and Adaptive OMR System Including Fuzzy Modeling, Fusion of Musical Rules, and Possible Error Detection" EURASIP Journal on Advances in Signal Processing, no. 1 (2007), https://doi.org/10.1155/2007/81541.

[2] Couasnon, Bertrand. "DMOS: A Generic Document Recognition Method, Application to an Automatic Generator of Musical Scores, Mathematical Formulae and Table Structures Recognition Systems" in *Proceedings of the 6th International Conference on Document Analysis and Recognition* (ICDAR 2001), 215–20, https://doi.org/10.1109/ICDAR.2001.953786.

[3] Szwoch, Mariusz. "Guido: A Musical Score Recognition System" in *Proceedings of the 9th International Conference on Document Analysis and Recognition* (ICDAR 2007), vol. 2, 809–13, https://doi.org/10.1109/ICDAR.2007.4377027.

[4] Calvo-Zaragoza, Jorge, and David Rizo. "End-to-End Neural Optical Music Recognition of Monophonic Scores" Applied Sciences 8, no. 4 (2018), 606–29, https://doi.org/10.3390/app8040606.

[5] Rizo, David, Jorge Calvo-Zaragoza, José M. Iñesta, and Ichiro Fujinaga. "About Agnostic Representation of Musical Documents for Optical Music Recognition" presented at the Music Encoding Conference, Tours, France, May 16-19, 2017.

[6] Calvo-Zaragoza, Jorge, and David Rizo. "Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores" in *Proceedings of the 19th International Society for Music Information Retrieval Conference* (ISMIR 2018), 248–55.

[7] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks" in *Proceedings of the 27th International Conference on Neural Information Processing Systems* (NIPS'14), 3104–12, https://www.arxiv-vanity.com/papers/1409.3215/

[8] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective Approaches to Attention-Based Neural Machine Translation" ArXiv:1508.04025 (2015), http://arxiv.org/abs/1508.04025

[9] Sapp, Craig Stuart. "Verovio Humdrum Viewer" presented at the Music Encoding Conference, Tours, France, May 16-19, 2017.

Editing Italian Madrigals in the Digital World: The Tasso in Music Project

Emiliano Ricciardi
University of Massachusetts Amherst
ericciardi@music.umass.edu

Craig Stuart Sapp
Center for Computer Assisted Research in the Humanities, Stanford University/Packard Humanities Institute
craig@ccrma.stanford.edu

Abstract

Despite the interdisciplinary nature of the Italian madrigal—a genre in which poetry and music often stand on equal footing—critical editions of this repertoire tend to focus primarily on the *musical* text, devoting limited attention to the often-complex philological tradition of the poems set to music. Likewise, most critical editions are devoted to the works of a single composer—as opposed to settings of the same poetry by multiple composers—and thus offer a rather segmented perspective on the repertoire, which is not conducive to the study of musical traditions and to comparative analysis. This paper proposes a new model for critical editions of this repertoire, one in which musical *and* poetic texts are devoted equal attention. To do so, we will provide an overview of a digital project that follows this model, namely the Tasso in Music Project (www.tassomusic.org), showing how it draws on both musical (Humdrum, MEI) and textual (TEI) encoding systems to render the interdisciplinary nature of its repertoire.

Introduction

One of the distinctive features of the Italian madrigal is that it is as much about poetry as it is about music. Composers would often respond to the literary taste of their milieus by engaging in sophisticated renditions of poetry by notable authors, ranging from Petrarch to Marino. This poetry frequently had a distinguished textual tradition of its own, being widely disseminated through literary manuscripts and prints before it was set to music. This strong literary component is underrepresented in critical editions of the madrigal repertoire. These are typically devoted to the works of single composers—as opposed to settings of the same poetry by multiple composers—and feature critical apparatuses that rarely engage with the complex textual traditions of the poems set to music. This approach to making critical editions of Italian madrigals is a function also of the limitations of the printed medium, which typically does not allow for complex critical apparatuses encompassing both the musical and the literary tradition. Furthermore, the printed medium has historically been resistant to collaboration between musicologists and literary scholars, which would instead be desirable if one were to engage with the interdisciplinary nature of Italian madrigals.

Drawing on the possibilities afforded by digital encoding, the Tasso in Music Project[1]—the first complete edition of the early modern musical settings of poetry by Torquato Tasso (1544–1595)—provides a different approach to editing Italian madrigals and related genres. Indeed, the project is poet-centered and devotes equal attention to the musical and poetic realms, thanks also to collaboration between music and literary scholars. The goal of this presentation is to provide an overview of the musical and poetic repertoire under consideration and, most importantly, to illustrate the digital features that allow users of the Tasso in Music Project fully to appreciate the interdisciplinary nature of its repertoire. These features include musical encodings and renderings in Humdrum and MEI/Verovio, TEI literary encodings, and tools for the analysis of music-text relations. In doing so, this presentation seeks to provide an alternative model for critical editions of Italian madrigals, one that could be adapted also to other vocal repertoires.

1 www.tassomusic.org

Repertoire

Torquato Tasso (1544-1595) was arguably the most prominent poet of late sixteenth-century Italy. His works, most notably the epic poem *Gerusalemme liberata* (Jerusalem Delivered), achieved tremendous fame in literary circles, shaping the poetic culture of the time. Tasso's influence extended also well beyond the literary realm. Indeed, his poems became a source of inspiration for visual artists and, most importantly, for composers, among whom they became true hits. From the 1570s through the 1630s, virtually all composers of secular vocal music in Italy and Europe, including notable ones like Luca Marenzio and Claudio Monteverdi, set one or more of his poems, producing a total of over 750 settings [1, 2]. Composers were especially drawn to Tasso's lyric poems, collectively known as *Rime*, whose conciseness and wit reflected the taste for *concettismo* typical of the time and offered opportunities for equally clever musical renditions [3], but also proved fond of Tasso's *Gerusalemme* and the pastoral drama *Aminta*, whose dramatic features resonated with a growing tendency toward quasi-operatic styles in secular vocal music around 1600. As is typical with repertoire from this period, however, the majority of these settings, over three quarters, have been unavailable in modern editions. As a result, this significant corpus of works has remained largely unexplored in both scholarship and performance, which has in turn hindered a serious assessment of Tasso's influence on early modern musical culture.

Funded by a three-year NEH Scholarly Editions and Translations Grant (2016–19) and scheduled for completion in 2020, the Tasso in Music Project fills this lacuna through a complete digital edition of the extant settings of Tasso's poetry. Carried out by a team of musicologists, literary scholars, and digital humanities experts from North America and Europe,[2] the project provides open online access to one of the largest digital editions of early modern music, complemented by a rich literary component and tools for analysis. Representing the work of over 200 composers, the project provides a snapshot of secular vocal music in an age in which it underwent profound transformations. Accordingly, it lends itself especially well to comparative analysis and to the study of emulation among composers. Likewise, this corpus provides fertile ground for the study of music-text relations.

Web infrastructure

The website is created using a static site generator, Jekyll,[3] and is hosted on Github Pages.[4] Unlike other site generators, Jekyll is integrated into Github Pages. Therefore, the compiling of the website from source files occurs transparently and behind the scenes, allowing non-technical developers to edit content without the need to know how to regenerate the website from the source files.

Dynamic content is rendered with Handlebars.[5] This is a templating system that generates content on the fly within a user's web browser, as opposed to Jekyll, which can only prepare static textual content. Metadata is stored as text files in the ATON format,[6] which is then converted into JSON data in the user's browser. This conversion provides lookup tables of the metadata used in template filling within Handlebars. An advantage of ATON over JSON is that there is less formatting structure, which allows for easier editing by non-technical users. In addition, it enables comments, so that internal documentation of the data can be contained within each metadata file. Use of Jekyll/Handlebars/ATON mimics the older, more centralized web architecture of PHP.[7] However, it allows for a serverless implementation of the website, which enables better security, long-term stability, and rapid adaptation to faster generation of web pages, as dynamic content is created within a user's web browser, rather than on the remote web server.

The raw files are stored in a Github repository,[8] and metadata is collected into a single directory/folder for ease of curation and long-term maintenance as a set of text files.[9] Metadata for musical settings of each liter-

---

2 https://www.tassomusic.org/about/participants

3 https://www.jekyllrb.com

4 https://pages.github.com

5 https://handlebarsjs.com

6 http://aton.sapp.org

7 https://www.php.net

8 https://github.com/TassoInMusicProject/tasso-website

9 https://github.com/TassoInMusicProject/tasso-website/tree/gh-pages/data/indexes

ary genre is stored in separate files,[10] and separate lookup tables are provided to minimize metadata redundancy in the data files, such as the composer index file[11] and the RISM sources list.[12] Initial entry of metadata is usually done using Google Spreadsheets, and smaller ATON metadata files such as the composer and RISM indexes are generated automatically from the spreadsheet files.[13] Spreadsheet files could be directly loaded into web browsers, bypassing storage in ATON files, by using the TSV export functionality from Google Spreadsheets.[14] This would simplify metadata management for non-technical developers, but loading data this way is slightly slower and is less stable for long-term management of the website, since metadata content would be dependent on an external source from the website.

Musical editions

The Tasso in Music Project provides newly made critical editions of the extant early modern settings of Tasso's poetry, for a total of over 750 scores. A detailed description of the editorial policies (source choice, editorial accidentals, formatting of the text underlay, etc.) is available on the website.[15] The musical scores are initially entered in a graphical music editor (either Finale or Sibelius, depending on the preference of the editors). Data is then saved in MusicXML and converted in Humdrum, which is the project's main music encoding system. There are bugs and complications when exporting from both Finale and Sibelius. For instance, Finale does not correctly export elision characters, which need to be replaced with underscores. Sibelius does not correctly encode visual accidentals on recurrent pitches within a measure, which requires post-processing editing.

MusicXml files are converted either individually using Verovio Humdrum Viewer,[16] or converted in batches on the command line with *musicxml2hum*[17] and then with *tassoize*[18] to refine the music representation. Certain musical features such as figured base are edited manually in the conversions. Figured bass is entered as lyrics in the graphical notation editors, and then transformed from lyrics to figured bass data within the final Hudrum encoding. Metadata from the ATON files on the website as well as original clefs and mensuration information are added in the final processing.[19] Although the critical editions are encoded in Humdrum, the project's primary and archival encoding system, they are available for consultation and download in a variety of electronic encodings and renderings, such as MEI, MusicXML, Musedata, MIDI, PDF and MP3 for use in various software systems and applications.

The final scores are rendered into graphical music notation using Verovio[20] and displayed on work pages.[21] Music notation is generated dynamically within the webpage, allowing for interactive notational features such as part extraction, highlighted notes for dynamic playback and searches, switching to original clefs/mensurations, removing lyrics, and collapsing the full score to an incipit. The graphical scores are rendered online with the JavaScript version of Verovio and displayed directly within the webpage as an SVG image, with additional conversion to PDF for downloading. Data links to Verovio Humdrum Viewer[22] are also provided for online editing of the scores, such as preparation for performance, with exports of Humdrum data as MEI or PDF files [4].[23]

10 https://github.com/TassoInMusicProject/tasso-website/blob/gh-pages/data/indexes/rime-settings.aton, https://github.com/TassoInMusicProject/tasso-website/blob/gh-pages/data/indexes/gerusalemme-settings.aton, https://github.com/TassoInMusicProject/tasso-website/blob/gh-pages/data/indexes/aminta-settings.aton, and https://github.com/TassoInMusicProject/tasso-website/blob/gh-pages/data/indexes/other-settings.aton.

11 https://github.com/TassoInMusicProject/tasso-website/blob/gh-pages/data/indexes/composers.aton

12 https://github.com/TassoInMusicProject/tasso-website/blob/gh-pages/data/indexes/sources.aton

13 https://docs.google.com/spreadsheets/d/1YcaAQc-mxFFDWyOsT7HOlwZkEqB4rPEuOzXCMTKprMc/edit#gid=2068334514

14 docs.google.com/spreadsheets/d/1YcaAQc-mxFFDWyOsT7HOlwZkEqB4rPEuOzXCMTKprMc/export?gid=2068334514&format=tsv

15 http://www.tassomusic.org/about/policies

16 http://doc.verovio.humdrum.org

17 https://github.com/craigsapp/humlib/blob/master/src/tool-musicxml2hum.cpp

18 https://github.com/craigsapp/humlib/blob/master/src/tool-tassoize.cpp

19 https://github.com/TassoInMusicProject/tasso-scores/blob/master/bin/fillinrefinfo for inserting metadata, and https://github.com/TassoInMusicProject/tasso-scores/blob/master/bin/addoriginal for inserting original clefs and mensuration signs.

20 https://www.verovio.org

21 https://www.tassomusic.org/work/?id=Trm0047m

22 https://verovio.humdrum.org

23 A sample edition can be viewed at http://www.tassomusic.org/work/?id=Trm0862a

Musical scores are stored in an independent repository from the website,[24] which allows for a more stable long-term maintenance of the scores (for example, the website could be reimplemented in the future). Individual scores are stored by literary genre, with a separate digital score in each file.[25] Filenames start with the Tasso in Music Project catalog number for automatic processing of the scores. The catalog number is then followed by title, composer, and publication data for readability and ease of data management. For example, Ruggiero Giovannelli's 1588 setting of "Non è questa la mano" (*Rime* 47) is catalogued as follows:

Trm0047m-Non_e_questa_la_mano--Giovannelli_1588.krn. Scores are archived in the Humdrum file format, with metadata pulled automatically from the ATON files into the scores and stored in reference records:[26]

```
!!!COM: Giovannelli, Ruggiero
!!!CDT: ~1560-1625/01/07
!!!OTL: Non è questa la mano
!!!PTL: Fiori musicali, libro secondo
!!!PPP: Venice
!!!PPR: Vincenzi & Amadino
!!!PDT: 1588
!!!PUB-format: anthology
!!!AGN: Madrigal
!!!SCT: Trm0047m
!!!SCA: Trm0047m
!!!rime: 47
!!!original-voices: 3
!!!extant-voices: 3
!!!complete: Y
!!!final: G
```

Literary variants

In addition to musical editions, the Tasso in Music Project features quasi-diplomatic TEI transcriptions of the poetic texts as they appear in the musical sources and in contemporaneous literary sources, both manuscript and printed. Variants in the poetic texts are marked-up and rendered online, facilitating a real-time assessment of the textual tradition. This apparatus caters to the research interests not only of music historians, but also of literary scholars and linguists. For instance, music historians may use it to assess possible relationships (or lack thereof) between settings of the same poem. A case in point is the *madrigale libero* "Tarquinia, se rimiri" (*Rime* 560), which was set by seventeen composers.[27] Sixteen composers set a version of the poem in which the opening line reads "Mentre, mia stella, miri," found also in two literary manuscripts (**A3**, **E7**) and in all literary prints, attesting to the extensive lineage of this *lectio*. In a setting published in 1571, however, Ferrarese composer Luzzasco Luzzaschi set a substantially different version of the poem, whose opening line diverges from both the musical and the literary tradition, as it reads "Mentre l'ardenti stelle." This case is particularly interesting because of Luzzaschi's proximity to Tasso, who had joined the Ferrarese court in 1565. As Newcomb and Piperno have suggested, Luzzaschi may have had access to an early version of the poem that did not find its way into the later literary tradition [5, 6].

24 https://github.com/TassoInMusicProject/tasso-scores

25 https://github.com/TassoInMusicProject/tasso-scores/blob/master/Trm/kern/Trm0047m-Non_e_questa_la_mano--Giovannelli_1588.krn.

26 https://www.humdrum.org/reference-records

27 http://www.tassomusic.org/variants/?id=Trm0560

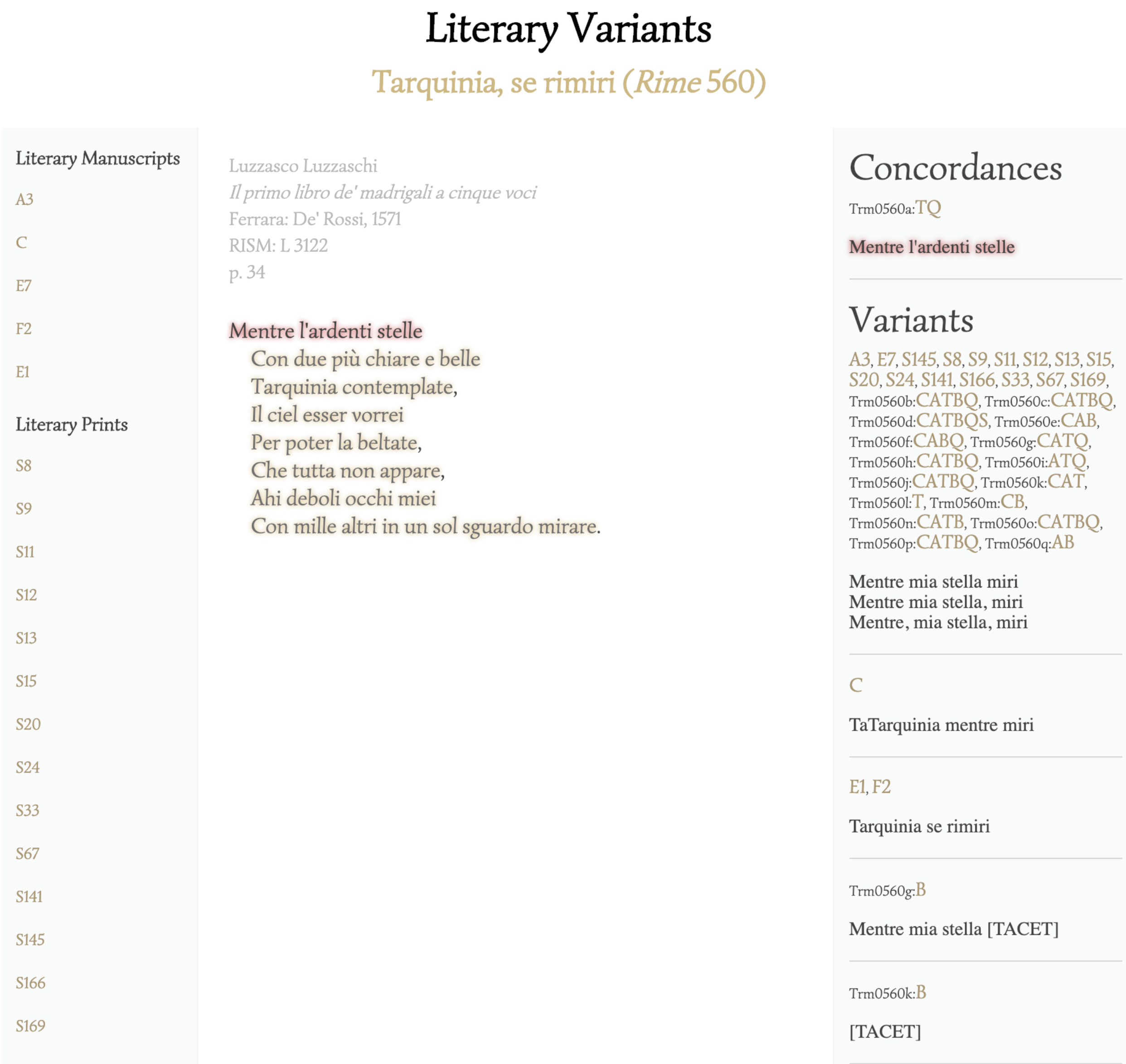


Figure 1: Literary variants for “Tarquinia, se rimiri” (*Rime* 560)

This tool for the study of literary variants can also point to possible manipulations of literary texts by composers. A notable example is that of Monteverdi’s rendition of the *madrigale libero* “Al lume delle stelle” (*Rime* 246), which, as Tim Carter has pointed out, features two additional lines that do not appear in any of the extant literary and musical sources [7].[28]

28 http://www.tassomusic.org/variants/?id=Trm0246

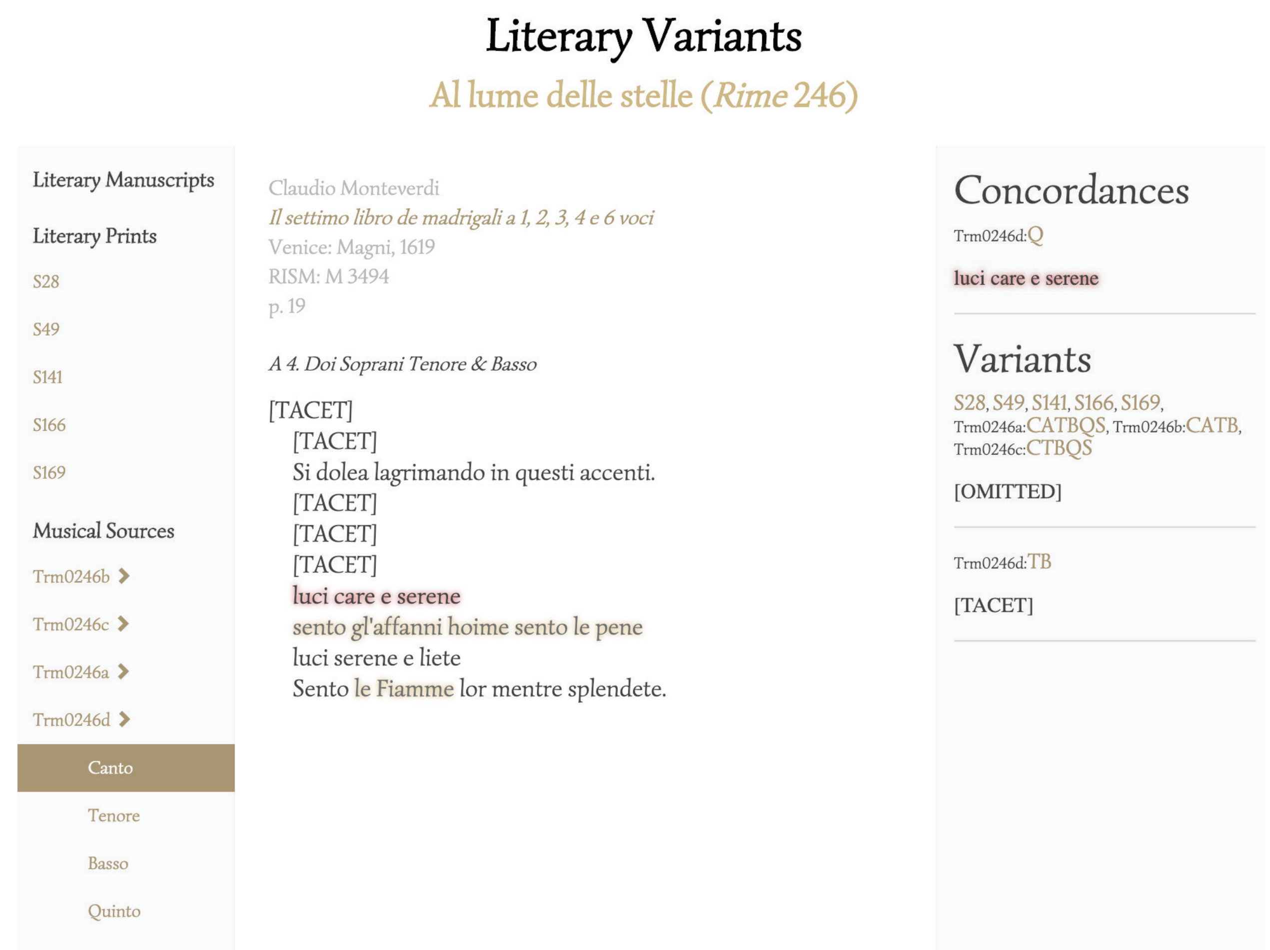


Figure 2: Literary variants for "Al lume delle stelle" (*Rime* 246)

The diplomatic transcriptions of the poetic texts and the marked-up variants are encoded in TEI, using a schema developed specifically for the Tasso in Music Project by Raffaele Viglianti (Maryland Institute for Technology in the Humanities, University of Maryland). Like the musical data, the content for textual variants, too, is stored in ATON files.[29] These are arranged by poem, with metadata and diplomatic transcriptions drawn from each literary and musical source for that poem.[30] Below, for example, is data from the literary manuscript **A3** for the poem "Tarquinia, se rimiri" (*Rime* 560):

```
@@BEGIN: VARIORUM
@CATALOGNUM:    Trm0560
@ID:            A3
@TYPE:          manuscript
@SMSIGLUM:      A<sub>3</sub>
@PAGE:          263v-264r
@PARATEXT:
@VERSE:
    {Mentre mia stella miri}
    {I bei celesti giri}
    {Il ciel esser vorrei}
```

29 https://github.com/TassoInMusicProject/tasso-website/tree/gh-pages/data/variorum

30 For a sample ATON file ("Tarquinia, se rimiri," Rime 560), see https://github.com/TassoInMusicProject/tasso-website/blob/gh-pages/data/variorum/Trm0560.aton

```
        {Perche ne gl'occhi miei}
        {Fiso tu rivolgessi}
        {Le tue dolci faville}
        {Io vaghegiar potessi}
        {Mille bellezze tue con luci mille}.
    @@END: VARIORUM
```

The @VERSE section of the entry contains the lines of poetry with analytic markup. The curly braces ({}) indicate variants regions across sources. All poem entries must contain the same variant markup structure. The entries are then compiled automatically into TEI files, with a separate file for each source entry in the originating ATON file,[31] with each directory of TEI files representing one of the source ATON files:[32]

```
<?xml version=“1.0“ encoding=“UTF-8“?>
<?xml-model href=“tassotei.rng“ type=“application/xml“ schematypens=“http://relaxng.orgns/structure/1.0“?>
<TEI xmlns=“http://www.tei-c.org/ns/1.0“>
    <teiHeader>
        <fileDesc>
            <titleStmt>
                <title>Tarquinia, se rimiri</title>
            </titleStmt>
            <publicationStmt>
                <publisher></publisher>
                <availability status=““></availability>
                <idno type=“local“>A3</idno>
            </publicationStmt>
            <sourceDesc>
                <msDesc xml:id=“A3“>
                    <msIdentifier>
                        <region>Italy</region>
                        <settlement>Milan</settlement>
                        <repository>Biblioteca Ambrosiana</repository>
                        <idno>I.149 inf.</idno>
                    </msIdentifier>
                    <msContents>
                        <msItem>
                            <locus>f. 263v-264r</locus>
                            <p>miscellaneous codex, containing copies of 21 lyr
                            ic poems by Tasso</p>
                        </msItem>
                    </msContents>
                    <history>
                        <origin>
                            <origDate>16th&#x2013;17th centuries</origDate>
                        </origin>
                    </history>
                </msDesc>
            </sourceDesc>
        </fileDesc>
    </teiHeader>
```

31 Such as https://github.com/TassoInMusicProject/variorum/tree/master/data/Trm0560.

32 Such as https://github.com/TassoInMusicProject/variorum/edit/master/data/Trm0560/A3.xml.

```
        <text>
                <body xml:space=“preserve“>
                        <div type=“dedication“>
                                <p xml:id=“p1“></p>
                        </div>
                        <div>
                                <lg>
                                        <l xml:id=“l1“><seg xml:id=“l1v1“>Mentre mia stella miri</seg></l>
                                        <l xml:id=“l2“><seg xml:id=“l2v1“>I bei celesti giri</seg></l>
                                        <l xml:id=“l3“><seg xml:id=“l3v1“>Il ciel esser vorrei</seg></l>
                                        <l xml:id=“l4“><seg xml:id=“l4v1“>Perche ne gl’occhi miei</seg></l>
                                        <l xml:id=“l5“><seg xml:id=“l5v1“>Fiso tu rivolgessi</seg></l>
                                        <l xml:id=“l6“><seg xml:id=“l6v1“>Le tue dolci faville</seg></l>
                                        <l xml:id=“l7“><seg xml:id=“l7v1“>Io vaghegiar potessi</seg></l>
                                        <l xml:id=“l8“><seg xml:id=“l8v1“>Mille bellezze tue con luci mille</
                                        seg>.</l>
                                </lg>
                        </div>
                </body>
        </text>
</TEI>
```

The ATON and TEI encodings contain the same information, although additional information about the manuscripts is inserted from separate ATON metadata files based on the manuscript ID. Note that line breaks in the @VERSE content are significant, and they are mapped to <l xml:id=”l1”> elements that contain each line in the TEI encodings. The variant markup, represented through curly braces, is mapped to <seg xml:id=”l1v1”> elements. TEI elements such as <l> and <seg> are automatically assigned IDs based on the line and variant number on the line, which are automatically calculated from the sequence of lines and variants within the source ATON data. TEI markup that cannot be automatically generated is embedded directly within the verse lines, such as the use of <del> to indicate deleted (crossed-out) text in the manuscript source. For example, the ATON line from the **C** manuscript source for “Tarquinia, se rimiri” (*Rime* 560)

```
{<del>Ta</del>Tarquinia mentre miri}
```

Is converted into the TEI content:

```
<l xml:id=“l1“><seg xml:id=“l1v1“><del>Ta</del>Tarquinia mentre miri</seg></l>
```

In principle, variant markup could be automatically generated. In practice, however, this could never become a closed system, since exceptions cannot be enumerated. Therefore, for the project, we decided on manual markup of the variants. The lines as they appear in each source are collated adjacently, enabling quick reference and efficient variant markup.[1] Below is the ATON file collation of the opening line of “Tarquinia, se rimiri” (*Rime* 560):

ID:A3	L9	V1:	{Mentre mia stella miri}
ID:C	L28	V1:	{~~Ta~~Tarquinia mentre miri}
ID:E1	L47	V1:	{Tarquinia se rimiri}
ID:E7	L65	V1:	{Mentre mia stella miri}
ID:F2	L84	V1:	{Tarquinia se rimiri}
ID:S8	L136	V1:	{Mentre, mia stella, miri}

1 For each line of Rime 560, see: https://github.com/TassoInMusicProject/tasso-website/blob/gh-pages/data/variorum/diff-input/Trm0560.txt

ID:S9	L155	V1:	{Mentre, mia stella, miri}
ID:S11	L174	V1:	{Mentre, mia stella, miri}
ID:S12	L193	V1:	{Mentre, mia stella, miri}
ID:S13	L212	V1:	{Mentre, mia stella, miri}
ID:S15	L231	V1:	{Mentre, mia stella, miri}
ID:S20	L250	V1:	{Mentre, mia stella, miri}
ID:S24	L269	V1:	{Mentre, mia stella, miri}
ID:S33	L287	V1:	{Mentre mia stella, miri}
ID:S67	L305	V1:	{Mentre mia stella, miri}
ID:S141	L324	V1:	{Mentre, mia stella, miri}
ID:S145	L343	V1:	{Mentre mia stella miri}
ID:S166	L362	V1:	{Mentre, mia stella, miri}
ID:S169	L380	V1:	{Mentre mia stella, miri}
ID:Trm0560a-Alto	L398	V1:	{Mentre l’ardenti stelle}
ID:Trm0560a-Tenore	L417	V1:	{Mentre l’ardenti stelle}
ID:Trm0560a-Quinto	L436	V1:	{Mentre l’ardenti stelle}
ID:Trm0560b-Canto	L455	V1:	{Mentre mia stella miri}
ID:Trm0560b-Alto	L474	V1:	{Mentre mia stella miri}
ID:Trm0560b-Tenore	L493	V1:	{Mentre mia stella miri}
ID:Trm0560b-Basso	L512	V1:	{Mentre mia stella miri}
ID:Trm0560b-Quinto	L531	V1:	{Mentre mia stella miri}
ID:Trm0560c-Canto	L550	V1:	{Mentre mia stella miri}
ID:Trm0560c-Alto	L569	V1:	{Mentre mia stella miri}
ID:Trm0560c-Tenore	L588	V1:	{Mentre mia stella miri}
ID:Trm0560c-Basso	L607	V1:	{Mentre mia stella miri}
ID:Trm0560c-Quinto	L626	V1:	{Mentre mia stella miri}
ID:Trm0560d-Canto	L645	V1:	{Mentre mia stella miri}
ID:Trm0560d-Alto	L664	V1:	{Mentre mia stella miri}
ID:Trm0560d-Tenore	L683	V1:	{Mentre mia stella miri}
ID:Trm0560d-Basso	L702	V1:	{Mentre mia stella miri}
ID:Trm0560d-Quinto	L721	V1:	{Mentre mia stella miri}
ID:Trm0560d-Sesto	L740	V1:	{Mentre mia stella miri}
ID:Trm0560e-Canto	L759	V1:	{Mentre mia stella miri}
ID:Trm0560e-Alto	L778	V1:	{Mentre mia stella miri}
ID:Trm0560e-Basso	L797	V1:	{Mentre mia stella miri}
ID:Trm0560f-Canto	L816	V1:	{Mentre mia stella miri}
ID:Trm0560f-Alto	L835	V1:	{Mentre mia stella miri}
ID:Trm0560f-Basso	L854	V1:	{Mentre mia stella miri}
ID:Trm0560f-Quinto	L873	V1:	{Mentre mia stella miri}
ID:Trm0560g-Canto	L892	V1:	{Mentre mia stella miri}
ID:Trm0560g-Alto	L911	V1:	{Mentre mia stella miri}
ID:Trm0560g-Tenore	L930	V1:	{Mentre mia stella miri}
ID:Trm0560g-Basso	L949	V1:	{Mentre mia stella [TACET]}
ID:Trm0560g-Quinto	L968	V1:	{Mentre mia stella miri}
ID:Trm0560h-Canto	L987	V1:	{Mentre mia stella miri}
ID:Trm0560h-Alto	L1006	V1:	{Mentre mia stella miri}
ID:Trm0560h-Tenore	L1025	V1:	{Mentre mia stella miri}
ID:Trm0560h-Basso	L1044	V1:	{Mentre mia stella miri}
ID:Trm0560h-Quinto	L1063	V1:	{Mentre mia stella miri}
ID:Trm0560i-Alto	L1082	V1:	{Mentre mia stella miri}
ID:Trm0560i-Tenore	L1101	V1:	{Mentre mia stella miri}

ID:Trm0560i-Quinto	L1120	V1:	{Mentre mia stella miri}
ID:Trm0560j-Canto	L1139	V1:	{Mentre mia stella miri}
ID:Trm0560j-Alto	L1158	V1:	{Mentre mia stella miri}
ID:Trm0560j-Tenore	L1177	V1:	{Mentre mia stella miri}
ID:Trm0560j-Basso	L1196	V1:	{Mentre mia stella miri}
ID:Trm0560j-Quinto	L1215	V1:	{Mentre mia stella miri}
ID:Trm0560k-Canto	L1234	V1:	{Mentre mia stella miri}
ID:Trm0560k-Alto	L1253	V1:	{Mentre mia stella miri}
ID:Trm0560k-Tenore	L1272	V1:	{Mentre mia stella miri}
ID:Trm0560k-Basso	L1291	V1:	{[TACET]}
ID:Trm0560l-Tenore	L1310	V1:	{Mentre mia stella miri}
ID:Trm0560m-Canto	L1329	V1:	{Mentre mia stella miri}
ID:Trm0560m-Basso	L1348	V1:	{Mentre mia stella miri}
ID:Trm0560n-Canto	L1367	V1:	{Mentre mia stella miri}
ID:Trm0560n-Alto	L1386	V1:	{Mentre mia stella miri}
ID:Trm0560n-Tenore	L1405	V1:	{Mentre mia stella miri}
ID:Trm0560n-Basso	L1424	V1:	{Mentre mia stella miri}
ID:Trm0560o-Canto	L1443	V1:	{Mentre mia stella miri}
ID:Trm0560o-Alto	L1462	V1:	{Mentre mia stella miri}
ID:Trm0560o-Tenore	L1481	V1:	{Mentre mia stella miri}
ID:Trm0560o-Basso	L1500	V1:	{Mentre mia stella miri}
ID:Trm0560o-Quinto	L1519	V1:	{Mentre mia stella miri}
ID:Trm0560p-Canto	L1538	V1:	{Mentre mia stella miri}
ID:Trm0560p-Alto	L1557	V1:	{Mentre mia stella miri}
ID:Trm0560p-Tenore	L1576	V1:	{Mentre mia stella miri}
ID:Trm0560p-Basso	L1595	V1:	{Mentre mia stella miri}
ID:Trm0560p-Quinto	L1614	V1:	{Mentre mia stella miri}
ID:Trm0560q-Alto	L1633	V1:	{Mentre mia stella miri}
ID:Trm0560q-Basso	L1652	V1:	{Mentre mia stella miri}

Since this poetic line reads differently across sources, the entire line is marked as a variant. Each line of the variant editing file contains four fields: the ID of the source, a line number in the ATON file for the poem that the verse line occurs on, the line number in the poem, and then the transcription of the line as it appears in the source. The line information is used to re-insert the edited variant text for the line back into the primary source file for the variants of the poem.

In the final display of variants, as illustrated in Figure 1, hortographical and punctuation variants are automatically grouped together and counted as concordances:

Mentre mia stella miri
Mentre mia stella, miri
Mentre, mia, stella, miri

By mousing over a source in the right column, however, users are still able to trace the concordant readings of the text exactly as they appear in their respective sources, as shown in the figure below for literary print **S20**.

Figure 3: Display of concordant readings by source

Search tools

Enabled by Humdrum, a music encoding system that is especially conducive to computational analysis [8], musical and textual searches can be run both at the level of the individual work and across the repertoire.[1] The latter type can be especially useful for corpus studies, given the breadth and diversity of the project's repertoire. The results of the repertoire-wide searches can be ranked in a variety of ways, with links to the individual works in which the searched musical patterns (pitch, interval, rhythm) or text are automatically highlighted.

1 http://www.tassomusic.org/search

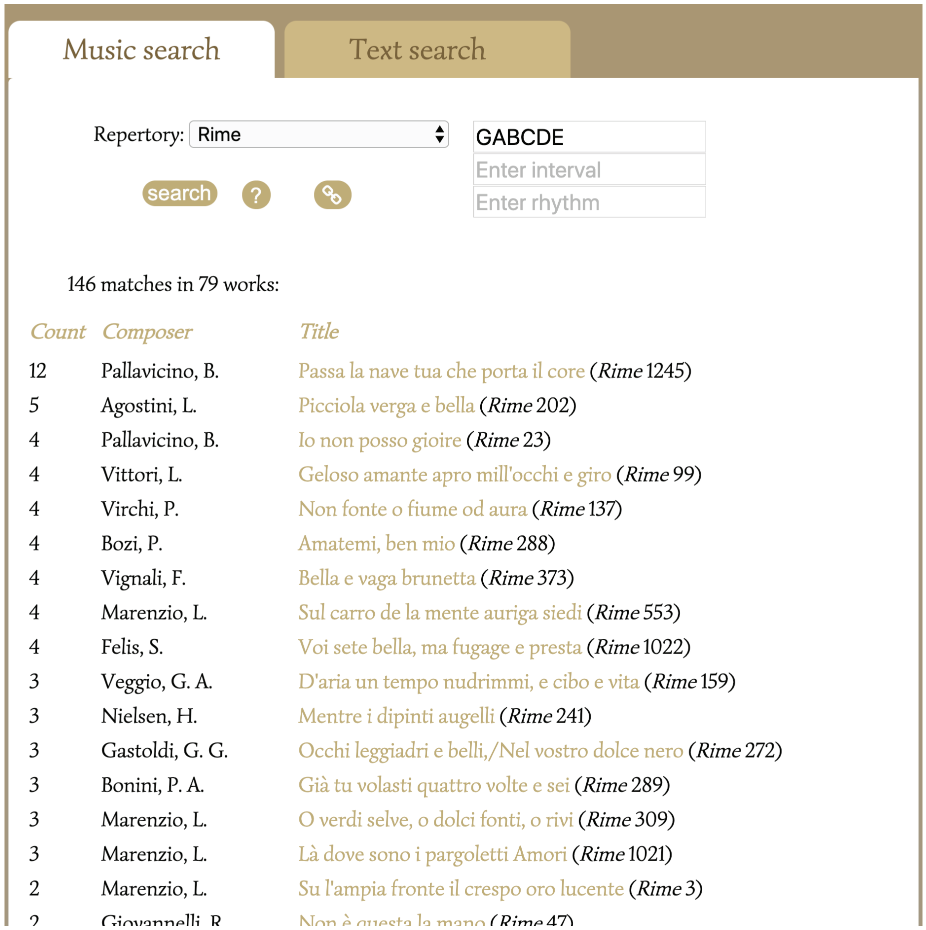


Figure 4: Results of a repertoire-wide pitch search

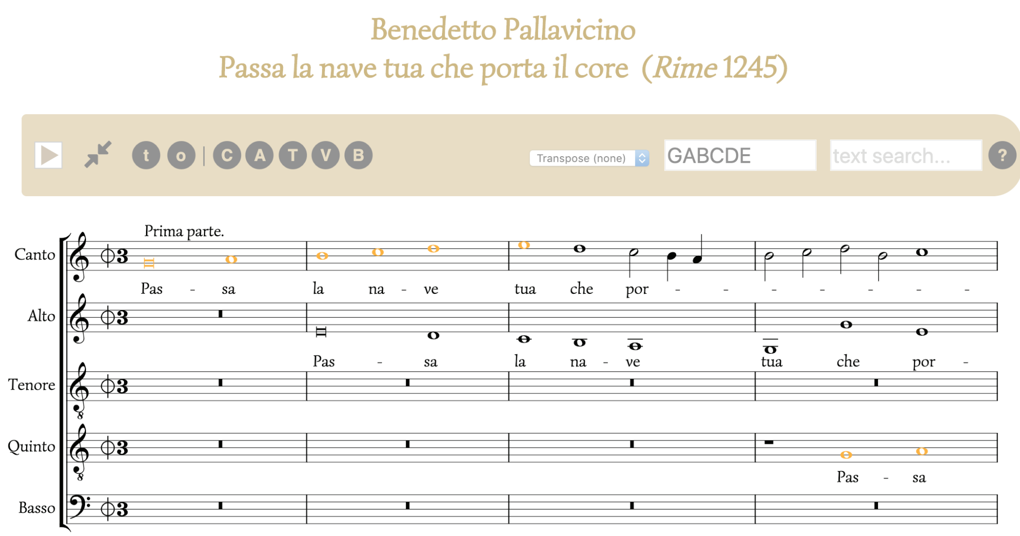


Figure 5: Results of the above pitch search highlighted in the Verovio score

Tools for analysis

The project features several Humdrum-enabled tools that facilitate analysis of the repertoire, with an emphasis on music-text relationships. For instance, a text-extraction tool allows users to visualize the text as it appears in the underlay of the settings, with automatic counting of the occurrences of words.[2] This tool highlights the importance of word repetition in this repertoire and allows users to quickly identify key words.

Canto

La bella e vagha man, <La bella e vagha man,> <La bella e vagha man,> <La bella e vagha man,> <La bella e vagha man,> <La bella e vagha man,> <La bella e vagha man> che le sonore Corde, <che le sonore Corde,> hor leggiera e presta, e prest'hor tarda e grave Percuote e son ne trahe, <e son ne trahe> vario e soave, soave, E'l dolce canto tuo che form'amore, E'l dolce canto tuo, <E'l dolce canto tuo> che form'amore, <che form'amore,> Percuote e son ne trahe, <e son ne trahe vario e soave, soave, E'l dolce canto tuo che form'amore, E'l dolce canto tuo, <E'l dolce canto tuo> che form'amore, che form'amore.

Alto

La bella e vagha man, <La bella e vagha man,> <La bella e vaga man,> La bella e vagha man, <La bella e vagha man,> <La bella e vagha man,> La bella e vagha man che le sonore Corde, che le sonore Corde, <che le sonore Corde, hor leggiera e presta, <e presta,> presta, presta, prest'hor tarda e grave, tardda e grave Percuote e son, Percuote e son ne trahe vario e soave, Percuote e son ne trahe vario e soave, E'l dolce canto tuo che form'amore, E'l dolce canto tuo, E'l dolce canto tuo che forma, che form'amore, Percuote e son, Percuote e son ne trahe vario e soave, Percuote e son ne trahe vario e soave, E'l dolce canto tuo che form'amore, E'l dolce canto tuo, <E'l dolce canto tuo> che form'amore, che form'amore.

Tenore

La bella e vagha man, <La bella e vagha man,> <La bella e vagha man,> <La bella e vagha man,> <Le bella e vagha man,> La bella e vagha, La bella e vagha man che le sonore Corde, <che le sonore Corde,> <che le sonore Corde> e preta, presta, <presta, prest'hor tarde e grave, tarda e grave Percuote e son ne trahe vario e soave, Percuote e son ne trahe vario e soave, E'l dolce canto tuo, E'l dolce canto tuo che form'amore, che form'amore, Percuote e son ne trahe vario e soave, Percuote e son ne trahe vario e soave, E'l dolce canto tuo, <E'l dolce canto tuo> che form'amore, che form'amore.

Quinto

La bella e vagha man, <La bella e vagha man,> La bella e vagha man,> <La bella e vagha man,> <La bella e vagha man,> <La bella e vagha man> che le sonore Corde, che le sonore Corde, <che le sonore Corde> e presta, presta, prest'hor tarda e grave, tarda e grave Percuote e son ne trahe, <e son ne trahe> vario e soave, soave, E'l dolce canto tuo, <E'l dolce canto tuo> che form'amore, che forma, che form'amore, Percuote e son ne trahe, <Percuote e son ne trahe> vario e soave, soave, E'l dolce canto tuo, <E'l dolce canto tuo> che form'amore, che forma, che form'amore.

Basso

La bella e vagha man, <La bella e vagha man,> <La bella e vagha man,> <La bella e vagha man,> <La bella e vagha man> che le sonore Corde, <che le sonore Corde,> Hor leggiera e presta, presta, prest'hor tarda e grave vario e soave, Percuote e son ne trahe vario e soave, E'l dolce canto tuo che form'amore, E'l dolce canto tuo che forma, che form'amore, vario e soave, Percuote e son ne trahe vario e soave, E'l dolce canto tuo che form'amore, E'l dolce canto tuo che forma, che form'amore.

Word frequencies

sort numerically data

Count	Word	Count	Word	Count	Word	Count	Word	Count	Word	Count	Word
23	amore	24	e'	31	la	5	prest'	13	sonore	24	tuo
32	bella	23	form'	14	le	11	presta	6	tarda	1	vaga
24	canto	5	forma	3	leggiera	1	preta	1	tardda	31	vagha
41	che	8	grave	31	man	20	soave	1	tarde	16	vario
13	corde	8	hor	18	ne	20	son	18	trahe		
24	dolce	24	l	17	percuote						

Figure 6: Text extraction for Sebastiano Raval's "La bella e vaga man che le sonore" (*Rime* 862)

Likewise, a melisma tool allows users to study the occurrence of melismatic writing across the repertoire, ranked by composer or melisma score and with links to the dynamic scores in which the melismas are automatically highlighted.[3]

2 http://www.tassomusic.org/lyrics/?id=Trm0862a

3 http://www.tassomusic.org/analysis/melisma/

Melisma analysis

▶ Description

Sort by: Composer

Trm0202a

Composer: Agostini, Lodovico / 1582

Title: *Picciola verga e bella* (rime 202)

Score: 20.9% of notes are in a melisma, C (C I): 25.2%, VI (C II): 25.2%, A (C III): 21.1%, V: 22.5%, T: 19.9%, B: 10.2%

View: 2+ notes, 3+ notes, 4+ notes, 5+ notes

Words:

C (C I): verga, trionfale, cresci, rinverde, novella, errori, dolci, amori

VI (C II): bella, cresci, rinverde, verde, errori, errori, amori

A (C III): bella, d'alloro, cresci, pianta, verde, errori, amori

V: cresci, pianta, verde, errori, amori, errori, pargoletti, amori

T: verga, cresci, eguale, amori

B: bella, bella, onde, rinverde, pargoletti

Figure 7: Melismatic analysis of Lodovico Agostini's "Picciola verga e bella" (*Rime* 202)

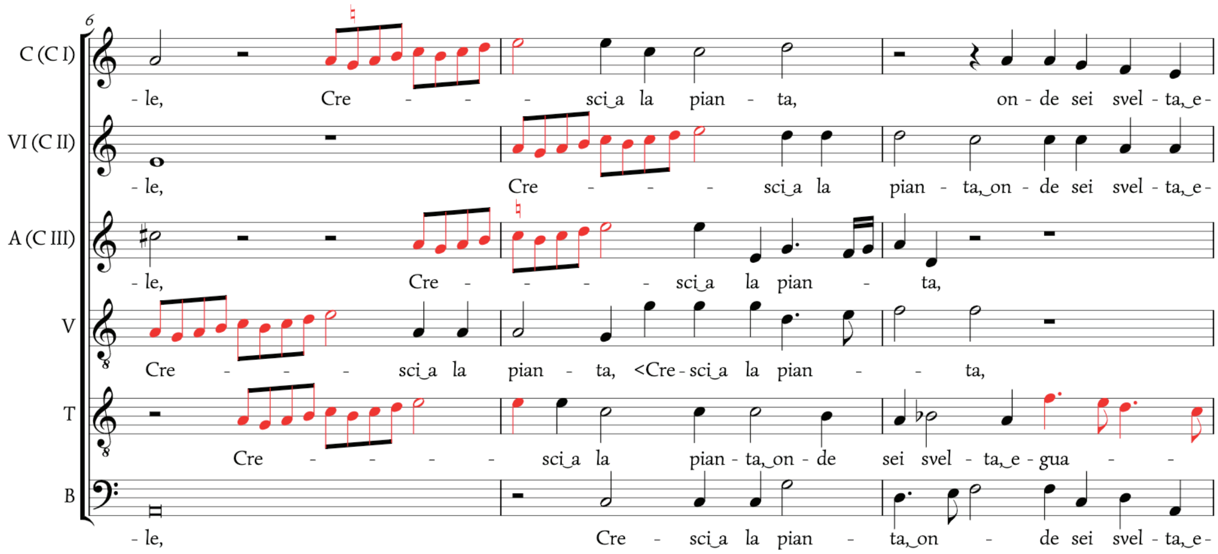


Figure 8: Highlighted melismas in the Verovio score of Agostini's "Picciola verga e bella" (*Rime* 202)

Equally useful for the statistical study of music-text relations is the ranking of the settings by their music/text ratio, that is, a comparison of the length of the musical settings expressed in minims (half-notes) versus length of the poems expressed in number of syllables. This allows users to determine how rapidly composers move through a poetic text, or, conversely, how much they indulge in setting it to music.[4] The example below points to a composer whose music-text ratio is consistently low, namely Filippo di Monte, whose Tasso settings stand out for their musical compactness, achieved by eschewing textual/musical repetition and extensive word painting.

4 http://www.tassomusic.org/analysis/syllable

1.17	104	89	Trm0047o	Monte	1591	4/2	*Non è questa la mano*
1.16	108	93	Trm0048e	Monte	1590	4/4	*Amor l'alma m'allaccia*
1.49	128	86	Trm0075d	Monte	1591	4/2	*Poiché madonna sdegna*
0.85	76	89	Trm0090b	Monte	1588	4/2	*Mentre nubi di sdegno*
0.88	120	137	Trm0093b	Monte	1591	4/2	*Disdegno e gelosia*
1.65	124	75	Trm0165e	Monte	1591	4/2	*Vorrei lagnarmi a pieno*
1.49	112	75	Trm0166f	Monte	1591	4/2	*Se taccio, il duol s'avanza*
1.14	176	154	Trm0183d	Monte	1591	4/2	*Di nettare amoroso ebro la mente*
0.96	100	104	Trm0257a	Monte	1588	4/2	*Ardi, Amor, se ti piace*
1.27	100	79	Trm0279a	Monte	1591	4/2	*Al discioglier d'un groppo*
1.56	100	64	Trm0284e	Monte	1591	4/2	*Questa vostra pietate*
1.31	80	61	Trm0333g	Monte	1590	4/2	*Gelo ha Madonna il seno*
1.35	92	68	Trm0342e	Monte	1591	4/2	*Già fu dolce mia speme*
1.25	192	154	Trm0366a	Monte	1586	4/2	*Odi, Filli, che tuona; odi che 'n gelo*
1.32	204	154	Trm0380a	Monte	1591	4/2	*Viviamo, amianci, o mia gradita Jelle*
0.96	148	154	Trm0389b	Monte	1586	4/2	*Donna, se ben le chiome ho già ripiene*
1.4	216	154	Trm0395c	Monte	1586	4/2	*In un bel bosco leggiadre fronde*
1.54	88	57	Trm0418g	Monte	1586	4/2	*Ardi e gela a tua voglia*
1.8	108	60	Trm0560h	Monte	1591	4/2	*Tarquinia, se rimiri*
0.78	120	154	Trm0592a	Monte	1588	4/2	*Negli anni acerbi tua purpurea rosa*
1.05	108	103	Trm0602c	Monte	1590	4/2	*Caro amoroso neo*
1.2	120	100	Trm0611n	Monte	1591	4/2	*La bella pargoletta*
1.16	92	79	Trm0614h	Monte	1590	4/2	*Al vostro dolce azzurro*
1.32	104	79	Trm0615b	Monte	1586	4/2	*Al tuo vago pallore*
2.16	332	154	Trm0865a	Monte	1590	2/1	*Deh! perché amar chi voi con pari affetto*
0.99	152	154	Trm1688d	Monte	1589	4/2	*Padre del cielo, or ch'atra nube il calle*

Figure 9: Filippo di Monte's music-text ratio

Due to limitations of processing all scores in a timely manner, the raw data for analysis pages is typically processed offline and can be seen on the website's Github repository.[5] This raw data is then transformed into more readable results on the analysis webpages.[6]

Conclusions

Through its content, as well as through its digital features, the Tasso in Music Project benefits a wide audience encompassing music historians and theorists, literary scholars, linguists, performers, and more generally anyone with an interest in the intersection of music and poetry. In addition, the project provides a new model for editions of Italian madrigals, one that restores the centrality of poetry in this repertoire. The project also offers a model for the possible integration of music and textual encoding in a single platform, which could be adopted for editions and repositories of other vocal repertoires, ranging from opera to Lied. Our future plans for the project will focus on the expansion of its interdisciplinary scope through the development of additional tools for the study of music-text relations. These will include tools for the study of the relationship between the prosody of the poetic texts (accented and unaccented syllables, primary and secondary accents) and rhythmic durations, as well as the relationship between literary and musical syntax (e.g., the musical treatment of enjambments).

5 https://github.com/TassoInMusicProject/tasso-website/tree/gh-pages/analysis

6 https://www.tassomusic.org/analysis

Works cited

[1] Ricciardi, Emiliano, ed. *"Qual musico gentil": New Perspectives on Torquato Tasso and Early Modern Music*. Turnhout: Brepols Publishers, forthcoming.

[2] Balsano, Maria Antonella, and Thomas Walker, eds. *Tasso, la musica, i musicisti*. Florence: Olschki, 1988.

[3] Ricciardi, Emiliano. "The Musical Reception of Torquato Tasso's *Rime*, 1571-1620". Stanford University, 2013. PhD dissertation.

[4] Sapp, Craig Stuart. "Verovio Humdrum Viewer" presented at the Music Encoding Conference, Tours, France, May 16-19, 2017.

[5] Newcomb, Anthony. "Texts, Translations and Commentaries" in *Luzzaschi, Luzzasco, Il primo libro de' madrigali a cinque voci (Ferrara, 1571)*, ed. A. Newcomb. Middleton, WI: A-R Editions, 2010.

[6] Piperno, Franco. "La tradizione musicale delle rime di Torquato Tasso (1571-1581)" Ellisse 2 (2013), 25-63.

[7] Carter, Tim. "From Diegesis to Mimesis (and Back): Monteverdi, Tasso, and the Seventh Book of Madrigals (1619)" presented at the Tasso in Music Conference, University of Massachusetts Amherst, April 17-19, 2020. An expanded version of the talk is forthcoming in Ricciardi, ed. "*Qual musico gentil*", Brepols Publishers.

[8] Sapp, Craig Stuart. "Computational Methods for the Analysis of Musical Structure". Stanford University, 2011. PhD dissertation.

Probstücke Digital – A Critical Digital Edition of Johann Mattheson‘s 24 Probstücke of the Ober-Classe

Niels Pfeffer
HMDK Stuttgart
niels.pfeffer@gmail.com

Klaus Rettinghaus
Leipzig University
klaus.rettinghaus@gmail.com

Introduction

In 1731, Johann Mattheson writes in the preface to the *Große Generalbass-Schule*:

> "The complaint, however, which I made in the first edition of this Organisten-Probe about the badly printed notes, is still in its full strength, and patience is the only remedy."[1]

Probstücke Digital is an open and critical digital edition project of the 24 test pieces of the Ober-Classe ("upper class") by Johann Mattheson and as such an example for the use and application of MEI and TEI in an integrated environment.[2] After almost 300 years it also seeks to finally give remedy to Mattheson's complaint by editing his Probstücke[3] and by providing perhaps a little more than merely "prettifying" the original print.

"Incomplete notation"

The musical material that is being edited consists of commented partimenti, unrealized bass lines, that are conceptually open and yet-to-be-finished drafts, skeletons rather than self-contained works, leaving a vast space for creative inventions of the performer.[4]

From a performer's view, space is utterly important in the process of working with the Probstücke – space to sketch and to work out different melodic, counterpuntal or harmonic ideas or realizations of different complexity. Thus, one of the first goals of the present edition is to provide the performer with a virtually unrestricted amount of space for the creative process.

1 "Die Klage aber, so ich in der ersten Auflage der Organisten-Probe, wegen der schlechten Druck-Noten geführet habe, ist noch in ihren vollen Kräfften, und die Gedult das eintzige Mittel" [1, p. 156].

2 For aspects of combining MEI and TEI see e.g. [2].

3 Already in 1965, Wolfgang Fortner published a modern edition [3] of the Mattheson's Probstücke of the Mittel-Classe – a second volume with the pieces of the Ober-Classe was announced by the publisher but never published since then. Fortner's edition is of great practical use e.g. by providing an additional staves and by embedding all of Mattheson's suggestions into the score. However, many of his simplifications do imply realizations that cannot be deduced from the original text.

4 In reference to [4, p. 68], we call this phenomenon "incomplete notation". Already [5, p. 440] considered the Probstücke as "foundations of independent improvisation". For a more recent study on the Probstücke, in particular on their pedagogical aspects, see [6].

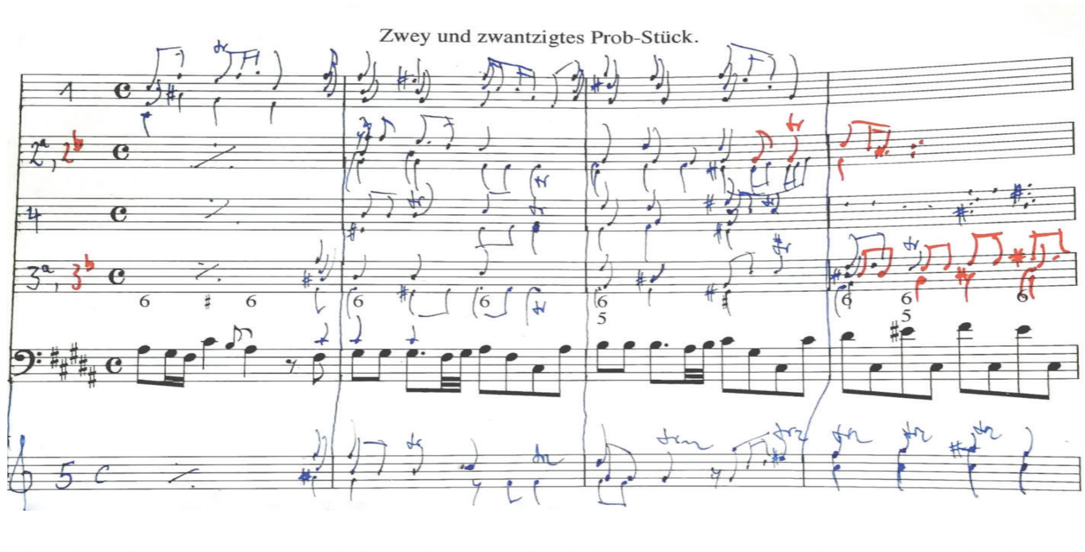



Figure 10: The first four measures of the 22nd *Probstück* as an example for the practical use of empty staves.

This is made possible by the option of adding an arbitrary amount of staves above or below the original bass line.[5]

Readings, editorial regularizations and additions

Although by now there are many digital edition projects on the web to be discovered, surprisingly many of them seem to use MEI only as a basis for an engraving with Verovio.[6] The goal of this project is to provide a full critical edition in pure MEI and TEI[7] with a correct encoding of variants in reading,[8] editorial additions and regularizations, which require heavy manual encoding.

The general approach of *Probstücke Digital* is to provide diplomatic transcriptions of both text and music, and letting the user choose whether or not to modernize the original – regarding accidentals, clefs, orthography etc.

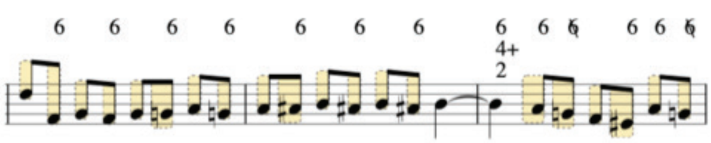



Figure 11: A passage from *Probstück 1,* as displayed with the options to show only original accidentals and to hide accidentals supplied by the editor.

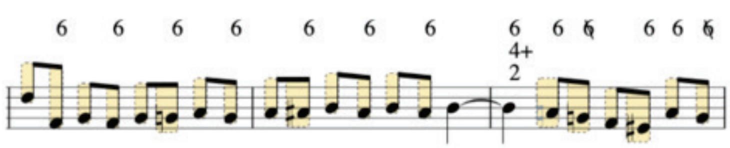


Figure 12: The same passage rendered as a “modern” score, without displaying the originally repeated accidentals and with cautionary accidentals supplied by the editor.

5 Technically achieved by performing an XSL transformation.

6 Verovio (https://www.verovio.org) is an open-source library for engraving MEI music scores into SVG.

7 The edition uses a slightly adapted TEI customization based on the DTA Base format [7].

8 In particular the different readings of the Exemplarische Organisten-Probe [8] and Große Generalbaß-Schule [1].

Other idiosyncrasies of 18th century prints can be perfectly addressed by a digital edition, e.g. by providing tools to automatically replace typographic peculiarities of 18th-century prints like the long s (ſ) or the umlauts with a superposed e (ů) as well as potentially unfamiliar clefs in the score with their modern equivalents. Where possible, these transformations are achieved using pure CSS, but as soon as more heavy interventions in the musical text are required (such as replacing ancient clefs) XSL transformations are performed on the original encoding.

At the same time, many of Mattheson's comments require an editorial supplement or explanation – may it be only the correction of miscounted bar numbers, pseudonyms that can be identified with contemporaries of Mattheson as well as hints on disputes and arguments of the time that Mattheson is referring to or was involved in, assuming the reader's knowledge of these quarrels.

Maximum of visible information

In the common practise, scholarly music editions aim to provide a clean Urtext, that – from a performer's perspective – "bans" a large portion of the actual information on the text into a separate critical apparatus, which is barely looked into by a non-musicologist. Rather than hiding, Probstücke Digital tries to lay open as much information and material as possible at the spot.[9]

Lessons

This includes the linking and presentation of additional material, such as transcriptions of oral lessons, theoretical analyses, realizations and recordings.

Lessons are encoded in TEI as transcriptions of spoken material. The transcription of a lesson given by Robert Hill in Freiburg 2019 shall serve as an example.

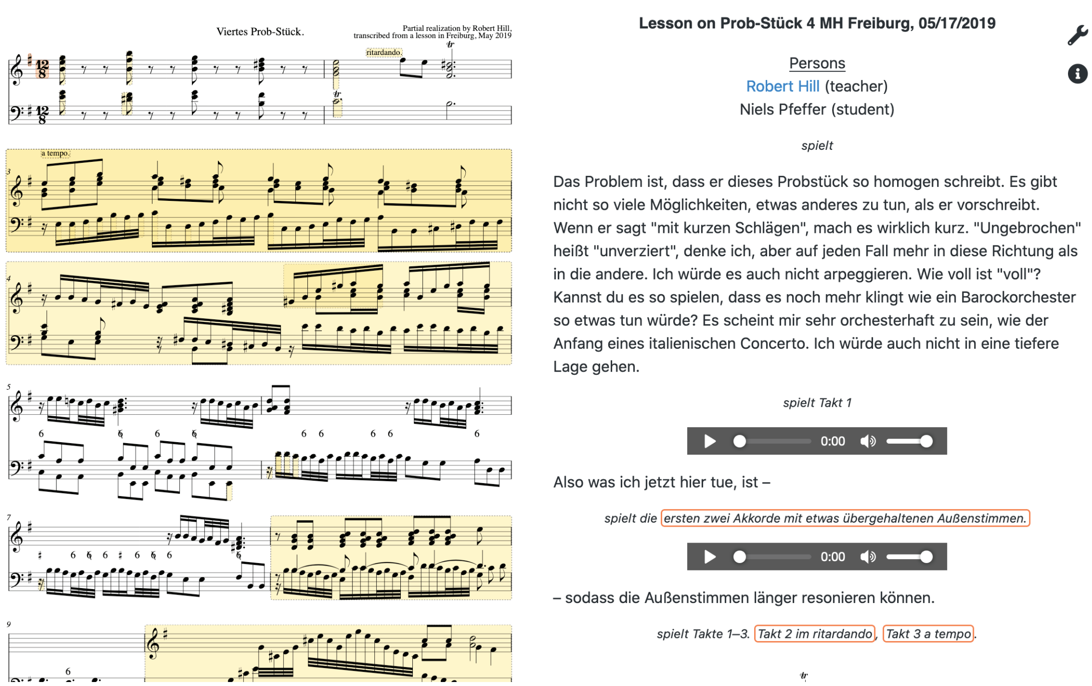


Figure 13: A lesson by Robert Hill on Probstück 4.

9 In that regard, some 19th century "critical" editions like Hans Bischoff's edition of the Well-tempered Clavier ([9] and [10]) may serve as examples of a practise where all the information on sources and different readings are directly integrated into the score itself.

This transcription contains references between the spoken material and the score – as it is present on the score stand during the lessons – may they be explicitly pronounced or only implied. As soon as for the purpose of demonstration the harpsichord comes into play, a transcription of that particular example as well as an corresponding audio fragment is made available.

Realizations

Based on the idea that the Probstücke provide a canvas that can be filled with arbitrarily complex realizations, Probstücke Digital provides examples for such realizations. Since these may alter the original and deviate from it rather much, they are encoded as independent documents.

Just like lessons, realizations can also include a corresponding recorded audio.[10]

Key characteristics

Arno Forchert considered the demonstration of the advantages of the “new” system of major-minor-tonality to be the main purpose of the pieces of the *Große Generalbaß-Schule* – against the proponents of the traditional system based on modality.[11] In that respect, Mattheson’s goal is to give two complete cycles of 24 pieces in all possible major and minor keys. Closely related to his thoughts on tonality are the characteristics of keys which he attempted to set down in the *Neu-eröffnete Orchestre* (1713) [12] – next to characteristics of all the meter signatures. Both are made accessible for each Probstück with an overlay on the key signature that displays Mattheson’s characterisation of that present key.

Technically, these key and meter characteristics are edited as separate TEI encodings, which are included into the Probstück based on the on key signature and meter signature found in the <scoreDef>-element.

Facsimile linking

Furthermore, *Probstücke Digital* provides linking to and presentation of the digitized sources[12] as either full-page facsimiles embedded with Mirador[13] or as the extracted zone of a particular measure or paragraph.

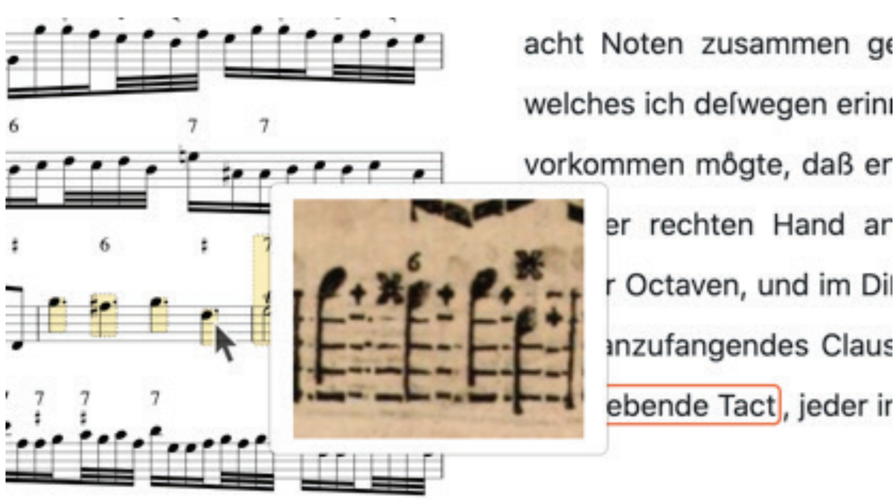



Figure 14: Example of a measure that is associated with the corresponding zone of the original print.

10 See e.g. the realization of Probstück 10.

11 [11, p. 205f.]

12 Utilizing the International Image Interoperability Framework https://iiif.io with images courtesy by the Bavarian State Library https://www.bsb-muenchen.de/en/

13 Mirador is an open-source, web based, multi-window image viewing platform https://projectmirador.org

These regions are encoded in the source files with the corresponding coordinates using MEI's and TEI's facsimile and zone elements. When deploying, Probstücke Digital extracts those zones and turns them into IIIF annotation lists that annotate the digital IIIF "canvases" as provided by the Bavarian State Library.


```
{
  „@context“: „http://iiif.io/api/presentation/2/context.json“,
  „@id“: „https://probstuecke-digital.de/iiif/2/annotation/measure-4“,
  „@type“: „oa:Annotation“,
  „motivation“: „sc:painting“,
  „resource“: {
    „@id“: „http://probstuecke-digital.de/view/2/mattheson/secondEdition#m-199“,
    „@type“: „dctypes:Text“,
    „format“: „text/html“
  },
  „on“: „https://api.digitale-sammlungen.de/iiif/presentation/v2/bsb10598495/can-
vas/392#xywh=951,971,927,347“
}
```


Figure 15: Above an example of a generated IIIF annotation linking a region on the facsimile canvas with the corresponding measure on *probstuecke-digital.de*.

Indices

Mattheson often refers to works by other composers or texts from older music theorists. A modernized bibliography and names index based on authority controlled data will help scholars to find and get into those external sources and musicians to find links between Mattheson's musical material and material of composers he regarded as exemplary and refers to rather often – such as Keiser, Telemann, Heinichen, Mossi, or dall'Abaco etc.

Technical components[14]

eXist-Db is used as a database for storing the encodings as well as processing XSL transformations and XQueries.	server-side
express.js in a node.js environment as a server and router between client and database.	server-side
CETEIcean and Verovio for rendering the TEI and MEI encodings.	client-side
Mirador for rendering facsimile images.	client-side

Licenses

The complete edition is available under Creative Commons Licenses and all used software is available under free licenses. The complete software package and the edition are available on GitHub.[15] We will examine what parts of our software could be useful to other projects and will release them independently eventually.

14 https://github.com/TEIC/CETEIcean. For more about CETEIcean see [13].

15 https://github.com/pfefferniels/probstuecke-digital

Works cited

[1] Mattheson, Johann. *Große Generalbaß-Schule.* Hamburg, 1731.

[2] Viglianti, Raffaele, et al. "Visualizing Fedora-managed TEI and MEI documents within Islandora" Code4Lib 44 (2019). https://journal.code4lib.org/articles/14532

[3] [3] Mattheson, Johann. *Grosse Generalbass-Schule* 1731, neu hrsg. und bearbeitet von Wolfgang Fortner. Mainz, 1956.

[4] [4] Sanguinetti, Giorgio. "A Partimento in Classical Sonata Form by Giacomo Tritto", in *Das flüchtige Werk. Pianistische Improvisation der Beethovenzeit,* eds. Michael Lehner, Nathalie Meidhof, and Leonardo Miucci (Musikforschung der Hochschule der Künste Bern, Bd. 12). Schliengen, 2019 , 57–68.

[5] [5] Arnold, Franck Thomas. *The Art of Accompaniment from a Thorough-bass.* Oxford, 1931.

[6] [6] Dijoux, Jean-Christophe. "Matthesons Probstücke als Partimenti?" Schola Cantorum Basiliensis, 2013. MA dissertation.

[7] [7] Haaf, Susanne, Alexander Geyken, and Frank Wiegand. "The DTA "Base Format": A TEI Subset for the Compilation of a Large Reference Corpus of Printed Text from Multiple Sources" Journal of the Text Encoding Initiative 8 (2014/15). doi:10.4000/jtei.1114. https://journals.openedition.org/jtei/1114

[8] [8] Mattheson, Johann. *Exemplarische Organisten-Probe Im Artikel Vom General-Bass*. Hamburg. 1719.

[9] [9] Bach, Johann Sebastian. *Das wohltemperirte Clavier. Erster Theil.* ed. Hans Bischoff. Hannover, 1883.

[10] [10] Bach, Johann Sebastian. *Das wohltemperirte Clavier. Zweiter Theil.* ed. Hans Bischoff. Leipzig, 1884.

[11] [11] Forchert, Arno. "Polemik als Erkenntnisform: Bemerkungen zu den Schriften Matthesons" in *New Mattheson Studies*, eds. George Buelow and Hans-Joachim Marx. Cambridge, 1983, 199-212.

[12] [12] Mattheson, Johann. *Das neu-eröffnete Orchestre.* Hamburg, 1713.

[13] [13] Cayless, Hugh, Raffaele Viglianti. "CETEIcean: TEI in the Browser" in *Balisage: The Markup Conference 2018*. https://www.balisage.net/Proceedings/vol21/html/Cayless01/BalisageVol21-Cayless01.html

MEI and Verovio for MIR: A Minimal Computing Approach

Mark Saccomano
Columbia University
m.saccomano@columbia.edu

Natalia Ermolaev
Princeton University
nataliae@princeton.edu

Abstract

While the increase in digital editions, online corpora, and browsable databases of encoded music presents an extraordinary resource for contemporary music scholarship, using these databases for computational research remains a complex endeavor. Although norms and standards have begun to emerge, and interoperability among different formats is often possible, researchers must devote considerable time to discover, learn, and maintain the skill sets necessary to make use of these resources. This talk will discuss our work with the Serge Prokofiev Archive and the creation of a prototype to browse, display, and play notated music from Prokofiev's notebooks via a web browser. The project is an example of how using the principles of minimal computing can reduce the burden of technological expertise required to both disseminate and access encoded music.

The archive

The Serge Prokofiev Archive,[16] housed at Columbia University, contains more than 17,500 diverse items: music manuscripts, letters, scores, financial documents, notebooks, photographs, and recordings. Originally a personal collection amassed by Prokofiev's widow Lina, the materials were first established as an archive in 1994 at Goldsmith's College in London. As the archive grew, a complex, intricate, and item-level descriptive apparatus evolved alongside it. By the time the collection came to Columbia, the archival items were accompanied by hundreds of metadata files in a wide variety of formats, including Word documents, spreadsheets, text files, PDFs, Endnote databases, Access databases, MARC records, and various XML encodings.

Typically, archival collections are accessed through an online Finding Aid, which users often find not only difficult to use, but whose underlying structure and interface can obfuscate the richness of a collection. The blocks of narrative and long lists of items found in a finding aid, especially in a collection of our scope, are a barrier to true discovery. We sought to improve the experience of navigating a large archival collection by affording users the opportunity to make new, spontaneous discoveries.

Our *Serge Prokofiev Archive as Data*[17] project was guided by two important conceptual shifts in the library and archives profession. First is the "Collections as Data" movement, which encourages reframing the digital object itself as data [1].[18] The second is Kate Theimer's notion of "archives as platform," a move away from locating value exclusively in the objects of a collection to the impact collections have on people and communities [2]. In Theimer's view, the notion of an archive includes the tools and technologies that help users interact with it in creative ways that add value to their lives and experiences.

Accessible technology and minimal computing

Because we were looking for solutions that could be adapted for researchers with varying skill sets and with different computing needs, we tested out a variety of freely available software to store, structure, clean, analyze and display our data. Also, we had no budget: necessity dictated that we seek out non-proprietary tools. Thus, we placed ourselves in the position of many researchers (independent and graduate student researchers in particular) looking for ways to disseminate their work to a wider audience. Following this path, we soon

16 https://findingaids.library.columbia.edu/ead/nnc-rb/ldpd_10815449
17 https://mss2221.github.io/spademo/
18 See also https://collectionsasdata.github.io/statement/

became introduced to the principles of minimal computing and discovered their applicability to our own project's goals.

Minimal computing[19] is a design philosophy that seeks to maximize access to digital materials through reducing reliance on specific hardware and software requirements [3]. Organized around the question "What do we need?", Alex Gil describes minimal computing as a conscious effort to "harness the new media in smart, ethical and sustainable ways." In addition to reducing reliance on multiple, and opaque, processes, minimal computing also implies "learning how to produce, disseminate and preserve digital scholarship ourselves, without the help we can't get [4]." This DIY approach helps minimize dependence on institutional resources and funding, as well as proprietary tools (which, in addition to their cost, often require a high level of expertise as well).

One of the first steps we took was to avail ourselves of systems and workflows with ample documentation. We were also cognizant of the advisability for scholars to publish digital materials in a versatile format that requires little or no maintenance and can easily be ported to other systems. This way, digital materials remain accessible even as technology develops in ways that are impossible to foresee today. We soon created a repository for the Serge Prokofiev Archive as Data project on GitHub and created a static website for display on GitHub Pages using a Jekyll template. Because a static site does not require knowledge of server operations or database design, it simplifies the task of individual researchers to disseminate their work.

For the musical component of our project, we wanted to create not only an attractive front end and simple user interface, but a simple back end as well. The idea was to provide a repository of encoded music that could not only be seen but heard—a difference that could make such a repertory valuable beyond the specialized scholar in computing or musicology. Aficionados and researchers in other fields who may not be able to read code or read music could nonetheless hear the music in Prokofiev's manuscripts—and could hear for themselves the jagged rhythms and unexpected chromatic alterations that are hallmarks of his style. We also developed a simple workflow for creating the encoded files (one very similar to the process now detailed in the tutorial "Introduction to the Music Encoding Initiative"[20] by Anna Kijas and Raffaele Viglianti). To publish to the encoded materials, we used our GitHub website and Jekyll template.

The notebooks

One of the highlights of the collection are Prokofiev's notebooks. Here, in an interview transcript from the archive, Prokofiev's widow Lina described how he used the notebooks in his creative process.

> SP never stopped creating…. At the most unexpected moment, at the most unusual circumstances—during a conversation or while walking—he would make a note of a new theme in a special notebook he kept in his pocket or on any scrap of paper or on his cuff—on paper napkins in a restaurant. Then on returning home he would copy the themes into a more permanent notebook.

The sketches we display on the site are from these "more permanent" notebooks Lena mentions. We began by simply browsing through the notebooks and taking some pictures. Displayed in a web exhibit, these images would be interesting on their own. But we also knew that by adding the sounded music represented by these scores, we would greatly increase the usefulness of these notebooks to scholars, as well as to the general public. Not all musicologists and music theorists have sufficient musicianship skills to fluently imagine the sound of notated music. For archival materials such as unlabeled sketches, this can aid in identification of fragments, suggesting how and where they might have been used in published scores.

MEI was chosen as the encoding format, not only because of its adaptability and increasingly common use in digital musicological projects, but also due to the availability of Verovio, an engraving library that can be used to display and play MEI files in a web browser. As these were short, handwritten passages of only a few measures each, they were entered manually into the music notation program Sibelius. (Because they were written by hand, an OCR program would likely not have been the most efficient method of encoding). Next, the files were exported to MusicXML using the export function of Sibelius. To convert the MusicXML files to MEI,

19 https://go-dh.github.io/mincomp/

20 https://dlfteach.pubpub.org/pub/intro-mei/release/1

we used the automated converter[21] available on the Verovio website. This worked extremely well and yielded excellent results. The light editing that remained to be done was mostly for aesthetic purposes of display. The editing was done in Atom, using the MEI-Tools-Atom[22] package, which renders MEI in a separate pane within the application.

Once the MEI files were checked and polished, they were uploaded to the GitHub repository. The challenge, then, at this stage, was to create page templates that would incorporate Verovio. Although it took many attempts to pull everything together, the results were encouraging, and a prototype was developed that could display an engraved version of the score derived from a digitally encoded version of the manuscript, as well play the score in a browser using a simple interface:

https://mss2221.github.io/spademo/sketches/

Implementation

Development challenges proved to be formidable. While finding appropriate tools for coding, display and playback of manuscripts was reasonably easy, getting them to work together was exceedingly complex. Documentation, though rich, can be dense; the largest impediment to timely progress is access to consultant who can assist in troubleshooting. Without this, the plethora of manuals and tutorials become an obstacle to learning, creation, and design. (Think of the myriad articles, tips, and guidelines many of us received back in March of this year on how to migrate our courses online—such a wealth of material can be overwhelming). Even with access to university assistance, this site took nearly a year to assemble. However, the skills to use Github and Jekyll are within reach via ground-up tutorials available from such sites as The Programming Historian.[23]

Difficulties still remain, specifically, those arising from technical solutions that push the limits of common browser capabilities. For example, problems with audio playback, such as web MIDI players clipping notes (due to possible buffering or threading issues) have driven developers on some projects to insert an extra musical object into their encoded scores.[24] We also encountered this clipping problem, and were only able to come up with a temporary work around through a laborious trial-and-error process. To ensure the MEI would play properly in the browser, a <space> element or "dummy" <note> event with @visible="false" had to be inserted before the first and final notes in order for them to be heard. Such inelegant solutions are highly undesirable for an archival representation of a manuscript. Presumably, improvements in how browsers and system players handle MIDI will soon make such workarounds unnecessary. In the meantime, these ad hoc solutions need to be specially commented in the MEI files.

Extensibility and Future Directions

Sample sites using MEI, Verovio and Ed template

El corrido mexicano	https://mss2221.github.io/corridosEd/
Serbian hymns	https://mss2221.github.io/zagreb/

In order to test the extensibility of this project, we tried it out with texted music in a special Jekyll template for minimal literary editions, "Ed."[25] developed by Alex Gil and associates. The resulting sites showed the flexibility of the Ed theme to handle some of the more complex requirements of Verovio and web MIDI, while still remaining a project that could be managed by a single researcher. They also demonstrate the utility of our chosen suite of open source tools for musicologists, music theorists, and music archivists. In the future, we

21 https://www.verovio.org/musicxml.html

22 https://atom.io/packages/mei-tools-atom

23 https://programminghistorian.org/ We are particularly indebted to Amelia Visconti for her Jekyll tutorial https://programminghistorian.org/en/lessons/building-static-sites-with-jekyll-github-pages

24 https://github.com/cuthbertLab/music21/issues/332

25 https://elotroalex.github.io/ed/

hope to incorporate search tools for specific series of notes and an analytical component that could be used to identify the stylistic traits of a corpus.

A note about program evaluation: one aspect of design that is often overlooked in digital musicology projects is user testing. As noted by David Weigle in his study of the academic use of digitized online resources [5]: "the needs and behaviours of musicologists in particular remain relatively underexplored". This is not just an issue in musicology. The statement made by Warwick, et al. [6] in 2008 (cited by Murray and Wiercinski [7]) rings true today in 2020: "User testing, like disseminating information, is a skill that most humanities scholars have not acquired". However, as Murray and Wiercinski point out, a strictly user-centered development might restrict a project's ability to make full use of nascent technology. For them, the ideal interface would "provide the more familiar and comfortable features that facilitate the types of activities that scholars know," while affording new opportunities for discovery and experimentation "of which they are currently unaware" [7]. Until more research like Weigle's is conducted on users in music studies, we can only note that all development is an iterative process, an attempt to anticipate needs, get feedback, address shortcomings, and get more feedback. In the meantime, having robust models that can be easily adapted for use by others is a positive step toward increasing access to archival materials.

Conclusions

While the raw data of much notated music may be ready to be downloaded for analysis, the high-level computing skills required to retrieve and analyze that data means that it remains out of reach to many. In order to make collections such as these more accessible, both the resources and the training for encoding, retrieval, analysis, and display of encoded music need to be made available to researchers. We would like our prototype to be a resource to scholars in music studies—an example of open data and code that will lessen the demand for technical expertise for both the researcher and the user, while demonstrating the functionality that can be added to a single site accessed through an ordinary web browser.

As music OCR technology continues to become more successful at first-pass recognition, we will want to be prepared to make repositories available to more than just the technologically savvy few. With encoded music, the difference between a mode of access that involves scrolling through a list of text files and one that features an interactive display of scores and sound, is analogous to the difference between retrieving library materials through an institution with open stacks and one with closed stacks. Refining interests and homing in on relevant and interesting material are often the result of seeing a book on a shelf, opening it up and thumbing through it—reading a few sentences, checking out the TOC, skipping to the index, looking at the color plates in the middle. We don't always need or want to engage with materials in this manner, but having the option to do so is invaluable.

Works cited

[1] Padilla, Thomas. "On a Collections as Data Imperative". Conference Report. *Collections as Data: Stewardship and Use Models to Enhance Access*, Library of Congress, Washington, DC, September 27, 2016. http://digitalpreservation.gov/meetings/dcs16/tpadilla_OnaCollectionsasDataImperative_final.pdf

[2] Theimer, Kate. "The Future of Archives Is Participatory: Archives as Platform; or, A New Mission for Archives" presented at the Offene Archive 2.1 Conference, Stuttgart, Germany, April 3-4, 2014.

[3] Sayers, Jentery. "Minimal Definitions". 2016. https://go-dh.github.io/mincomp/thoughts/2016/10/02/minimal-definitions/

[4] Gil, Alex. "The User, the Learner, and the Machines We Make". 2015. https://go-dh.github.io/mincomp/thoughts/2015/05/21/user-vs-learner/

[5] Weigl, David, et al. "On Providing Semantic Alignment and Unified Access to Music Library Metadata" International Journal on Digital Libraries 20, no. 2 (2019), 25-47.

[6] Warwick, Claire, et al. "The Master Builders: LAIRAH Research on Good Practice in the Construction of Digital Humanities Projects" Literary & Linguistic Computing 23, no. 1 (2008), 383-96.

[7] Murray, Annie, and Jared Wiercinski. "A Design Methodology for Web-Based Sound Archives" Digital Humanities Quarterly 8, no. 2 (2014). https://www.digitalhumanities.org/dhqdev/vol/8/2/000173/000173.html

Rehearsal Encodings with a Social Life

David M. Weigl and Werner Goebl, Dept. of Music Acoustics—Wiener Klangstil,
University of Music and Performing Arts Vienna, Austria
weigl@mdw.ac.at, goebl@mdw.ac.at

Introduction

MEI-encoded scores are versatile music information resources representing musical meaning within a finely addressable XML structure. The Verovio MEI engraver reflects the hierarchy and identifiers of these encodings into its generated SVG output, supporting presentation of digital scores as richly interactive Web applications [1].

Typical MEI workflows initially involve scholarly or editorial activities to generate an encoding, followed by its subsequent publication and use. Further iterations may derive new encodings from precedents; but the suitability of MEI to interactive applications also offers more dynamic alternatives, in which the encoding provides a framework connecting data that is generated and consumed simultaneously in real-time. Exemplars include compositions which self-modify according to external contextual parameters, such as the current weather at time of performance [2], or which are assembled by user-imposed external semantics, such as a performer's explicit choices and implicit performative success at playing musical triggers within a composition [3].

When captured, these external semantic signals (interlinked with the MEI structure) themselves encode the evolution of a dynamic score during a particular performance. They have value beyond the immediate performance context; when archived, they allow audiences to revisit and compare different performances [4].

Reviewing rehearsal renditions

This capacity for capturing dynamic interactions with musical score supports reflection and introspection on the music rehearsal process. To demonstrate, we have built a Companion for Long-term Analyses of Rehearsal Attempts (CLARA),[26] a web application allowing users to track performances as real-time MIDI streams. These are aligned with MEI encodings [5, 6] associating temporal positions along the performance timeline with corresponding note identifiers in the MEI. Repeats and expansions introduce some additional complexity, as the alignment process requires the score to be fully expanded. We have modified Verovio for this purpose to facilitate dynamic rendering of different expansions encoded within the MEI.[27]

Close alignment of performance timeline and score allows musicians to revisit and review their rehearsal renditions, simultaneously navigating a score and a corresponding MIDI stream. Notes highlight corresponding to the current playback position; clicking on a note seeks playback to the corresponding instant; and, changing playback position flips to and highlights the appropriate place in the score.

This alignment of timeline and score further allows particular performance features (e.g. tempo curves) to be visualised, providing immediate feedback regarding corresponding stylistic and technical aspects of the musician's rehearsal rendition (Figure 1). CLARA feature visualisations, like Verovio engravings, are generated as semantically structured SVGs, supporting in-browser interactions such as highlighting visualisation regions during playback, and clicking on regions to seek to the appropriate playback position. Beyond review of a single rendition, this enables systematic comparison of multiple rehearsals, e.g. by clicking on different tempo curves to listen in to their corresponding rehearsal recordings at the appropriate playback position.

26 Code and demo available at https://iwk.mdw.ac.at/trompa-clara

27 Code changes incorporated into the main "develop" branch of the Verovio GitHub repository at https://github.com/rism-ch/verovio/ at time of writing (February 2020).

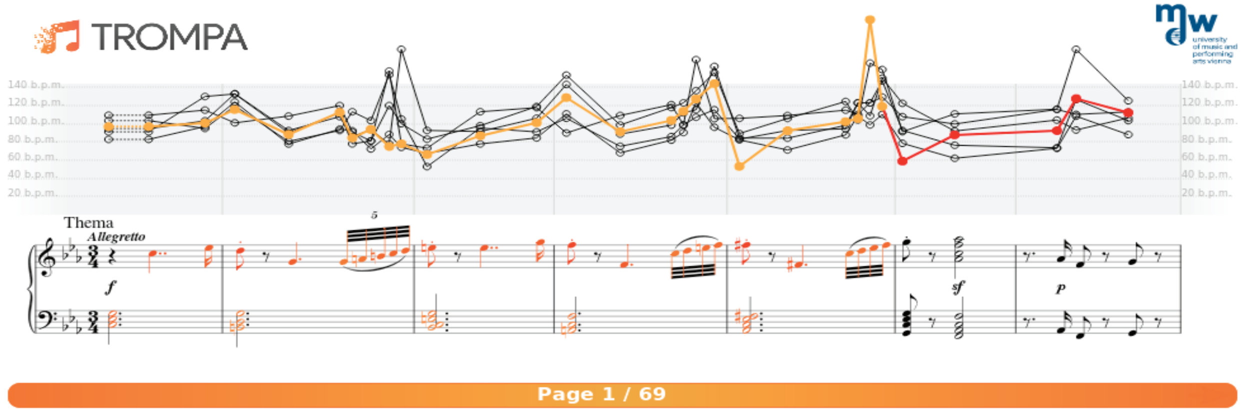


Figure 16: CLARA interface visualising tempo curves for six renditions of Beethoven's 32 Variations in c minor (WoO 80). Coloured tempo curve corresponds to currently selected rendition; colouration of tempo curve and notes indicates current playback position.

The rehearsal companion as a social machine

CLARA is a powerful tool for review of rehearsal progress, allowing renditions to be captured, gathered, and compared with fine granularity, thus providing insights on the evolution of performative aspects of one's rehearsals over time.

Beyond this, CLARA supports comparison of different performers' renditions. CLARA is implemented as a MELD (Music Encoding and Linked Data) [7] application; all alignment information is expressed as RDF triples,[28] identifying each timeline instant with a URI and interconnecting instants with the MEI structure through fragment URIs. Timelines are gathered for comparison according to their URI's inclusion within a Linked Data Platform (LDP)[29] container, itself a simple RDF structure. A selected rendition can be shared by simply importing its URI into the appropriate LDP container; the same rendition can be included in many containers (potentially owned by different users), and one user may manage a number of different containers, each potentially including different users' renditions. CLARA also supports the creation of Web Annotations[30] targeting specified score regions and corresponding timeline intervals of selected renditions. These annotations are themselves RDF structures with their own URIs, meaning they too can be shared between different users.

Through these mechanisms, we foresee performers tracking their own rehearsal progress; comparing their playing with selected peers; communicating with their teachers, through annotations and by comparison to reference renditions; and, incorporating notable pianists' renditions into their comparisons, allowing a pianist user to attempt to emulate, say, the tempo curve of Claudio Arrau's performance in their own renditions of Beethoven's Appassionata.

This work is being pursued as part of the TROMPA[31] project—Towards Richer Online Music Public-domain Archives [8]. TROMPA is building an infrastructure interconnecting publicly licensed music resources on the Web, adhering to FAIR principles [9] of making data Findable, Accessible, Interoperable, and Reusable. This infrastructure will support musicians in locating or generating MEI encodings of the scores they wish to rehearse, and coordinate the recording, alignment, and storage of rehearsals and annotations, allowing users to control the accessibility (public/private) of individual contributions, as well as incorporating others' (publicly licensed) contributions into their own views. Beyond instrumental players, this data, expressed in interoperable fashion using web standards, becomes available for reuse by others—providing scholars with empirical data on performance practice (e.g., to determine a typical tempo profile of the Appassionata as rehearsed in the "wild"), or music enthusiasts with a landscape of renditions to listen into and explore.

Together, we envision these technologies and their user base to function as a social machine [10] generating an interconnected Web of music information in which "the people do the creative work and the machine

28 https://www.w3.org/TR/rdf11-primer/

29 http://www.w3.org/TR/ldp/

30 https://www.w3.org/TR/annotation-model/

31 https://trompamusic.eu

does the administration" [11, p. 172] — and, in our case, the music information retrieval. We are faced, however, with a cold-start problem; in order to be attractive to new users, we require MEI encodings to rehearse, and users' rehearsal renditions to seed comparisons. Within TROMPA we are addressing this issue through crowd-sourcing techniques and by recruiting participants at partner institutions.[32] We will require coordination with the wider community of music encoding and music information researchers and practitioners in order to fully achieve our vision of a shared, dynamic, and richly interactive repertoire of publicly licensed scores and performance recordings.

Acknowledgements

The TROMPA project has received funding from the European Union's Horizon 2020 research and innovation programme *H2020-EU.3.6.3.1. - Study European heritage, memory, identity, integration and cultural interaction and translation, including its representations in cultural and scientific collections, archives and museums, to better inform and understand the present by richer interpretations of the past* under grant agreement No 770376.

Works cited

[1] [1] Pugin, Laurent. "Interaction Perspectives for Music Notation Applications" in *Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music* (SAAM 2013), published in the Association for Computing Machinery Digital Library, 54–58.

[2] Arkfeld, Joseph, and Raffaele Viglianti. "'Fortitude flanked with melody': Experiments in Music Composition and Performance with Digital Scores" in *Digital Humanities 2018 Book of Abstracts*, 2018, 315–17.

[3] Kallionpää, Maria, Chris Greenhalgh, Adrian Hazzard, David M. Weigl, Kevin R. Page, and Steven Benford. "Composing and realising a game-like performance for Disklavier and electronics" in *New Interfaces for Musical Expression*, 2017, 464–69.

[4] Benford, Steven, et al. "Designing the audience journey through repeated experiences" in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, published in the Association for Computing Machinery Digital Library, 1–12.

[5] Cancino-Chacón, Carlos, et al. "The ACCompanion v0.1: An Expressive Accompaniment System" in *Late Breaking/Demo Session, 18th International Society for Music Information Retrievall,* https://ismir2017.ismir.net/lbds/Cancino-Chacon2017.pdf

[6] Nakamura, Eita, Kazuyoshi Yoshii, and Haruhiro Katayose. *Performance Error Detection and Post-Processing for Fast and Accurate Symbolic Music Alignment* in *Proceedings of the 18th International Society for Music Information Retrieval Conference* (ISMIR 2017), 347–53.

[7] Weigl, David M., and Kevin R. Page. "A framework for distributed semantic annotation of musical score: 'Take it to the bridge!'" in *Proceedings of the 18th International Society for Music Information Retrieval Conference* (ISMIR 2017), 221–28.

[8] Weigl, David M., et al. "Interweaving and enriching digital music collections for scholarship, performance, and enjoyment" in *6th International Conference on Digital Libraries for Musicology* (DLfM 2019), published in the Association for Computing Machinery Digital Library, 84–88.

[9] Wilkinson, Mark D., et al. "The FAIR Guiding Principles for scientific data management and stewardship". Scientific data 3 (160018 (2016)), doi:10.1038/sdata.2016.18.

[10] Hendler, Jim, and Tim Berners-Lee. "From the Semantic Web to social machines: A research challenge for AI on the World Wide Web" Artificial intelligence 174, no. 2 (2009), 156–61, doi:10.1016/j.artint.2009.11.010.

[11] Berners-Lee, Tim, and Mark Fischetti. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*, New York: Harper Collins.

32 MEI generated by TROMPA activities available at https://github.com/trompamusic-encodings

MIDI 2.0: Promises and Challenges

Paul D. Lehrman
Tufts University
paul.lehrman@tufts.edu

Abstract

MIDI, the musical instrument digital interface, is a highly successful protocol for conveying and, through the use of Standard MIDI Files, representing musical performance information. However, it lacks the ability to convey notation information. The newly approved MIDI 2.0 protocol gives us a chance to rectify that by including notation information in the next version of the MIDI File Specification.

Introduction

Outside of standard Western notation itself, MIDI is the longest-serving and most ubiquitous method of representing musical performance. Its advantage over standard notation is its finer resolution in many dimensions. Its disadvantage is that it is not readable and interpretable in real time by human performers. The recently adopted MIDI 2.0 specification improves its resolution by orders of magnitude, and that it is still a work in progress means we have a potential opportunity to align it with other encoding technologies so that it can be used to represent music in human-readable form.

MIDI and standard MIDI files

The original MIDI 1.0 Specification, which was adopted in 1982 [1], was designed to enable electronic instruments from different manufacturers to communicate with each other digitally. Computer programmers were quick to realize that the data stream created by a MIDI instrument could be digitally recorded, and multiple data streams could be combined in a software program similarly to a multitrack tape recorder, allowing the creation of computer-controlled digital orchestras [2]. These programs, called sequencers, stored the MIDI stream in proprietary file formats, but by 1988, an addition to the MIDI specification created the Standard MIDI File (SMF), an open-source format for storing MIDI sequences [3]. Almost all makers of MIDI software, including makers of notation-based programs, adopted SMF as an alternative means of storage, thereby allowing users to bring sequences across multiple platforms, with minimal loss of performance information.

But SMFs do not carry much information specific to notation. While the MIDI Specification itself has expanded greatly since its initial adoption, it is still very much oriented to performance. Most musical gestures are recordable and reproducible in a SMF, but notation elements are limited to time signatures, tempos, key signatures, and lyrics. Beams, stems, ties, clefs, bowings, articulations, expression marks, repeats, and many other features of standard notation are not part of the SMF specification, and thus the conversion of a notation file into SMF, although a feature of many popular notation programs, results in a significant loss of information that cannot be recovered.

Advantages of MIDI

On the other hand, MIDI has several distinct advantages over standard notation. For one thing, it is exquisitely precise. The timing or length of a note in a Standard MIDI File can be resolved to as little as 1/3000th of a second, or 0.33 milliseconds, which is the equivalent of a triplet 1/2048th note at MM=120. The dynamic level of the onset of a note, called “velocity” in MIDI, can be specified to be any of 127 discrete values. Expressive information, including volume changes, portamento, vibrato depth and speed, and timbral changes, can also be resolved to 127 values, with the same timing resolution of 0.33 ms. (Pitch bend resolution is even higher, with 16,383 values.) Over 120 different expressive parameters can be controlled on each instrument in a MIDI orchestra using "continuous controllers" and other commands.

Unlike performances of printed music, a MIDI performance from a computer sequencer will always come out exactly the same if the performer or programmer wishes it to—but although a MIDI file cannot be "read" and interpreted by a musician the way a printed score can, it can be manipulated offline or in real time in terms of tempo, instrumental balance, orchestration, mode, or many other aspects of performance.

MIDI 2.0

From its beginning nearly 40 years ago until this year, the MIDI Specification has been labelled "1.0". Although there have been many additions to the Specification, MIDI instruments introduced at the beginning of the MIDI era are still 100% compatible with instruments and programs being developed today—that is, although such early instruments will not recognize (and in fact will specifically ignore) commands that were added to the Specification subsequent to their introduction, their original capabilities remain completely viable.

Earlier this year, however, after several years of negotiation among the industry groups responsible for supervising the MIDI Specification in North America, Europe, and Asia, a new set of protocols known as MIDI 2.0 was adopted. While care has been taken to preserve compatibility with MIDI 1.0 devices, the 2.0 Specification greatly expands MIDI's capabilities for a new generation of hardware and software [4].

Resolution

Primary among MIDI 2.0's features are a greatly expanded feature set and greatly expanded resolution of musical parameters. When MIDI 1.0 was introduced, 8-bit data paths and computer clock speeds of 1 MegaHertz or less were standard. Today 32- and 64-bit data paths are the rule, and clock speeds are several orders of magnitude faster in the multi-GigaHertz range. MIDI 2.0 takes advantage of these greater bandwidths by expanding the resolution of commands from 8 bits (actually 7, since the first bit is used to determine whether a byte is a command or a data point), to 16. This allows, for example, the possible value of a note's velocity byte to be expanded from 127 points to over 65,000.

Continuous controllers

Another important feature involves the implementation of continuous controllers. In MIDI 1.0, controllers are "per-channel," e.g., if an instrument is using a single MIDI channel to produce the sound of a brass ensemble, introducing vibrato or pitch bend affects all of the notes on the channel identically. MIDI 2.0 has the ability to apply controller or pitchbend information to each note individually. Rather than 127 controllers per channel, there are now 512 available controllers *per note*. The resolution of all of these controllers is now 32 bits: that's over 4 billion separate values. The controller set is expandable and customizable, with the potential to have over 32,000 discrete controllers.

Note messages

The note messages themselves in MIDI 2.0 carry a lot more information. A note can have an "attribute" assigned to it, which can communicate articulation, like a string sforzando or pizzicato; position of a hit on a drum or cymbal; or pitch information totally independently of the note number, making it easy to construct non-tempered or real-time variable scales. Since pitch information and note number are now separate parameters, multiple notes with the same note number but with different attributes can be transmitted and understood.

Channels

MIDI 1.0 limited the number of MIDI channels addressable over a single cable to 16. This was in large measure because at the original data rate of 3,125 bytes per second, attempting to control more instruments than that would likely have resulted in delays or dropped commands. MIDI 2.0 does not use the extremely slow—by today's standards—MIDI cable defined in the MIDI 1.0 Specification, but instead is "transport independent," meaning it will potentially be able to use any common connection protocol. The first transport for the new protocol will be USB, but it is expected in the near future that other mechanisms including Thunderbolt, WiFi, and Bluetooth will be adopted. Freed from this speed restriction, MIDI 2.0 offers 16 "groups", each of which has 16 channels, for a total of 256 channels per "cable." And unlike MIDI 1.0, which has separate "In" and "Out" connections on each device, MIDI 2.0 is bidirectional.

Hardware communication

The other improvements in MIDI 2.0 are primarily on the hardware side. It introduces new technologies called "Property Exchange" and "Profiles," designed to take advantage of this two-way communication. They are part of a new set of commands called MIDI Capability Inquiry, or MIDI-CI. Devices will include MIDI-CI "profiles" built into their operating systems. If two connected devices use MIDI-CI, they will be able to exchange important information about each other: their profiles will announce whether each device supports per-note pitchbend and controllers, how many channels or streams it responds to, how it handles controller commands, and what kind of instrument or device it is: a synthesizer, a silent keyboard, a sequencer, an arpeggiator, a rhythm computer, a mixer, an effects device, a lighting board, a video switcher, or even a drone.

For example, in the world of electronic organs, although many instruments have the standard nine drawbars, different manufacturers map different MIDI continuous controllers to the drawbars; but if two instruments subscribed to an agreed-upon "Drawbar Organ" profile, files would have identical drawbar settings when transferred from one instrument to the other.

Standard MIDI files 2.0

What remains to be written into the MIDI 2.0 Specification is how Standard MIDI Files will be updated to handle the new commands and resolutions. The Technical Standards Board of the MIDI Manufacturers Association—the volunteer industry group that oversees the Specification—is in the initial stages of developing a specification provisionally known as "SM2F."

In addition to implementing the new features of MIDI 2.0, this early stage of SM2F development offers an opportunity to integrate information not strictly related to performance, and that includes notation data. Given the large bandwidth and open structure of MIDI 2.0, there is plenty of room for the exchange of notation data in all of its forms in both real time and as part of a file. While it is much too early to even speculate whether SM2F will address notation issues, it is worth noting that one member of the group working on SM2F is Michael Good, the inventor of MusicXML, the expansive and expandable music notation file format that is the equivalent of SMF (1.0) in the area of music notation software [5]. Good represents the intersection of the MIDI community with the notation community, two bodies that previously have had little in common.

Conclusion

MIDI 2.0 is a major update to a highly successful technology that brings digital music-making up to date and opens up new means of expression and precision. The new Standard MIDI File 2.0 specification, which is to follow, represents an opportunity to include many musical features not available in the current Standard MIDI Files. Perhaps the ability to transfer both performance and notation information between applications and platforms could be among them.

Acknowledgements

Special thanks to Rick Cohen, chairman of the MIDI Manufacturers Association Technical Standards Board, and former chair of the MIDI 2.0 Protocol Working Group; and Michael Good, Vice-president of MusicXML Technology at MakeMusic, Inc.

Works cited

[1] The Complete MIDI 1.0 Detailed Specification. Version 96.1. MIDI Manufacturers Association, 1996.

[2] Lehrman, Paul, and Tim Tully. MIDI For the Professional. Second edition. New York: Music Sales Corporation, 1992.

[3] Standard MIDI Files, (Recommended Practice 001). MIDI Manufacturers Association, 1996.

[4] MIDI 2.0 Specification Overview. Version 1.0. Association of Musical Electronics Industry/MIDI Manufacturers Association, 2020.

[5] musicXML FAQ, https://www.musicxml.com/tutorial/faq/ (retrieved 12 May 2020).

MusicDiff – A Diff Tool for MEI

Kristin Herold
Beethovens Werkstatt
herold@beethovens-werkstatt.de

Dr. Johannes Kepper
Beethovens Werkstatt
kepper@edirom.de

Ran Mo
Beethovens Werkstatt
mo@beethovens-werkstatt.de

Agnes Seipelt
Beethovens Werkstatt
seipelt@beethovens-werkstatt.de

Introduction

For musicologists, the collation of multiple sources of the same work is a frequent task. By comparing different witnesses, they seek to identify variation, describe dependencies, and ultimately understand the genesis and transmission of (musical) works. Obviously, the need for such comparison is independent from the medium in which a musical work is manifested.

In computing, comparing files for difference is a common task, and the well-known Unix utility *diff* is almost 46 years old [1]. However, *diff*, like many other such tools, operates on plain text. While many music encoding formats based on plain text exist, formats used in the field of Digital Humanities are typically based on XML. There are dedicated algorithms for comparing XML as well,[1] but they only focus on the syntax of XML, but not the semantic structures modelled into such standards as MEI. MEI seeks to describe musical structures, and the XML syntax is just a means to express those structures. A diff tool for music should focus on comparing musical structures, but not the specifics of their serialization into a file format.

In *Beethovens Werkstatt*, a 16-year project focussed on exploring the concepts and requirements of digital genetic editions of music, based on and arguing with examples from Ludwig van Beethoven, a case-bound diff tool for music was developed. The following paper discusses how that specific tool can be generalized, and which use cases such a tool may support.

VideApp$_{Arr}$

Beethovens Werkstatt seeks to explore compositional processes from different perspectives. In its recently completed second module, the project dealt with a number of Beethoven's works that the composer re-arranged for other performing forces. For these works, printed editions of both the original works and their respective arrangements were fully encoded in MEI, following a rather plain style, i.e. no typographical or genetical details about the sources were preserved. Instead, an additional file per comparison with merely more than pointers to both source encodings was provided. With this data model, it is possible to automatically align both files and present them from multiple perspectives with an application called VideApp$_{Arr}$ – the component dealing with arrangements within the (modular) VideApp.[2]

1 https://pypi.org/project/xmldiff/, http://diffxml.sourceforge.net/, https://www.oxygenxml.com/files_compare_img.html
2 https://videapp-arr.beethovens-werkstatt.de

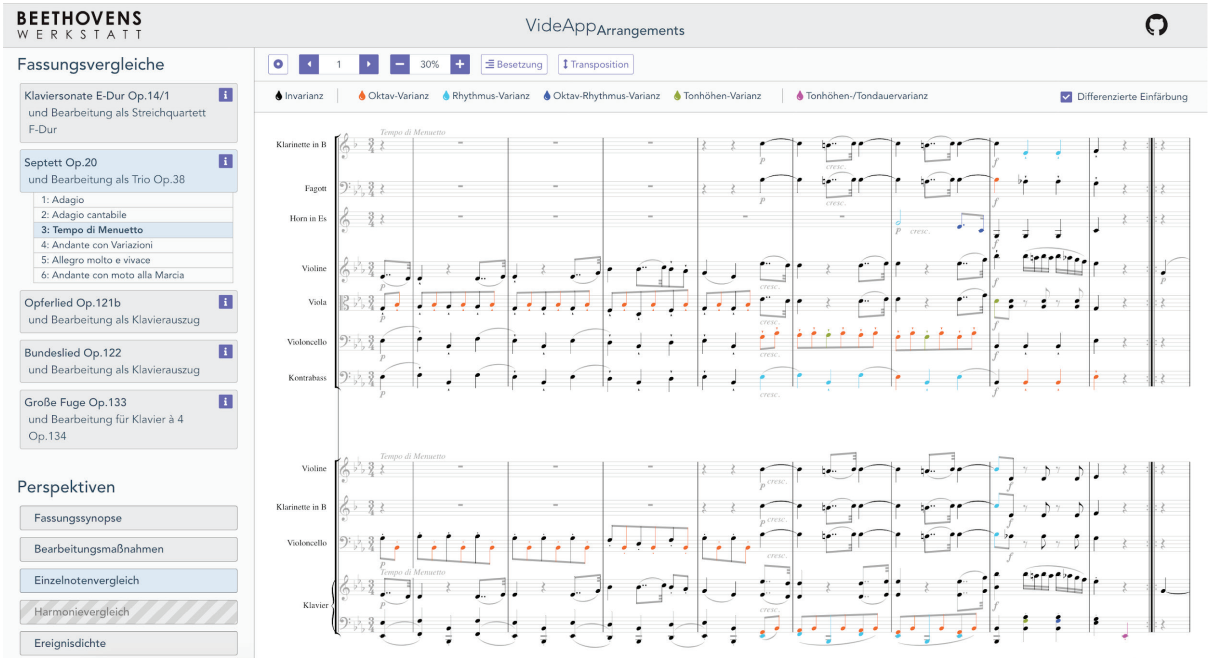


Figure 1: VideApp$_{Arr}$ showing the "Single Note Comparison" of Beethoven's op. 20 (top) and op. 38 (bottom).

Most of these perspectives are based on the comparison of individual notes in three different "dimensions": metrical position, pitch, and rhythm. Metrical position means that only notes sounding simultaneously will be compared. For pitch, octave and pitch class are evaluated independently, while rhythm is taken into account directly. Variation of these three parameters is organized into different combinations, such as notes in a different octave, notes with different duration or other types of variation, but also notes which have an exact match. No attention is paid to beams and similar features, as they are mostly visual artifacts, which typically do not affect the musical structure. By intention, accidental aspects of the score such as dynamic markings are not taken into account for comparison either, as their high incidence may easily conceal the more significant substantial differences. Voice leading is also ignored by this comparison, as it would be misleading in the context of a comparison of rearranged works. Especially in a piano reduction, "voices" from multiple instruments are condensed in a way that frequently fails to show the same "melodic lines" for middle voices and others, so that the aspect of preceding and / or succeeding notes can hardly be made a default criterion for comparing two arrangements.

Generalising VideApp$_{Arr}$ to MusicDiff

The data model underlying VideApp$_{Arr}$ is a rather strict version of MEI, disallowing variation, editorial intervention, and other more complex concepts of MEI. While it took significant effort to ensure correctness of the encodings used in *Beethovens Werkstatt*, the generation of these encodings was straightforward in principle, as they were just transcribed from the original prints using scorewriting applications, and then transformed into MEI via MusicXML conversion. This workflow is all but unique, and we anticipate that numerous other projects create MEI files with about the same information value, though perhaps expressed in slightly different models of MEI.

In the process of proofreading the files relevant for the second module of the project, it became apparent that the VideApp$_{Arr}$ is actually very supportive in this task, as it consequently highlights differences of all kinds, even when some of which are not visible in a rendered score. This is particularly true for the correct encoding of *gestural* information in MEI, which in this context means sounding pitch affected by the general key signature at the beginning of the piece, but not local accidentals.

This observation led to the idea of broadening the scope of this tool beyond the original context of *Beethovens Werkstatt*, and to modify it so that users can actually upload and *diff* their own MEI files. While several of the perspectives offered by the VideApp$_{Arr}$ may be useful for this purpose, we intentionally focussed on the most simple diff view to begin with. This view has been condensed into a separate web application called MusicDiff.[3] The following examples illustrate the use of this app for musicological purposes beyond the original scope of *Beethovens Werkstatt*.

Example use cases

In opera, the music was usually adjusted to local requirements, settings, and expectations. Pieces from different works were frequently integrated (in)to performances, which led to the need to create smooth transitions between those pieces. The research project “Pasticcio. Ways of arranging attractive Operas”[4] explores such pasticcii. This includes the recitative “Ah Per te solo” from the pasticcio “Catone” by G. F. Handel, which was first performed in London in 1732. In Handel’s manuscript, two versions of this recitative are transmitted, one ending on G# major, leading to the aria “Care faci del ben mio” (E major), and one leading to the replacement aria “Sento in riva all’altre sponde” (A major). Obviously, the different key of the substituted succeeding aria required some adjustments to the music.

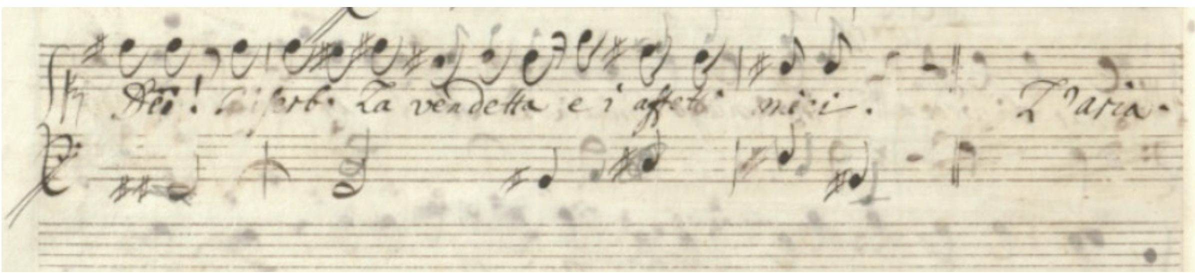


Figure 2: Recitative “Ah Per te solo” from the Pasticcio “Catone” by G. F. Handel. Staats- und Universitätsbibliothek Hamburg Carl von Ossietzky, D-Hs M A/1012, p. 187.[5]

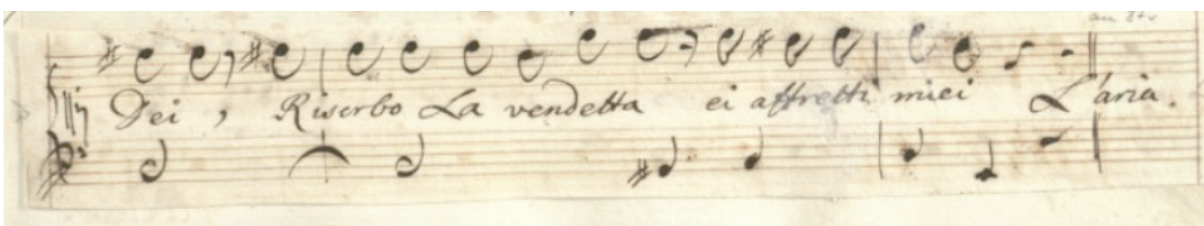


Figure 3: Substituted recitative “Ah Per te solo” from the Pasticcio “Catone” by G. F. Handel. Staats- und Universitätsbibliothek Hamburg Carl von Ossietzky, D-Hs M A/1012, p. 184.[6]

Obviously, it is possible to manually compare the score images, and there are also tools supporting such an approach at least with musical prints,[7] this approach doesn’t scale well and may take significant time when comparing works larger than these four measures. However, when looking at the rendition provided by MusicDiff, the difference between both versions becomes immediately imminent:

3 Available from https://music-diff.edirom.de

4 https://www.pasticcio-project.eu/

5 https://digitalisate.sub.uni-hamburg.de/de/nc/detail.html?id=1901&tx_dlf%5Bid%5D=22734&tx_dlf%5Bpage%5D=187

6 https://digitalisate.sub.uni-hamburg.de/de/nc/detail.html?id=1901&tx_dlf%5Bid%5D=22734&tx_dlf%5Bpage%5D=184

7 https://ehinman.edirom.de/

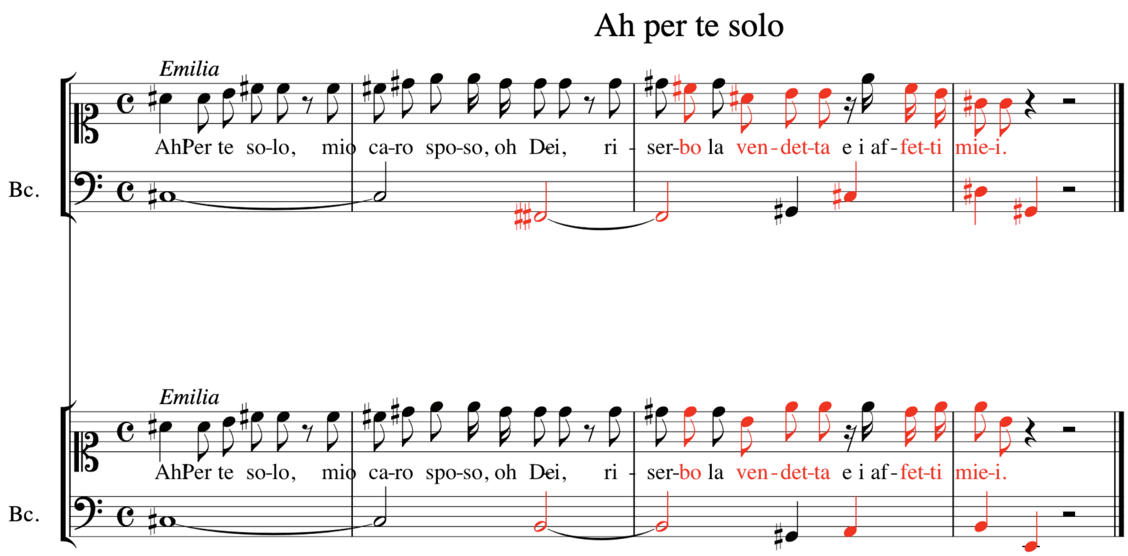



Figure 4: Comparing encodings from both versions of "Ah Per te solo" with MusicDiff.

A second example helps to illustrate the flexibility of MusicDiff, and why a regular diff tool would necessarily fail to recognize musical differences at the level of abstraction considered here. This example deals with two independent encodings of Grieg's "Erotikon" op. 43, Nr. 5. One of these encodings has been made available as Humdrum file by KernScores,[8] while the other comes as MusicXML file, derived from an original Capella transcription.[9] Even though the origins of both encodings do not suggest this interpretation, one may wonder if these files share a history, i.e. if one has been converted from the other, or both have been derived from an (unknown) original encoding. If that would be the case, their content would probably be almost identical, with differences being caused by transformation loss (and thus highly systematic differences). In order to answer these questions, both encodings have been transformed to MEI, and processed by MusicDiff.

8 https://kern.humdrum.org/cgi-bin/ksdata?location=users/craig/classical/grieg/op43&file=erotic-poem.krn&format=info

9 http://www.hausmusik.ch/notenregal/g/grieg/klavierstuecke/lyrische_stuecke/erotik_edvard_grieg_/

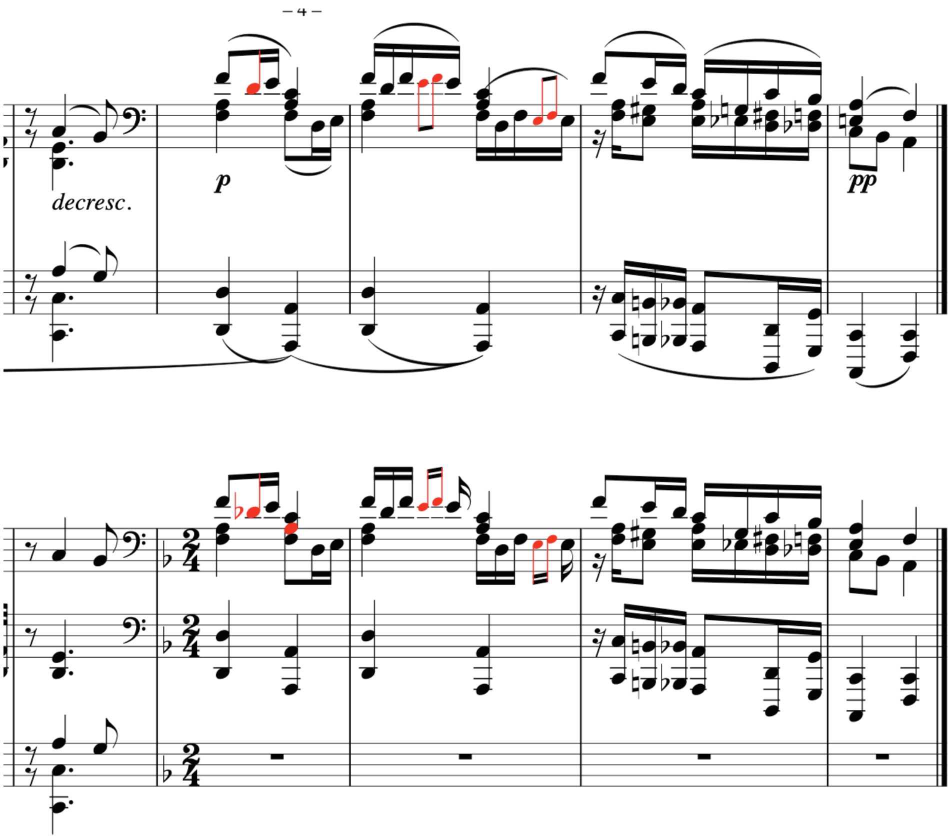



Figure 5: The last 5 measures of the lyrical piano piece “Erotikon” op. 43, Nr. 5 of Grieg (top: Humdrum file converted to MEI, bottom: MusicXML file converted to MEI.

Apparently, there is a small level of variation between both encodings, with only a small number of regular and grace notes being highlighted by MusicDiff. This seems to indicate that the original encodings have been generated independent of each other. However, this example perfectly illustrates how MusicDiff is able to overlook structural differences: The fact that the second version of Grieg’s piece is laid out on three staves does not affect the comparison. In the same way, MusicDiff is able to go over music written in chords vs. music written in voices, or, more generic, layers. Admittedly, other interpretations of what qualifies as variation are possible and equally valid.[10]

Keeping an overview

While the collation provided by MusicDiff offers a very striking emphasis of the variation in the current viewport, this perspective does not provide a wider overview of the work in total – the user has to flip through all pages to get an impression of the distribution of differences between the compared encodings. In order to facilitate getting such an impression, *Beethovens Werkstatt* has integrated the concept of Sunburst diagrams[11] into VideApp$_{Arr}$, and this feature has been carried over to MusicDiff as well. Sunburst diagrams visualize hierarchical data by concentric circles. On the outer ring, all measures of a piece are given, while the second ring denotes musical sections, and the inner ring reflects movements. The user may click on any measure, and the page holding this measure will be displayed. However, as seen in Figure 6, the measures may be used to provide additional information as well.

10 It would be certainly possible to include those different interpretations into the stylesheets underlying MusicDiff and let the user pick the “strictness” of the comparison according to her specific needs, but this would require significant work clearly out of scope for Beethovens Werkstatt.

11 https://en.wikipedia.org/wiki/Sunburst_chart

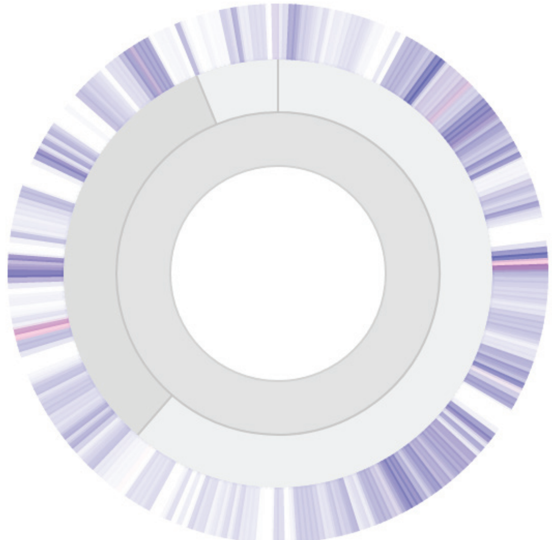


Figure 6: A Sunburst diagram for the comparison of Beethoven's op. 20 and the rearrangement into op. 38 based on it. White color indicates identity between both versions, blue indicates variance, and red indicates difference.

In this example from *Beethovens Werkstatt*, measures are colored depending on the comparison results. First, the saturation of a measure indicates the level of identity between original version and rearrangement – a measure displayed in white is unchanged, while a colorful measure has a high degree of variation. In *Beethovens Werkstatt*, however, a distinction is made between variant notes (which still share the pitch class or duration with their counterpart) and different notes (which have no counterpart at their respective metrical position at all). While variant notes typically indicate local adjustments of some sort, differences in this sense indicate major compositional processes. While the first are indicated by blue color, the latter make use of red tones. Both colors may blend according to the ratio of their respective notes within each measure. With this mechanism, it becomes possible to get a very quick overview over the distribution of variation across all 288 measures of this example, and to navigate within the piece for closer inspection very easily.

Technical setup, limitations, and potentials

As mentioned earlier, the MusicDiff app is a stripped-down version of VideApp$_{Arr}$. It allows the user to upload her own MEI encodings. At this point, no validation happens while processing the data – it is upon the user to ensure that the input conforms to the schema[12] expected by the tool.

MusicDiff itself does not come with a backend. Instead, it utilizes a varied toolbox[13] for converting between different music encoding formats, manipulating MEI instances, and other related workflows. It is based on TEI's OxGarage and actually uses the same backend, gently adjusted to music needs. The user's uploaded files are wrapped in a new file, and sent to MEIGarage, which runs a fairly complex series of XSLT transformations[14] on the files, enriching them with various information needed to perform the actual comparison. The output is ultimately sent back to the user of MusicDiff and displayed there. This setup allows MusicDiff to be a rather lightweight application, which could be integrated into other tools quite easily.

MusicDiff relies on the MEI profile developed for VideApp$_{Arr}$. This profile strictly requires a very simple use of MEI. While it wasn't available at the time when work on VideApp$_{Arr}$ was begun, the recent MEI Basic profile seeks to serve the same purpose: the definition of a strongly simplified and strictly controlled version of MEI which may serve as a common ground for interchange both within MEI (i.e., between projects relying on different richer flavors of MEI) and outside of MEI (i.e., to simplify conversion with other, less expressive formats). As we expect significant uptake of MEI Basic, it seems sensible to modify MusicDiff to operate on this profile

12 https://github.com/BeethovensWerkstatt/module2/blob/dev/data/odd/bw_module2_works.odd

13 https://meigarage.edirom.de

14 https://github.com/Edirom/data-configuration/blob/dev/scripts/compare.files.xsl

instead of the current one. As both have an almost identical coverage of MEI features, and merely differ in how they are expressed, this seems like a reasonable goal which will significantly help to improve the applicability of MusicDiff.

Some more interesting features are available in VideApp$_{Arr}$, which haven't been ported to MusicDiff yet. This includes the possibility to transpose the encodings to be compared to a common key, should they be written in different keys – the actual comparison is already capable of comparing encodings independent of the key they use, but it sometimes helps the user to bring everything to C Major / A minor for better legibility. Another feature already available in the underlying transformations is the possibility to omit one or more staves from the versions to be compared. That way, it becomes possible to answer questions like how the clarinet of version A relates with the clarinet of version B. Both of these features are fully functional in the underlying code, but don't have a user interface in MusicDiff yet. We hope to add support for both these features in the near future.

A significantly more challenging issue is a general limitation of the comparison scripts, which, at this point, require that both versions use the same number (and distribution) of measures, i.e. measures are compared according to their position in the piece. For the examples covered in the second module of *Beethovens Werkstatt*, this was a safe assumption to make, but obviously, this isn't generally true. However, it is fairly complex to recognize whether two measures differ because of some variation between them, or because an additional measure has been inserted in one of the compared versions. While this is clearly an interesting and challenging issue, we don't expect to support this use case anytime soon, but may instead ask the user to submit a concordance of measures.

Conclusion

MusicDiff is a compelling tool for various use cases, musicological and beyond. It allows comparison of two files with encoded music scores, and will clearly highlight the differences between these encodings. In larger scores, it directs the user to variant spots using a Sunburst diagram. That way, comparing two music encodings becomes significantly easier. This is especially true, because MusicDiff correctly handles differences between written and sounding pitches – it resolves transposition and correctly considers key signatures. It also puts aside visual structures and artifacts to some degree, and thus helps to focus on "real" differences. This requires MusicDiff to be a tool guided by certain concepts – it implements *Beethovens Werkstatt's* model of identity and variation, which may or may not apply equally well to other contexts. Being released under an open license, however, it can be adjusted to other concepts. Even as it stands, MusicDiff is the authoritative tool for a semantic comparison of encoded music scores.

Work Cited

[1] Hunt, James W., and M. Douglas McIlroy. "An Algorithm for Differential File Comparison". Computing Science Technical Report, Bell Laboratories (June 1976).

Beethovens Werkstatt on the Test Bench

Salome Obert
Department of Musicology Detmold/Paderborn
Oberts@campus.uni-paderborn.de

Abstract

In my master thesis I am working on the analysis of scriptural problems, trying to discuss the chronology of Ludwig van Beethoven's entries in the autograph of his *Flohlied* op. 75, no. 3. This is in order to examine the efficiency of the so-called *VideApp* of the research project *Beethovens Werkstatt* which studies sketches and manuscripts of Beethoven by combining methods of genetic criticism and digital edition.

Introduction

The research project *Beethovens Werkstatt* studies sketches and manuscripts of Ludwig van Beethoven by combining methods of genetic criticism and digital edition.[1] It is a joint project located at the Beethoven-Haus Bonn and the Department of Musicology Detmold/Paderborn, and is funded by the Academy of Sciences and Literature Mainz. It started in 2014. In the first of its five modules, which took until 2016, the project focussed on the description of Beethoven's revision processes in several manuscripts of different genres. Several tools to show various layers of the compositional process as well as a reconstruction of a piece's chronology have been developed in the project. Additionally, a terminological base was formed with a glossary, which is still constantly refined.

The project's so-called *VideApp*[2] is an example of an open access-web application which was developed during the first module. It combines a digital presentation of the composer's manuscript via MEI-data (representing the musical text) with SVG-shapes (marking the content of the document itself). Additionally, the *VideApp* gives a description of the sources, an overview of current research and a detailed analysis of textual genesis and of compositional chronology – not only verbally but also in a synoptic visualization of the source, its transcription and corresponding MEI-data. The different methods and forms of representation which were worked out in this way should be transferable both to other compositions and other composers.

In my master thesis I probe whether *Beethovens Werkstatt* can keep the promise of the *VideApp*'s transferability by studying Beethoven's *Flohlied* op. 75, no. 3. The song's autograph (D-BNba, NE 220) was presumed to be lost; for this reason, it could not be taken into account for the 'Beethoven Gesamtausgabe' in 1990. Only in 1998 the manuscript was bought from private hands and brought to the public with a facsimile edition by Helga Lühning, who explains that the manuscript is probably an autograph transcription which served as engraver's model [1, p. 37]. On different levels, Beethoven did his typical corrections like cancellations, overwriting, and he used different writing media and jump marks, all of which were described by Lühning in her edition.

Figure 1 gives an exemplary impression of the manuscript.

1 Beethovens Werkstatt (2020). Beethoven-Haus Bonn, Musikwissenschaftliches Seminar Detmold/Paderborn. https://www.beethovens-werkstatt.de

2 Recently the project decided to name all web applications which are developed during the modules VideApp. Different suffixes specify the modules focuses'. The suffix 'Var' refers to the first module's focus on Variants.

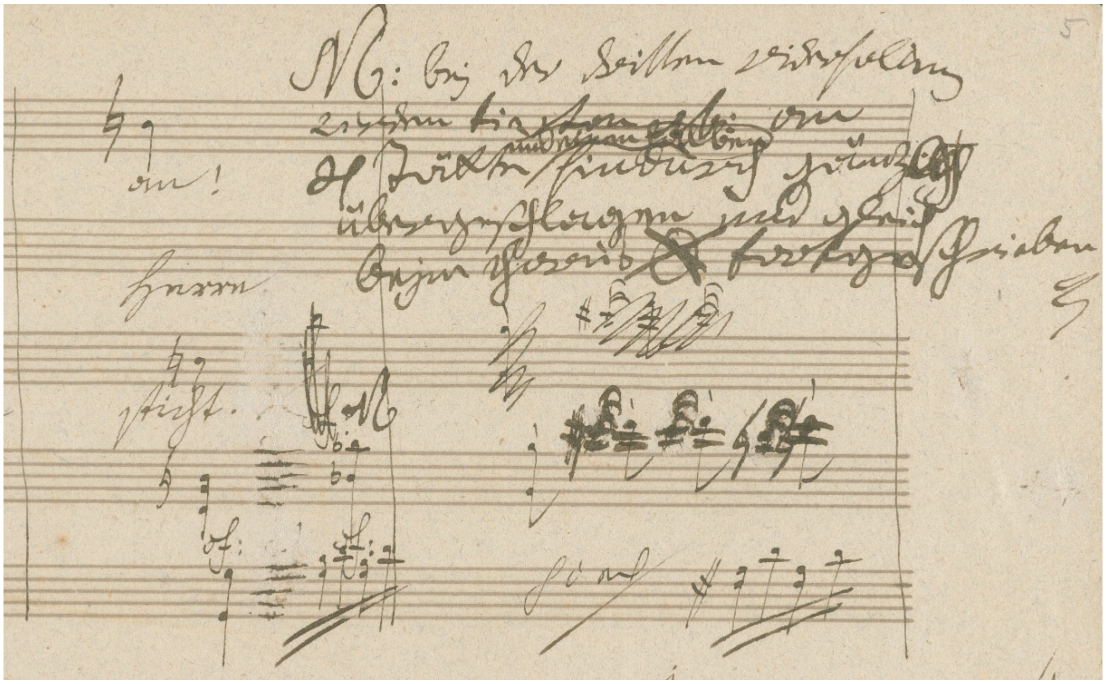


Figure 1: *Beethoven op. 75, no. 3 Flohlied.* Autograph D-BNba, NE 220, p. 5.

A scriptual analysis Beethoven's *Flohlied*

Guiding questions and methods

In my own approach I describe the source verbally and analyse some parts in which the writing flow was interrupted and modifications were added – *Beethovens Werkstatt* calls such areas 'Textnarben' (i.e. 'textual scars'). The analysis does not include technical examinations of the paper, but only a scriptural analysis trying to discuss the chronology of Beethoven's entries. For this purpose, I discuss the following questions:

- Are there musical reasons for explaining a modification, referring to harmonic, melodic, rhythmic or lyric aspects?
- Are there non-musical reasons which explain a modification, for example additional hints for a copyist?
- Into which categories could we classify Beethoven's entries?

In reconstructing the compositional chronology I use the same methods as the *VideApp*. Besides a verbal description of the piece's scriptural state, I will model its text in MEI and generate SVG-shapes by tracing the manuscript's digitization on a graphic tablet.

At the moment I am creating SVG-shapes by tracing each entry in the manuscript's digital images with the aid of a graphic tablet. From this working process I am presenting now two examples of textual scars in Beethoven's *Flohlied* which I examined already.

Two examples of textual scars

The first textual scar is located on the second page of the autograph. It belongs to the third strophe. In Kurrent script Beethoven wrote the word *ge=stochen* (which means *biten* [by a flea]). At the first syllable the letter *G* is written two times: once as a capital and once more as a small letter which can be seen in Figure 2.

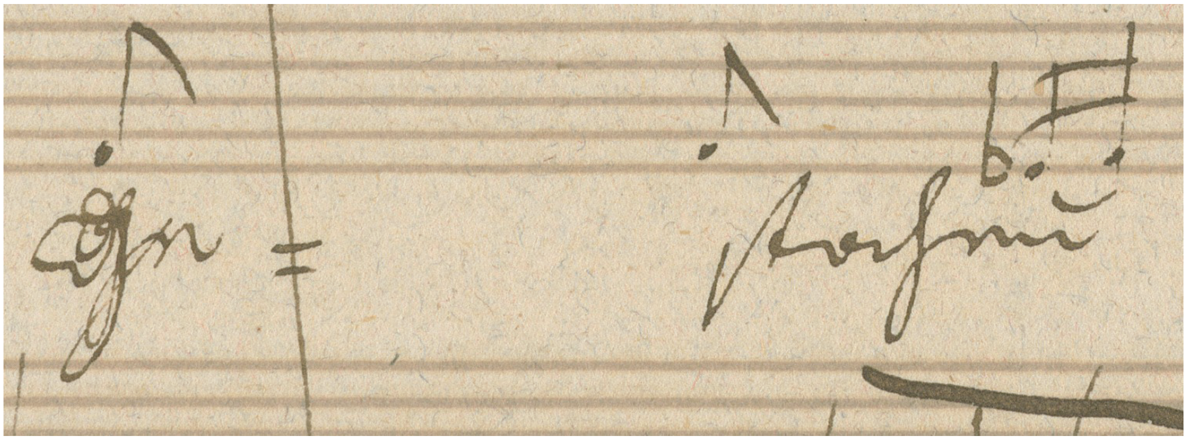


Figure 2: *Beethoven op. 75, no. 3 Flohlied*. Autograph D-BNba NE 220, p. 3.

A possible explanation could be that Beethoven started the word with a capital and stopped while realizing the word has to be written with a small letter. So, he corrected the capital into a small letter and continued writing the word. Another explanation is that Beethoven wrote the whole syllable starting with a small letter and changed it afterwards into a capital letter. In this way, Beethoven could have marked the begin of a new verse. The second hypothesis is more obvious because all new verses begin with a capital letter. Furthermore on the next page we find two more similar corrections. In general, Beethoven used the same orthography and punctuation as in the printed version of Goethe's poem which can be considered a reverence to the poet [1, p. 38].

The second textual scar is on the penultimate page in the piano part. In the bass voice Alberti basses are noted in four groups of semiquavers. According to the time signature of 2/4 two semiquaver groups are too much here. Beethoven cancelled the first and third group of semiquavers. To understand which kind of compositional problem Beethoven tried to solve here, it is necessary to have a look at the piano's upper part as well (see Figure 3).

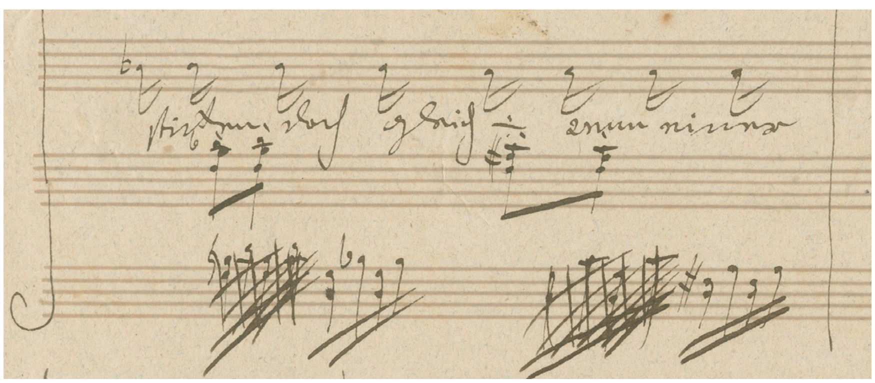



Figure 3: *Beethoven op. 75, no. 3 Flohlied*. Autograph D-BNba NE 220, p. 7, first measure.

The cancelled semiquavers are written directly below the right hand which is an indication that Beethoven cancelled the bass voice after having written both the upper and the bass voice. Probably he saw an error in the accompaniment, cancelled the semiquaver groups and set two new groups of semiquavers to correct the error. But what kind of error did Beethoven see? If the cancelled passage would sound simultaneously with the upper part, sharp dissonances would result not only within the piano part (e.g. b″ flat – b natural on the first beat) but also with the voice part (b′ flat). Therefore it is helpful to have a look at the previous and the following measure in which the harmonies which are identical with the cancelled part of the bar shown above really sound consonant (see Figure 4 and 5).

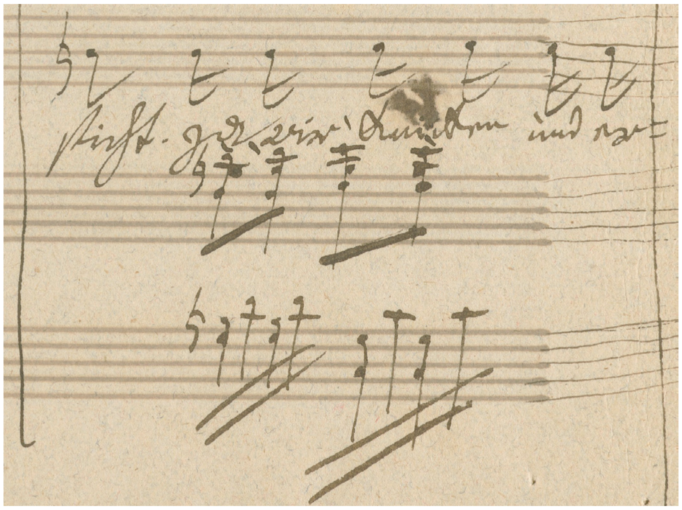



Figure 4: *Figure 4: Beethoven op. 75, no. 3 Flohlied*. Autograph D-BNba, NE 220, p. 6, last measure.

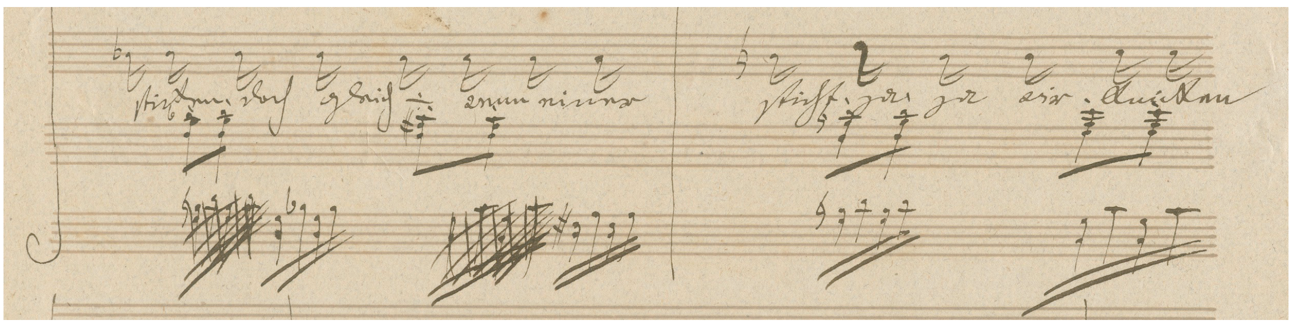


Figure 5: *Figure 5: Beethoven op. 75, no. 3 Flohlied*. Autograph D-BNba, NE 220, p. 7, first and second measure.

In starting to write on a new page (with the bar in Figure 3), probably Beethoven was wrong about the correct place in which the harmonical phrase was to be repeated – which was actually only in the following measure (see Figure 5). This is a typical mistake in a copying process – so judging from this mistake it is most likely that Beethoven copied this song from an already existing draft or manuscript [1, p. 38].

Conclusion

The results of my master thesis, presented with the aid of the *VideApp*, help to understand the compositional process of Beethoven's *Flohlied*, but also approve the *VideApp*'s transferability and verify generally that the *VideApp* can show those compositional processes adequately. At the same time my work hints at details where improvements should be made in order to facilitate a better understanding of the genetic processes. Besides the discussion of the *Flohlied* itself, I will identify limitations of the current software, especially with regard to re-using it with custom data, and propose simple revisions and additions, which may still make a big difference for an average user with less access to the original developers.

Work cited

[1] *Ludwig van Beethoven. Drei Lieder nach Gedichten von Goethe. Facsimile.* With a commentary by Helga Lühning. Bonn, 1999.

Figured Bass Encodings for Bach Chorales in Various Symbolic Formats: A Case Study

Yaolong Ju
McGill University
yaolong.ju@mail.mcgill.ca

Sylvain Margot
McGill University
sylvain.margot@mail.mcgill.ca

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

Ichiro Fujinaga
McGill University
ichiro.fujinaga@mcgill.ca

Abstract

The computational study of figured bass remains an under-researched topic, likely due to the lack of machine-readable datasets. This paper is intended to address the paucity of digital figured bass data by 1) investigating procedures for systematically annotating symbolic music files with figured bass, and 2) producing and releasing a model annotated dataset as an illustration of how these procedures can be applied in practice. We introduce the Bach Chorales Figured Bass dataset, which includes 103 chorales composed by Johann Sebastian Bach that includes both the original music and figured bass annotations encoded in MusicXML, **kern, and MEI formats.

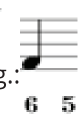
Introduction

Figured bass (FB) is a type of music notation that uses numerals and other symbols to indicate intervals to be played relative to a bass note [1]. FB was commonly used in Baroque music, and served as a guide to keyboards, strings, and other instruments improvising the *basso continuo* accompaniment. Not only does FB serve as a guideline for performers, it also reveals insights into the chords and harmonic rhythm intended by composers, beyond what is readily available in the notes themselves.

Encoding figured bass

Despite its seeming simplicity, encoding FB is not a trivial task. This section investigates (A) the extent to which musicXML, **kern, and MEI support FB, and (B) how well FB annotations are preserved when translating from one symbolic format to another.

Although the majority of FB consists only of numerals and accidentals, our examination of the Bach chorales in the Neue Bach Ausgabe (NBA) critical edition [2] revealed three additional types of notation:

1. figures with slashes (augmented or diminished intervals), e.g.: 
2. figures with continuation lines (prolongation of the harmony), e.g.: 
3. multiple figures over a stationary bass, e.g., 6–5 over the same bass note, e.g.: 

We chose BWV 33.6[1] as the basis for a case study on how well these types of notation can be encoded and translated, as it contains all three of these elements. We used MuseScore (v.3.3.2) to encode[2] the FB in musicX-

1 We referred to the Neue Bach Ausgabe (NBA) critical edition [2] for FB encodings.

2 All the encoded symbolic files are available at https://github.com/juyaolongpaul/Bach_chorale_FB/tree/master/FB_source. We chose GitHub because of its capability of version control.

ML,[3] and a text editor for **kern[4] and MEI.[5] No problems were encountered encoding but there were some issues translating between the three formats.

In general, the standard FB notation (numbers and accidentals) were properly preserved when translating between the three file formats, except for MEI to musicXML or to **kern, where all FB information was lost in both cases. There were also some additional issues with the three special types of notation introduced above as shown in Table 1 and described in more detail below.

	musicXML	**kern	MEI
musicXML		*musicxml2hum* (1): yes (2): no (3): yes	*Verovio* (1): no (2): no (3): no
**kern	*hum2xml* (1): no (2): no (3): no		*Verovio* (1): partially yes (2): yes (3): yes
MEI	*music21* (1): no (2): no (3): no	*mei2hum* (1): no (2): no (3): no	

Table 1: The results of the file translation for special cases. The first column indicates the original format, and subsequent columns indicate target formats. We examined the FB elements (1) to (3) mentioned above. The first row of each cell indicates the software used for the translation [3]. "Yes" means the translation was successful.

*MusicXML to **kern (musicxml2hum)[6]:* (2) the continuation line could not be translated, and the resulting **kern file had syntactical errors. Translations worked for chorales with no continuation line.

MusicXML to MEI (Verovio)[7]: (1) accidentals and slashes were all missing; (2) continuation lines were missing; (3) although all figures were preserved, they all shared the same "tstamp" value, which should be different.

***kern to musicXML (hum2xml)[8]:* (1) slashes were not translated properly, including (2) continuation lines, and (3) figures over a stationary bass were partially lost. The reason is that FB is translated as lyrics, rather than the "<figure-bass>" tag musicXML natively supports for FB encodings.

***kern to MEI (Verovio):* (1) although "6" with backslashes were correctly translated they could not be rendered properly using Verovio.

*MEI to musicXML (music21)[9] and MEI to **kern (mei2hum)[10]*: all FB information was lost.

3 FB encoding instructions for musicXML https://musescore.org/en/handbook/figured-bass
4 FB encoding instructions for **kern https://doc.verovio.humdrum.org/humdrum/figured_bass/
5 FB encoding instructions for MEI https://music-encoding.org/guidelines/v4/elements/fb.html
6 https://github.com/craigsapp/humlib
7 https://github.com/rism-ch/verovio
8 https://github.com/craigsapp/humextra
9 https://github.com/cuthbertLab/music21 (v. 5.1.0)
10 https://github.com/craigsapp/humlib/

Bach Chorales Figured Bass Dataset

We present Bach Chorales Figured Bass dataset (BCFB),[11] a dataset we constructed containing FB encodings in musicXML, **kern, MEI, and MIDI formats for 103 Johann Sebastian Bach chorales.[12] We began with an existing **kern edition [4], which is based on the fourth printed edition of the 371 chorales [5] and does not contain any FB. We automatically translated the music from **kern into musicXML with music21.[13] Of the 371 chorales, we manually added FB encodings to 103 chorales with FB indicated in the NBA edition[14] using MuseScore (v. 3.3.2). We also made some changes to match the NBA edition such as transposing, changing the meter, pitch, and duration of certain notes, and adding a fifth voice.

We also used our findings from above to produce FB encodings for other symbolic formats. We started with our master musicXML files, and translated them into **kern files, with minor manual corrections to the chorales with FB continuation lines. We then obtained MEI files from the **kern files. This diversity of symbolic formats offers researchers the opportunity to use the format best suited to their preferred software.[15]

We hope that BCFB will facilitate computational studies, such as comparative studies on the temporal development of Bach's FB and harmonic organization, and that it will be of use for applications such as teaching computers to arrange FB for unfigured chorales.

In the future, we will focus on adding symbolic encodings (musical content and FB) for Bach chorales beyond the 371 Breitkopf & Härtel edition,[16] with the goal of producing a comprehensive symbolic dataset of Bach chorales.

Works cited

[1] Williams, Peter, and David Ledbetter. "Figured Bass" *Grove Music Online.* https://doi.org/10.1093/gmo/9781561592630.article.09623.

[2] Bach, Johann Sebastian, et al. *Neue Ausgabe sämtlicher Werke*. Bärenreiter, 1954–2007.

[3] Nápoles López, Néstor, Gabriel Vigliensoni, and Ichiro Fujinaga. "The Effects of Translation Between Symbolic Music Formats: A Case Study with Humdrum, Lilypond, MEI, and MusicXML" presented at the Music Encoding Conference, Vienna, Austria, May 29-June 1, 2019.

[4] Condit-Schultz, Nathaniel, Yaolong Ju, and Ichiro Fujinaga. "A Flexible Approach to Automated Harmonic Analysis: Multiple Annotations of Chorales by Bach and Prætorius" in *Proceedings of the 19th International Society for Music Information Retrieval Conference* (ISMIR 2018), 66–73.

[5] Bach, Johann Sebastian. *371 vierstimmige Choralgesänge*. Breitkopf & Härtel, 1870.

11 Available at: https://github.com/juyaolongpaul/Bach_chorale_FB.

12 The complete reference table is available at https://github.com/juyaolongpaul/Bach_chorale_FB/blob/master/Reference%20Table.xlsx.

13 We asked a music theorist to compare the translated musicXML against the original **kern, and found no significant differences in musical content.

14 Eight of the 111 chorales were omitted for encoding reasons (instrumental interlude chorales, two-voice chorales, and bass independent chorales), and will be treated in a later phase of our project.

15 If only one format were offered, but is not supported by a given piece of research software, then it would need to be converted to the format supported by the software, which could lead to a loss of FB information, as discussed above.

16 There seem to be 69 extra chorales attributed to Bach http://www.bach-chorales.com/ChoralesNotInRiemenschneider.htm

Crafting TabMEI, a Module for Encoding Instrumental Tablatures

Reinier de Valk
Goldsmiths, University of London
r.f.de.valk@gold.ac.uk

David Lewis
Goldsmiths, University of London
d.lewis@gold.ac.uk

Tim Crawford
Goldsmiths, University of London
t.crawford@gold.ac.uk

Ryaan Ahmed
Digital Humanities, MIT
rahmed@mit.edu

Laurent Pugin
RISM CH, Switzerland
laurent.pugin@rism-ch.org

Johannes Kepper
University of Paderborn
kepper@edirom.de

Abstract

In this progress report, we describe the issues encountered during the design and implementation of TabMEI, a new MEI module for encoding instrumental tablatures. We discuss the main challenges faced and lay out our workflow for implementing the TabMEI module. In addition, we present a number of example encodings, and we describe anticipated applications of the module.

Introduction

A substantial part of Western art music for plucked, bowed, and keyboard instruments from roughly the early 16th to the late 18th century is notated in *tablature*, a prescriptive notation system that provides the actions a player must take rather than a description of the sounds these actions produce [1]. The mid-20th century saw a revival of tablature for plucked instruments — principally the same as the earlier system, but now for modern (electric) guitar and bass guitar — with the rise of popular music, enabling a large audience to reproduce its favourite music. With the emergence of the personal computer and, especially, the internet in the late 20th century, enormous amounts of user-created tablature — now in various *digital* formats, and increasingly linked to performance material (audio, video) — have become available. Music in tablature, in short, is a force to be reckoned with.

Yet, with the exception of a handful of recent attempts [2, 3, 4, 5, 6, 7, 8], large-scale computational research into music in tablature is lagging behind. We hypothesise that this is to a large extent due to the lack of a suitable digital format capable of encoding not only the explicit, but also the implicit and the contextual information conveyed by a piece in tablature. We think that MEI, which "brings together specialists from various music research communities [...] in a common effort to define best practices for representing a broad range of musical documents and structures" is such a format.[1] In this paper, we describe TabMEI, a module modelling the various tablature variants, to be included into MEI.

At this early stage, TabMEI focuses on tablature for plucked instruments, and includes historical lute tablature in three different types (Italian, French, and German) and tablature for the modern (electric) guitar. We do not yet attempt to model historical guitar or keyboard tablatures, which bring their own challenges. We aim to implement a basic set of elements and attributes — reusing, in the spirit of MEI, existing ones as much as possible — that cover most of the repertories and their performance techniques to a usable level.

In what follows, we discuss the main challenges faced, illustrated where appropriate with real-life examples (Section 2); our workflow for designing and implementing the TabMEI module (Section 3); three example encodings addressing some of the aforementioned challenges (Section 4); anticipated applications of the module (Section 5); and, finally, several of the many avenues of future work (Section 6).

1 https://music-encoding.org/

Challenges

Designing and implementing a new MEI module involves considerable challenges. Below, we describe five such challenges.

First, there is the issue of reconciling proposed new MEI elements and attributes with existing ones: are they really needed if MEI already contains mechanisms that model highly similar concepts? This applies at the most basic level: as Figure 1 illustrates, like mensural forms of music, music in tablature consists of a staff-like object containing symbols (notes) possibly arranged in vertical events (chords). Despite the fact that the 'staff' is now a visual representation of the courses (i.e., strings or string pairs) on the instrument, most of the elemental building blocks of MEI — `<staff>`, `<layer>`, `<note>`, and `<chord>`, as well as many of their attributes — can either be reused or be repurposed.

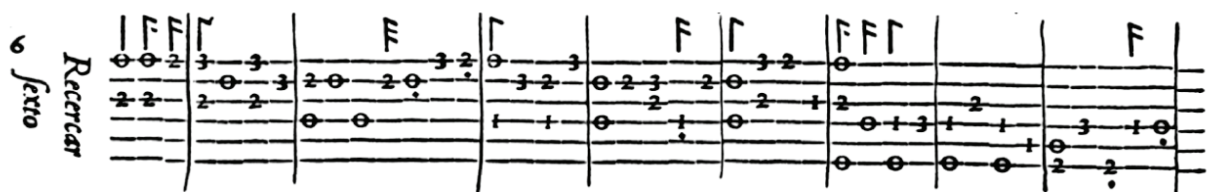



Figure 1: Giovanni Maria da Crema, *Intabolatura de lauto, Libro primo* (Venice, 1546). *Recercar sexto*, first system. Italian lute tablature with numbers indicating the frets and lines indicating the courses to be played.

Second, modern guitar tablature contains a substantial range of indications of very common performance techniques particular to the instrument (e.g., various legato techniques such as *hammer-on*, *pull-off*, and *slide*; string bending techniques; or articulation techniques such as *palm muting* or vibrato). Often, these require the introduction of new, idiomatic concepts; an example is a 'virtual' note reflecting the current 'state' of a note whose pitch is being inflected (bent) while retaining properties of that initial note. Figures 2 and 3 show examples of such techniques and concepts.

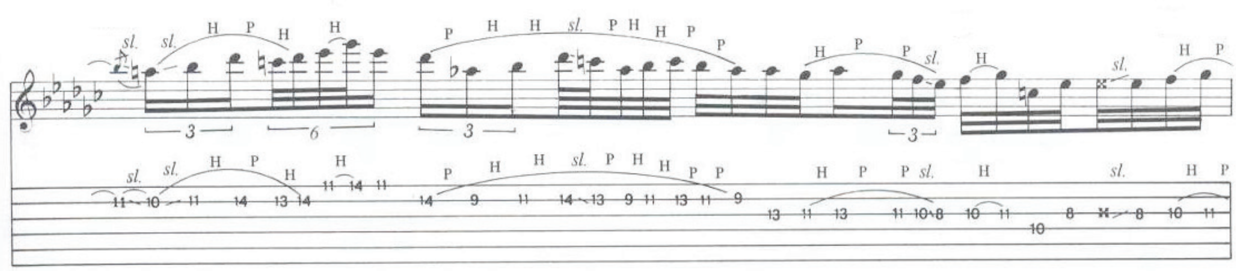



Figure 2: Joe Satriani, *Surfing with the Alien* (Relativity Records, 1987). *Ice 9*, fragment. Modern guitar tablature with a transcription in CMN superimposed. The fragment displays examples of (combinations of) the legato techniques hammer-on (*H*), pull-off (*P*), and slide (*sl.*).

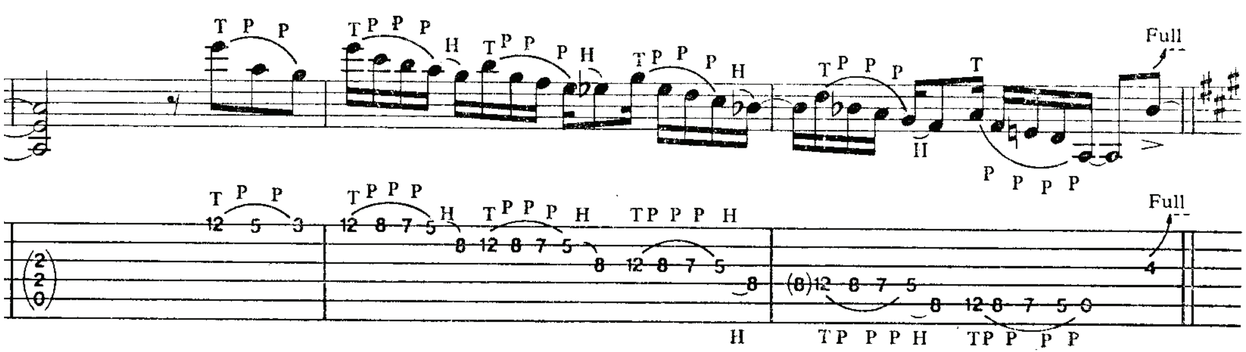


Figure 3: Van Halen, *1984* (Warner Records Inc., 1984). *Hot for Teacher*, fragment. Modern guitar tablature with a transcription in CMN superimposed. In addition to examples of (combinations of) the legato techniques hammer-on (*H*) and pull-off (*P*), the fragment displays examples of the right-hand finger tapping technique (*T*) and of the string bending technique (arrow with *Full*). The note following the last note in the example (not shown) is a 'virtual' note reflecting the current 'state' of that last, bent note.

Third, as Figures 2 and 3 show, modern guitar tablature is often accompanied by a transcription into CMN, which may contain relevant information, added by the transcriber, that is only implicitly or ambiguously present in the tablature (e.g., the exact duration of a note). How should such different levels of objectivity be modelled?

Fourth, when dealing with online tablatures in ASCII (plain text) format, one sees a high variance in quality and, since there is no notational standard and anyone can make their own encoding with just a text editor, in representation. Both complicate, among other things, any necessary data preprocessing.

Fifth, German lute tablature, which, as Figure 4 shows, contains no staff but represents each fret-course coordinate by a unique symbol, requires a different rendering paradigm. Although this presents a challenge now, the experience gained modelling this type of tablature will be useful when dealing with keyboard tablatures later.

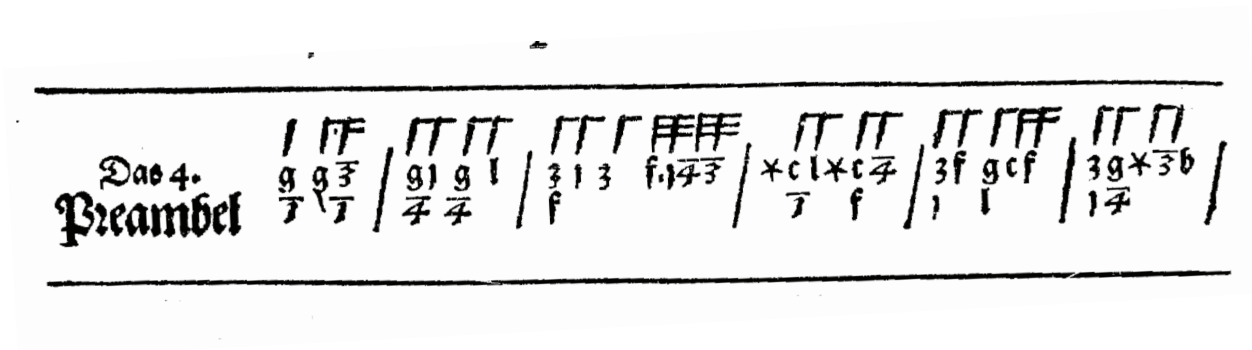


Figure 4: Hans Gerle, *Eyn newes sehr künstlichs Lautenbuch* (Nuremberg, 1552). *Das 4. Preambel*, first system. German lute tablature with unique symbols indicating the fret-course coordinates to be played. This is the same piece as the one shown in Figure 1.

Workflow

We adopt the following workflow for designing and implementing the new MEI module:

- Identify notational features specific to tablatures, always taking into account the domain — visual, gestural, or analytical — to which they belong. A feature frequently belongs to more than one domain.
- List requirements based on a set of examples. Complex and rare examples (such as, for instance, those in Figure 2 and 3) should be considered in order to validate an approach.
- Ensure that the proposed model fits the MEI approach.
- Ensure that existing MEI elements and attributes are reused when appropriate.
- Ensure that the module's granularity is in line with that of existing MEI modules (i.e., avoid a surplus of new elements and attributes).
- Prepare a customisation and the accompanying documentation, both of which are required to make a proposal (in the form of a pull request) to MEI.
- Incorporate feedback from the larger MEI community.

Example encodings

Figure 5 presents the TabMEI encoding of the antepenultimate bar of the fragment shown in Figure 1. `<measure>`, `<staff>`, and `<layer>` elements can be reused from the CMN MEI module, but `<chord>` elements have been replaced with the idiomatic `<tabGrp>` elements, which themselves contain the idiomatic `<tabRhythm>` elements (whose presence indicates that the tablature chord is provided with a rhythm symbol) and `<note>` elements. Because the duration of the rhythm symbols in lute tablature is often open to interpretation, on the `<tabGrp>` element the `@dur.ges` (and not `@dur`) attribute is used, and on the `<note>` element the `@pname` and `@oct` attributes, which depend on the tuning used, are replaced by the idiomatic `@tab.course` and @tab.fret attributes. (The tuning itself, along with the tablature type, is specified in the <staffDef>; see the TabMEI GitHub repository for full examples.)[2]

```
<measure n='7'>
  <staff n='1'>
    <layer n='1'>
      <tabGrp dur.ges='4' dots='1'>
        <tabRhythm/>
        <note tab.course='6' tab.fret='0' xml:id='m7.n1'/>
        <note tab.course='4' tab.fret='2' xml:id='m7.n2'/>
        <note tab.course='1' tab.fret='0' xml:id='m7.n3'/>
      </tabGrp>
      <tabGrp dur.ges='8'>
        <tabRhythm/>
        <note tab.course='3' tab.fret='0' xml:id='m7.n4'/>
      </tabGrp>
      <tabGrp dur.ges='4'>
        <tabRhythm/>
        <note tab.course='3' tab.fret='1' xml:id='m7.n5'/>
        <note tab.course='1' tab.fret='0' xml:id='m7.n6'/>
      </tabGrp>
      <tabGrp dur.ges='4'>
        <note tab.course='3' tab.fret='3' xml:id='m7.n7'/>
      </tabGrp>
    </layer>
  </staff>
</measure>
```

Figure 5: TabMEI encoding of Figure 1, antepenultimate bar.

Figure 6 presents the TabMEI encoding of the second half of the last bar of the fragment shown in Figure 3. It shows the reuse of the <dir> control event, now with the value 'tap-fing' for the idiomatic @technique attribute, to encode the right-hand finger tapping technique; the reuse of the <slur> control event to encode the legato techniques hammer-on and pull-off, and the use of the idiomatic <pitchInflection> control event to encode the string bending technique.

```
<measure n='3' right='dbl'>
  <staff n='1'>
    <layer n='1'>
      <beam>
        ...
        <tabGrp dur='16'>
          <note tab.course='5' tab.fret='12' xml:id='m3.n7'/>
        </tabGrp>
      </beam>
      <beam>
        <tabGrp dur='16'>
          <note tab.course='5' tab.fret='8' xml:id='m3.n8'/>
        </tabGrp>
        <tabGrp dur='16'>
          <note tab.course='5' tab.fret='7' xml:id='m3.n9'/>
        </tabGrp>
        <tabGrp dur='16'>
```

2 https://www.github.com/music-encoding/tablature-ig/

```
        <note tab.course='5' tab.fret='5' xml:id='m3.n10'/>
      </tabGrp>
      <tabGrp dur='16'>
        <note tab.course='5' tab.fret='0' tie='i' xml:id='m3.n11'/>
      </tabGrp>
    </beam>
    <beam>
      <tabGrp dur='8'>
        <note tab.course='5' tab.fret='0' tie='t' xml:id='m3.n12'/>
      </tabGrp>
      <tabGrp dur='8'>
        <note tab.course='3' tab.fret='4' tie='i' xml:id='m3.n13'/>
      </tabGrp>
    </beam>
  </layer>
</staff>
...
<dir technique='tap-fing' startid='m3.n7'>T</dir>
<slur startid='m3.n7' endid='m3.n11' show.dirmark='true'/>
<pitchInflection startid='m3.n13' endid='m4.n1' dis='2'>
Full</pitchInflection>
</measure>
```

Figure 6: TabMEI encoding of Figure 3, second half of last bar.

Figure 7, finally, presents the TabMEI encoding of the first bar of the fragment shown in Figure 4. Apart from the correction, the only difference in material usage with the encoding shown in Figure 5 is the additional use of the idiomatic `<fretGlyph>` element on the `<note>` element, which facilitates the encoding of unique symbols for fret-course coordinates using the `@symbol` and `@symbol.mod` attributes. (For the sake of brevity of the example, the scribal error — the note with `@xml:id='m1.n4'` should move one `<tabGrp>` to the left — has not been corrected. For the full, corrected, example see the TabMEI GitHub repository.)

```
<measure n='1'>
  <staff n='1'>
    <layer n='1'>
      <tabGrp dur.ges='2'>
        <tabRhythm/>
        <note tab.course='6' tab.fret='0' xml:id='m1.n1'>
          <fretGlyph symbol='1' symbol.mod='strikethrough'/>
        </note>
        <note tab.course='4' tab.fret='2' xml:id='m1.n2'>
          <fretGlyph symbol='g'/>
        </note>
      </tabGrp>
      <tabGrp dur.ges='4' dots='1'>
        <tabRhythm/>
        <note tab.course='4' tab.fret='2' xml:id='m1.n3'>
          <fretGlyph symbol='g'/>
        </note>
      </tabGrp>
      <tabGrp dur.ges='8'>
        <tabRhythm/>
        <note tab.course='6' tab.fret='0' xml:id='m1.n4'>
          <fretGlyph symbol='1' symbol.mod='strikethrough'/>
        </note>
        <note tab.course='6' tab.fret='2' xml:id='m1.n5'>
          <fretGlyph symbol='3' symbol.mod='strikethrough'/>
```

```
      </note>
    </tabGrp>
  </layer>
</staff>
</measure>
```

Figure 7: TabMEI encoding of Figure 4, first bar.

Applications

The TabMEI module has several immediate applications. First, a simple Verovio tablature renderer, taking TabMEI as input, exists.[3] It is compatible with the Verovio CMN and mensural music renderer — meaning that tablature can be displayed together with music in CMN (e.g., a transcription of the tablature) or mensural music (e.g., a vocal part in a lute song) flexibly. An example of the former is shown in Figure 8. The renderer facilitates basic playback.

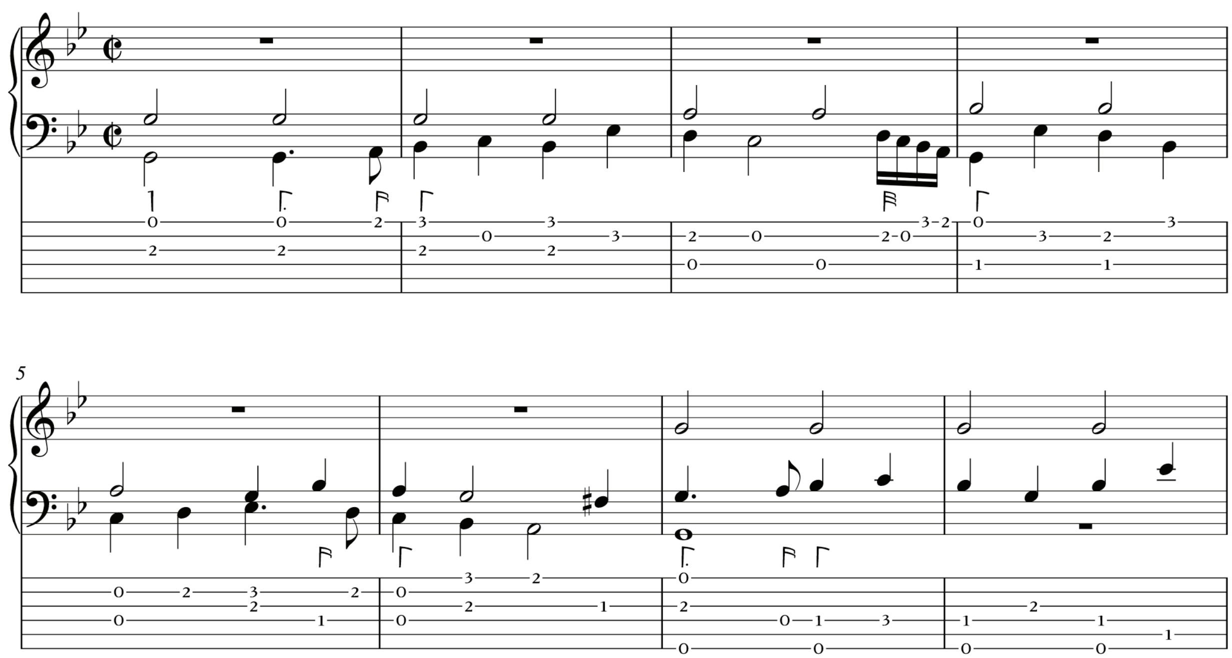



Figure 8: Verovio rendering of the tablature fragment shown in Figure 1, with a transcription in CMN superimposed.

Second, using a workflow involving the music21 tablature toolbox [2, 3] and a tablature mapping algorithm [8] or a voice separation model [7],[4] we can directly compare 16th-century lute intabulations — arrangements of vocal works — with their vocal models. Third, 'internet tabs' (i.e., online tablatures using an ASCII character set) can be ingested through the music21 tablature toolbox, displayed elegantly with Verovio, and analysed on a large scale, or connected to other digital datasets, for example through linked data techniques [6, 9, 10].

3 https://www.github.com/rism-ch/verovio/

4 https://www.web.mit.edu/music21/

Future work

In this early stage, there are many lines of future work to be explored. The most obvious — and most demanding — is to be more complete both in the coverage of repertories (e.g., for the historical guitar, or for the various historical keyboard instruments) and performance techniques. Furthermore, existing Standard Music Font Layout (SMuFL) fonts for displaying historical tablatures are incomplete, and should be extended;[5] this requires a discussion with SMuFL developers. Useful features, for example in the context of education or the preparation of scholarly or performance editions, would be interactive authoring and editing, and ingestion from a wider range of formats. Finally, the support of playback via soundfonts is envisaged.

Works cited

[1] Dart, Thurston, John Morehen, and Richard Rastall. "Tablature" in *The New Grove Dictionary of Music and Musicians*, ed. Stanley Sadie (2nd ed., vol. 24). London: Macmillan, 2001, 905-14.

[2] Ahmed, Ryaan, Reinier de Valk, Tim Crawford, and David Lewis. "A digital toolbox for exploring lute tablature" presented at the 47th Medieval and Renaissance Music Conference, Basel, Switzerland, July 3-6, 2019.

[3] Ahmed, Ryaan, Reinier de Valk, Tim Crawford, and David Lewis. "Hundreds of thousands of pieces in MEI: Encoding tablatures at scale" presented at the Music Encoding Conference, Vienna, Austria, May 29-June 1, 2019.

[4] Crawford, Tim, Jessica Schwartz, David Lewis, and Richard Lewis. "Encoding music as people play it: MEI and the role of tablatures in capturing musical performance" presented at the Music Encoding Conference, Montreal, QC, Canada, May 17-19, 2016.

[5] Lewis, David, Tim Crawford, and Daniel Müllensiefen. "Instrumental idiom in the 16th century: Embellishment patterns in arrangements of vocal music" in *Proceedings of the 17th International Society for Music Information Retrieval Conference* (ISMIR 2016), 524-30.

[6] Lewis, Richard J., Tim Crawford, and David Lewis. "Exploring information retrieval, semantic technologies and workflows for music scholarship: The Transforming Musicology project" Early Music 43, no. 4 (2015), 635-47.

[7] de Valk, Reinier. "Structuring lute tablature and MIDI data: Machine learning models for voice separation in symbolic music representations". City University London, 2015. PhD dissertation.

[8] de Valk, Reinier, Ryaan Ahmed, and Tim Crawford. "JosquIntab: A dataset for content-based computational analysis of music in lute tablature" in *Proceedings of the 20th International Society for Music Information Retrieval Conference* (ISMIR 2019), 431-38.

[9] Meroño-Peñuela, Albert, Reinier de Valk, Enrico Daga, Marilena Daquino, and Anna Kent-Muller. "The Semantic Web MIDI Tape: An interface for interlinking MIDI and context metadata" in *Proceedings of the Workshop on Semantic Applications for Audio and Music* (SAAM 2018), 24-32.

[10] David M. Weigl and Kevin R. Page. "A framework for distributed semantic annotation of musical score: "Take it to the bridge!"" in *Proceedings of the 18th International Society for Music Information Retrieval Conference* (ISMIR 2017), 221-28.

5 https://www.smufl.org/

Harmalysis: A language for the Annotation of Roman Numerals in Symbolic Music Representations

Néstor Nápoles López
McGill University, CIRMMT
nestor.napoleslopez@mail.mcgill.ca

Ichiro Fujinaga
McGill University, CIRMMT
ichiro.fujinaga@mcgill.ca

Abstract

High-quality annotations of harmonic analysis are scarce [1, 2, 3, 4]. Furthermore, the existing data usually follows different conventions for spelling scale degrees, inversions, and special chords (e.g., cadential six-four).

There have been efforts for standardizing the notation of harmonic analysis annotations [5], however, these have not been very successful because: 1) there are few software tools able to parse such notations 2) as a consequence, researchers have not adopted the suggested notations and it is more frequent to find a different notation with every new dataset.

We attempt to mitigate the limitations of existing notations through the definition of a new language for harmonic analysis, which we call *harmalysis*. This language 1) provides a notation that adjusts as much as possible to the way in which researchers have annotated roman numerals in existing datasets, 2) formalizes the resulting notation into a consistent and extensible context-free grammar, 3) uses the context-free grammar to generate tools that are able to parse and validate annotations in the syntax of the language.

We make the formal definition of the language, a context-free grammar described in the Extended Backus-Naur Form (EBNF), available as an open-source repository. Within the same repository, we make available tools for parsing annotations in the harmalysis language. The tools allow the users to extract high-level semantic information from their annotations (e.g., local key, root of the chord, inversion, added intervals, whether the chord is tonicizing another key or not, etc.) and to validate the correctness of a given annotation according to the grammar of the proposed language.

The language has been designed to be easily annotated through the addition of lyrics in music notation software or-when supported by the symbolic music format-in a dedicated data structure for indications of harmony (e.g., the function tag in MusicXML, the harm tag in MEI, and a **harm spine in Humdrum). This ensures that the users adopting the language find an immediate application for it.

The *harmalysis* language

Recently, the interest for harmonic analysis and its standardization in machine-readable contexts has been revisited by academics [4] as well as developers of music notation software [6].

We follow a similar approach by presenting a new language of roman numeral analysis, which can be encoded within symbolic music representations. The new language, harmalysis, is based principally on Huron's **harm syntax, which was originally intended for accompanying music scores encoded in the Humdrum(**kern) representation. We extend this syntax by borrowing elements from the RomanText format [4], MuseScore's notation for roman numeral analysis [6], and conventions observed in existing datasets of roman numeral analysis [1, 2, 3]. As a result, the harmalysis language is a superset of the **harm syntax, which includes additional features and supports a wider range of customs of harmonic analysis.

Goals of the language

The main goal of the language is to provide a convention for the annotation of roman numeral analysis, which the human analysts can use while they encode music through music notation software (e.g., MuseScore) or text-based encodings (e.g., Lilypond). These annotations, intelligible by the automatic tools accompanying the language, can later be used in machine-readable contexts, such as music information retrieval (MIR) tasks, computational musicology, and music engraving.

As an additional goal, the language attempts to integrate all the conventions observed in harmonic analysis practices that can be assimilated. This integration, however, is restricted to maintaining a rigorous definition of the language, which should always be characterized by a formal grammar. One example of such integration is the use of numeric inversions (e.g., V65) as well as inversions denoted by letters (e.g., V7b). Each of these conventions has its strengths and weaknesses, which is why they have been-individually-adopted in the past, however, they have now been adopted within the same annotation language. This presents an additional benefit, namely, using the same language and tools to process (although with limitations) existing datasets that have used different conventions of harmonic analysis.

As most formal languages, harmalysis is driven by a number of principles, which guided the decisions made during its design.

Principles of the language

The harmalysis language attempts to be:

- Similar-looking to a textbook analysis: The language should feel intuitive to annotators who are familiar with textbook conventions of roman numeral analysis.
- Compact: The labels of the language are relatively short and adequate for human annotators, although they may be too terse for some users.
- Flexible, but consistent over flexible: The language attempts to facilitate the preferred convention of most annotators, however, the formal definition of the language implies that sometimes the annotators will have to adopt a different convention than the one they usually follow (e.g., case-sensitive scale degrees are mandatory).
- Agnostic to the symbolic music format: The language is based on plain-text annotations, it does not enforce (but also does not oppose) other data-description languages (e.g., JSON or XML), and it is not tied to a specific symbolic music format.
- Stand-alone at the level of individual labels: Each label in the language can encode enough information to disambiguate its precise meaning without relying on a configuration file, the musical context, or previous labels.
- Application-driven: The language is meant to be accompanied by tools that facilitate the extraction of high-level information from its annotations, rather than facilitate the preservation of very specific, non-conventional harmonies. Nonetheless, a feature called descriptive chords is provided for encoding non-conventional harmonies.
- Extensible: The grammar of the language will always remain open-source and open to revisions and improvements.

Conclusion

In this paper, we introduced the harmalysis language for the annotation of roman numeral analysis in symbolic music representations. The language incorporates most of the features of the **harm syntax [5] as well as other conventions for the annotation of harmonic analysis [4], which are formalized in an open-source context-free grammar. Given the formal definition of the language and the tools that we make available with it, we consider that the harmalysis language is a valuable resource for researchers encoding or utilizing harmonic analysis datasets. The latest grammar of the language and its accompanying software can be found in the following website: https://github.com/napulen/harmalysis.

Works cited

[1] Devaney, Johanna, 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), 728–34, doi:10.5281/zenodo.1417497.

[2] Nápoles López, Néstor. "Automatic harmonic analysis of classical string quartets from symbolic score" Universitat Pompeu Fabra, 2017. Masters Thesis.

[3] Neuwirth, Markus, Daniel Harasim, Fabian C. Moss, and Martin Rohrmeier. "The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets" Front. Digit. Humanit. 3 (2018), doi: 10.3389/fdigh.2018.00016.

[4] Gotham, Mark, Dmitri Tymoczko, and Michael Scott Cuthbert. "The RomanText Format: A Flexible and Standard Method for Representing Roman Numerial Analyses" in *Proceedings of the 20th International Society for Music Information Retrieval Conference* (ISMIR 2019), 123–29, doi: 10.5281/zenodo.3527756.

[5] Huron, David. "Representation: **harm — humdrum-tools 1 documentation". https://www.humdrum.org/rep/harm/ (accessed Dec. 18, 2019).

[6] "Roman Numeral Analysis (RNA)". https://musescore.org/en/handbook/3/roman-numeral-analysis-rna (accessed Dec. 18, 2019).

Do Visual Features Matter? Studies in Phylogenetic Analysis of Mensural Music

Anna Plaksin
Max Weber Stiftung | Birmingham City University
plaksin@maxweberstiftung.de

Abstract

This paper reports on the task of developing concepts for a computational analysis of the transmission of mensural music based on concepts of phylogenetic analysis. Since the analysis of transmission aims for the reconstruction of relations between sources, it focuses on the differences of rather similar items. Therefore, it is necessary to find substitution models which are optimized for distinguishing fine levels of differences and to deal with the structural ambiguities and visual variance of mensural notation.

Introduction

The model of semantic domains in music notation is not only well known in the field of music encoding but is used as a common ground in reasoning about the representation of notated music. By modelling these domains in separate attribute classes, MEI provides a powerful feature offering the possibility to depict complexities of different kinds of music notation, e.g. mensural notation. Especially the lack of stable relationship between symbols and their interpretation is easily observed when encoding mensural music, but stemmatic analysis is typically led by the concept of significance commonly embodied in focussing on substantial variants, variants in pitch and duration. However, in the case of mensural music sources, with their richness of visual variance, where the particular context affects the process of reading and deciphering as well as developments in the notational system and varying concepts in mensural theory, that distinction reaches its limits:

> "The extent to which these can be considered 'non-substantive' is questionable: the positioning of line breaks, for instance, will have an effect on an editor's interpretation of the duration of manuscript accidentals, or stem direction may actually have an effect on rhythm in certain notational styles (as in some brands of 14th-century notation)" [1, p. 143].

This paper reports on the task of developing concepts for a computational analysis of the transmission of mensural music based on concepts of phylogenetic analysis. Starting with encodings of the sources of Josquin's *Missa D'ung aultre amer* and *Tu solus qui facis mirabilia*, it raises the question, how picking properties of mensural notation affects the resulting tree.

What to compare?

One main concern in methodological design is the preservation of the research object throughout the analysis. When dealing with questions, for example, like the effects of stress on people, the first and foremost task is the quantification of stress -- how can something so vague be detected by the means of measurable qualities. And even though we're dealing with music this question still matters. The analysis of the transmission of mensural music is a task that is inherently focused on the witnesses of this tradition, the sources itself. But these sources aren't digital objects. To make them available to such a task, the digitization the information about them is an inevitable bottleneck. And in this regard representation becomes crucial: If the representation of the source becomes distorted during the process of digitization, the whole analysis becomes flawed.

But, what in that particular context is the source and what features of it need to be maintained? First of all, when dealing with the transmission of music, luckily features of physical substance of the object could be regarded as subordinated. We're not interested in the object itself but in its role as a witness of the human

interaction with it, which is writing down music notation. And by following that track, because the sources are the only remaining witnesses of this interaction, we are faced with all the particularities and lapses that come with it.
First of all, a source of a piece of music isn't just the piece of music. A source can be erroneous, even to a point where the performance of the piece of music it bears isn't possible anymore.[1] Ambiguities in the position of notes on a staff are as likely as the challenge of deciding how long that line which represents a rest actually is.

And deciphering the notation is a complex task in itself. Limiting the scope to mensural notation, one main idiosyncrasy is the ambiguous relationship between a sign, its meaning and its result on performance. For this reason, visual variance of notation is not just likely but typical.[2] Hereby, the coexistence of single note shapes and ligatures adds to the various possibilities, but with only limited complexity.

For example, the phenomenon known as color minor is a one of these special cases. Usually, in transcription to modern Common Western Music Notation it is treated equally to the *punctum augmentationis*, even if they exist commonly in close succession (see e.g. figure 1) [3, p. 138f.]. But why would both manners be used side by side? While Stanley Boorman [4, pp. 72-75] describes possibilities of further implications, Ronald Woodley [5] describes a change in notation practice around 1500. Which position is to be followed doesn't really matter in this context -- rather the range of subtleties has to be considered.

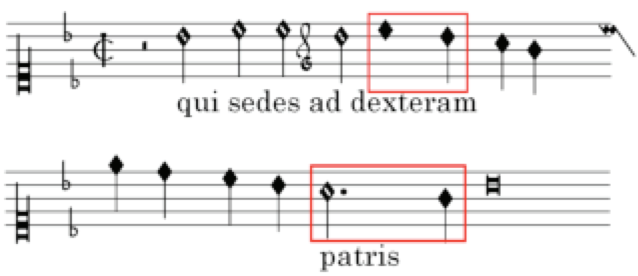


Figure 1: *Color minor* and punctuation is used in close succession. M. D'ung aultre amer, Gloria, Superius, after [VatS 41, fol. 150v].

As well, a focal point for the transmission of the *M. D'ung aultre amer* and *Tu solus qui facis mirabilia* are varying signs for the *sesquialtera* (see figure 2). It is not just a layer of visual variation but could be seen as well as a symptom of changes in the practice of musical notation [6]. As Anna Maria Busse Berger explains, the understanding of proportions evolved from a "substitute for mensuration signs" [7, p. 185] to a self-contained sign for diminution [7, pp. 182-96]. The variant reading of a *sesquialtera* including a change to *tempus perfectum* with a circle added on top of the **3** might be understood as effect of this trend. Following this argumentation, it would be essential to track this kind of variants during the whole process.

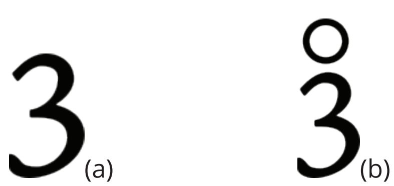


Figure 2: Different signs for sesquialtera: While (a) give no statement about the mensuration, (b) signals perfect time unambiguously [7, p. 230].

Bringing these aspects together, there are some valid conclusions concerning the machine-readable representation. When sources are the objects of our particular interest, it is necessary to allow for their idiosyncrasies to be pivotal, both unperformable corruptions in the source and the fine subtleties of notation. With the former, even if it means to accept that ‚the music' cannot be read from a manuscript [8, p. 169]. But when reading the music is impossible, a source still has its documentary value that can be captured. And moreover, con-

1 Like e.g. a suddenly ending superius in the Osanna of Dufays *Missa Se la face ay pale* in [VatS 14, fol. 35v].

2 "Mensural notation is at least partly redundant in that the scribe often has a choice of representing a certain musical content in different visual manifestations" [2, p. 58].

cerning notational idiosyncrasies, it is also possible to trace them from a point of describing the notation itself, without trying to resolve durations, reconstructing a conceptual piece of music – meant as thinking the parts together – or even suggest appropriate inflections of pitches. But the question arising from these thoughts is: Does that actually work in a computational analysis of transmission when similarity is usually estimated based on the perception of this conceptional or the aural dimension of music?

Distinction of difference

Other endeavours to use global sequence alignment for notated music [9], [10], and [11] focus on either retrieval scenarios or minimizing differences of notation, mode and/or tempo. Analysing patterns of transmission comes with a different scope. The goal of stemmatic analysis is more or less giving statements about the relationship between sources based on their variants. This means, instead of querying the most similar in a heterogeneous group, the main task is to cluster a group of rather similar objects according to their differences. To allow this clustering, we might not focus on their similarities but rather to distinguish the degree of deviation between a group of sources. Therefore, it is necessary to find substitution models which are optimized for distinguishing these fine levels of difference. But how could these models be developed without any advanced experience in measuring difference of mensural music?

On the one hand, there are mathematical models: But they are focused on similarity scores used in local sequence alignment. And they need to follow particular assumptions to be valid. The expected similarity score of an alignment of random sequences needs to be below zero while there is at least one positive score. Conversely this means it would be necessary to decide what level of similarity is to be denominated as neutral similarity: S = 0 What is different enough to be similar but similar enough to not be different?

On the other hand, there are already existing stemmata, made very cautiously e.g. while editing a certain piece of music. But analysing these shows, that every stemma is constructed strictly on its own terms. When comparing the stemmata of the joint transmission of Josquins *Missa D'ung aultre amer* and the motet *Tu solus qui facis mirabilia* – which is used as a replacement for the Benedictus and Osanna II – crucial disagreements become evident. And because of these disagreements, pre-existing stemmata cannot serve as a benchmark as well.

An obvious conclusion in addressing these challenges is using methods with few external prerequisites. First of all, a global alignment using distance-based substitution models is the chosen approach. By stating that identity as distance D = 0 the definition of neutral similarity is avoided. Moreover, a data-based process was developed for evaluating substitution models. Based on the method of surrogate data analysis [12], an approach was chosen, that scales the strength of separating levels of distance. Hereby sequences are shuffled to provide independent and identically distributed random sequences as a benchmark.

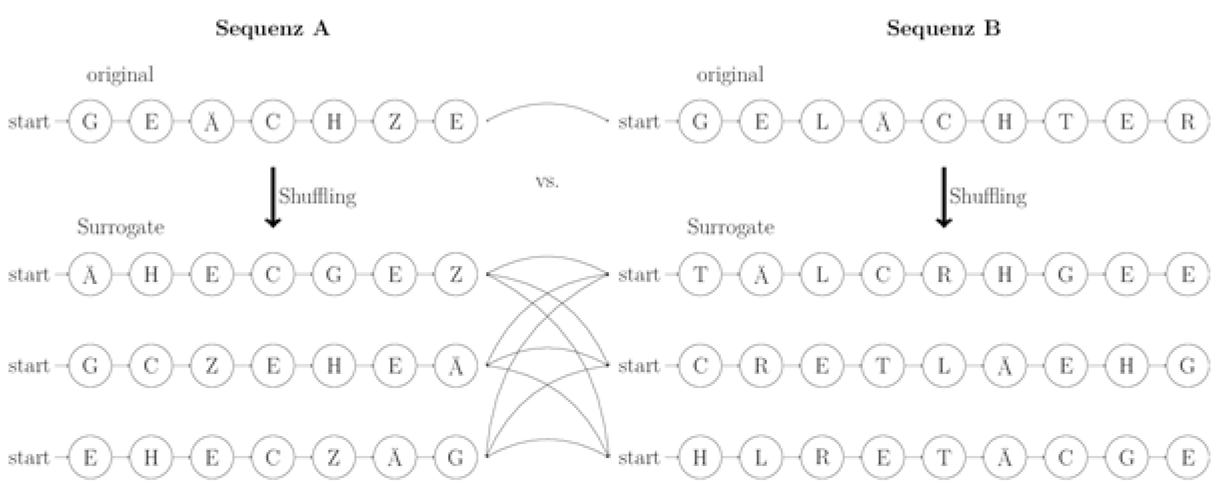


Figure 3: Figure 3: Comparing original data against independent and identically distributed random sequences.

This analysis makes use of three main preconditions:
First of all, its use in the course of finding models for the analysis of transmission depends on dissimilarity as the central criterion of stemmatic analysis. Second, the shuffling utilizes the assumption that the internal structure of a sequence is constitutive for similarity, in the way that the order of letters constitutes a word. And the third condition is, that the relative distance between original and surrogate comparisons is affected by the similarity of the original sequences. Therefore, it must be possible to estimate the dissimilarity of the original sequences by quantifying the deviance of the relative distance between these two original sequences and their shuffled surrogates. Moreover, this approach serves as the basis for an analysis of variance to detect a trend in comparing sequences, grouped by an estimated level of similarity. And observing the behaviour of a set of attributes with this test set-up can lead to an informed choice of analytical parameters.

Comparing feature sets

In this analysis of variance, the joint *M. D'ung aultre amer / Tu solus transmission* serves as the test case.[3] To detect a trend in the deviance of relative distance between original and surrogate data, groups of estimated similarity has been defined by the non-controversial relations of those two conflicting stemmata, together with arbitrarily chosen groups:

1. Different piece of music: Quis dabit capiti meo aquam
2. Different parts of the same mass section from the same source
3. Different parts of the same section from the same source
4. *Tu solus*: Mass vs. motet tradition
5. *Tu solus*: Same tradition
6. *Tu solus*: Direct dependency
7. Same part before and after scribal intervention

Since the main question that arose during the encoding of the sources is how much interpretative encoding of mensural notation is least necessary for performing, the tested parameter sets are mainly designed to capture certain states of interpretation of mensural notation.
The first state, labelled as **signbased.vis** is similar to recognising and describing symbols. Every symbol in a staff is described independently depending on the kind of symbol, mostly based on the element names and attributes used in the encoding. Regarding mensuration signs and proportion signs, only an identifier classifying the visual sign is used for further discrimination. The written pitch, which is used as a feature for notes and accidentals, can in this regard be seen as a classification of the vertical orientation within a staff – no further inflection of accidentals or *musica ficta* is intended. As well, notes and rests are merely distinguished by their types as encoded with *@dur*. In addition, notes have features regarding their coloration and the form of a ligature and their position within a ligature.

In contrast, **signbased.log** still records every symbol in the staff, but tries to capture the actual meaning of the symbols. This means, for mensuration and proportion signs, it records the Tempus, Prolatio, Modus minor and maior and the *@num/@numbase*. As well, the duration of notes and rests are resolved into relative durations, and the pitch is recorded including resolved written accidentals.

Since the aim of comparing sources makes it inevitable to follow one source as it presents itself without emendation of errors, a parameter set containing performance-related information would undermine this. Already the observance of written accidentals is a grey area, but resolving *musica ficta* is in this regard too interpretative a task. Therefore, data that needs to have taken more than a single part into consideration is excluded. Instead, another parameter set has been created as a further reduction to substantial parameters. The parameter set called **superlogical.gap** takes only notes and rests with their relative duration and resolved pitch into consideration. In this way, it mimics focussing exclusively on substantial variants. And in addition, another parameter set **signbased.all.gap** contains every created parameter.

3 For a more detailed description of the analysis, see [13].

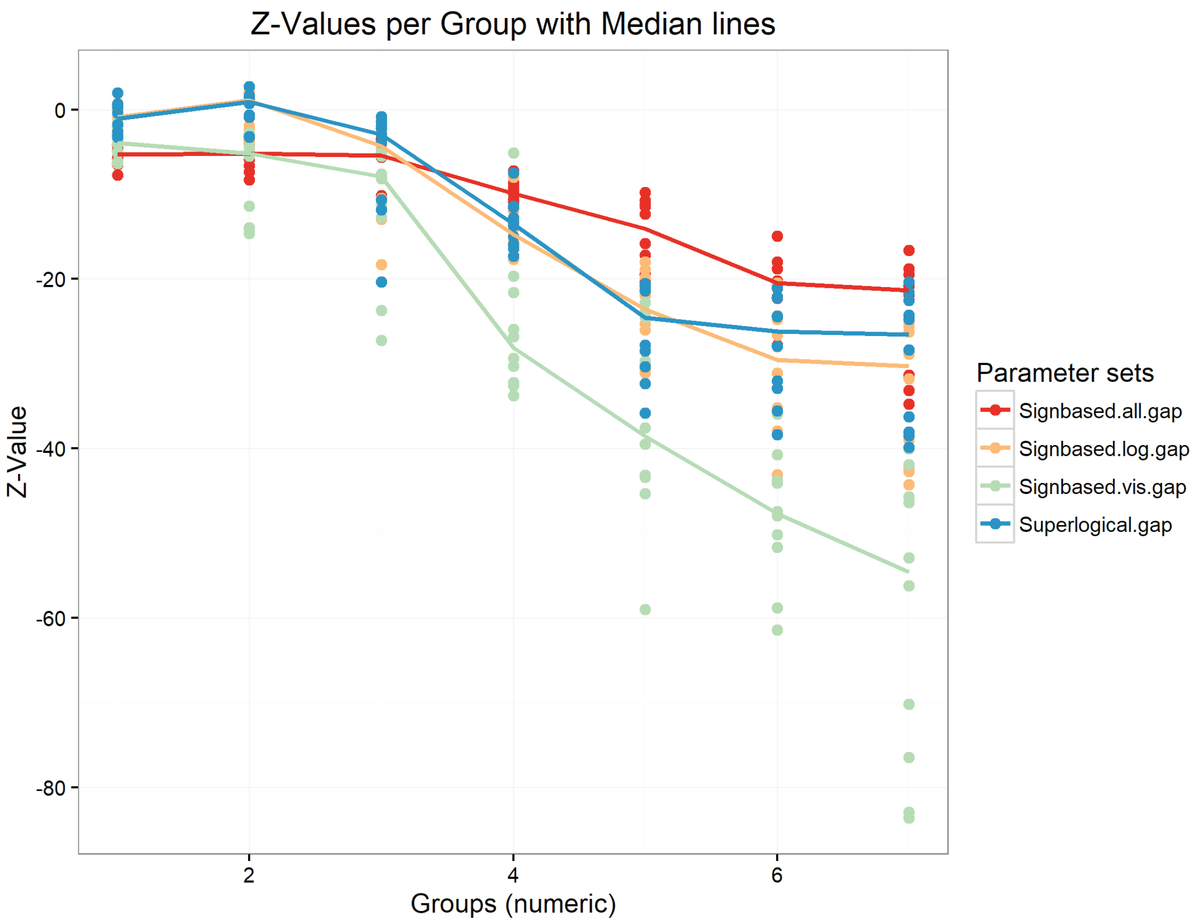


Figure 4: Deviance of distance between original and surrogate sequences per parameter set and group (see list). Median lines visualize trend.

By performing the comparison of original and surrogate sequences per parameter set and per group, several crucial observances can be made (see figure 4). The **signbased.all.gap** set not only have the slightest slope, but apart from that, the deviance between original and surrogate is already significant for the comparison of different pieces.[4] Having this group as a control group, which sets another piece of music at random against the chosen example, an acceptable parameter set must show now significant deviance from random comparisons, whereas the other sets match this demand. In conclusion, a set containing all parameters is not appropriate at all for this task.

The parameter sets **signbased.log.gap** and **superlogical.gap**, mainly focussing on the logical meaning of a symbol in the context of notation or the resulting impact as a series of notes and rests with a certain relative duration and a pitch, show a very similar behaviour. There is a visible difference between the arbitrarily chosen groups and the quite similar groups 5-7, with the group comparing different traditions in the middle. But the differences between the similar groups are hardly distinguishable. And moreover, the minimal deviance between original and surrogate is not the control group but the group comparing different voices. This could be explained by a high influence of absolute pitch. These parameters, therefore, might serve well in a setting of retrieving similar pieces from a heterogeneous group.

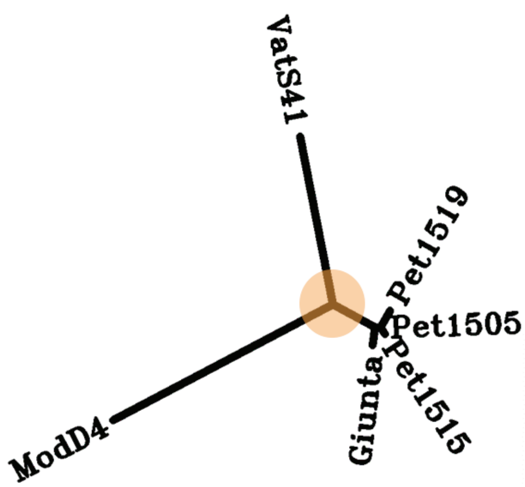
But when analysing transmission, the task is to cluster similar pieces according to their differences. In this regard, the set **signbased.vis.gap** seems more appropriate. It distinguishes well between the arbitary groups and the ‚realistic' groups and its minimum is at the control group. Moreover, it shows the steepest slope for the groups 4-7 based on realistic comparisons.

4 Using a Wilcoxon signed-rank test (α= 0.001): W = 17851.

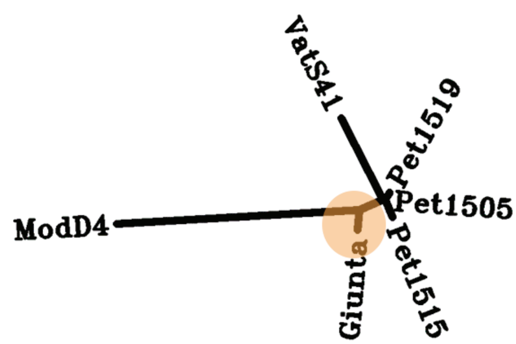
Comparing trees

But beside the results of the surrogate data analysis, it is worth to take another look at trees. As already mentioned, the transmission of the *M. D'ung aultre amer* and *Tu solus qui facis mirabilia* is of relevance because there are two conflicting stemmata [14, p. 34], [15, p. 43]. In detail, they show how much weighing different aspects and focussing on certain variants can lead to different points of view [13, pp. 79-82]. Crucial for the diverging layout is whether proportion signs are taken into consideration or not. While Noblitt ignores them, Blackburn traces one of the *sesquialtera* signs back to Petrucci' editor Petrus Castellaus [15, p. 40]. Conversely, the other variant is treated as authorial.[5] At the first glance, this might fit questions this paper is stating when observing the effect of a single visual feature on a stemma. But this is covered by another aspects. The central criterion for a stemma is usually significance, the likelihood of the same error occurring independently. The question of notational parameters is on side of the question, the other is weighing the influence – which has been done by stating authorial influence.

Therefore, it might be rather useful to start with a delimitation. The trees constructed as a part of this study are unrooted trees based on global distances of sequences.[6] Without root, they give no hint about a possible origin and in addition no information about relations to that origin. And, in contrast to a stemma, these trees don't follow any rule of significance. Every dissimilarity that has been detected by the chosen parameter set affects the tree.



(a) Signbased.log.gap.



(b) Superlogical.gap

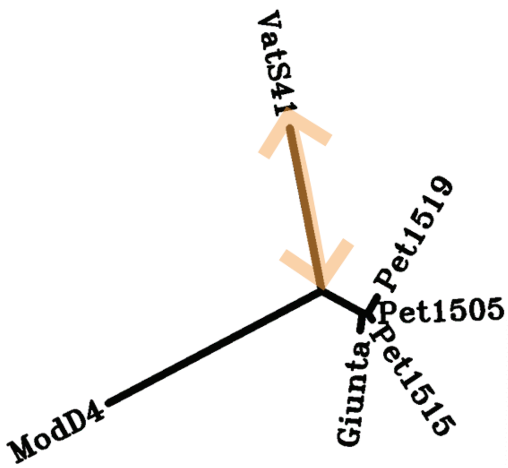
Figure 5: .Instability of unrooted trees: Superlogical.gap leads to a different layout than signbased.log.gap, disregarding that [Gio1526] is a reprint of the Petrucci prints.

Figure 5 shows the trees regarding the superius and tenor of the mass cycle[7] constructed on the basis of the *superlogical* and the *signbased logical* parameters. Obviously, the topology of both trees differs. While the printed sources are grouped together in figure 5a apart from the two manuscripts [ModD 4] and [VatS 41], the superlogical tree (figure 5b) sets only [ModD 4] apart. Moreover, it puts the Vatican source in closer relation to the three Petrucci prints than the Giunta reprint of 1526 – a highly doubtful result. In this regard, the tree based on the parameters per sign fits better to external knowledge about the sources.

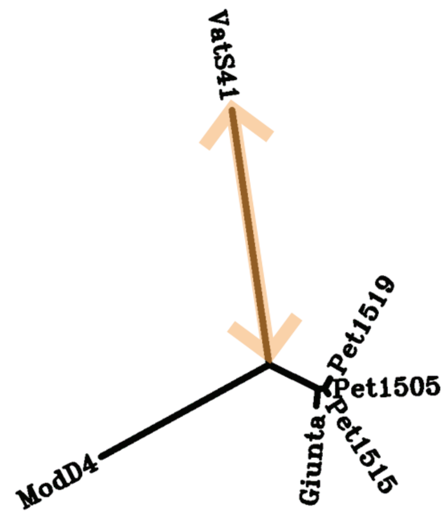
5 "Josquin himself normally used '3' to indicate sesquialtera" [15, p. 40]. As well Gaffurius attests to Josquin's use of "3" [6, p. 418f.].

6 The neighbor-joining algorithm according to [16] has been used.

7 Only superius and tenor were available in all sources.



(a) Signbased.log.gap.



(b) Signbased.vis.gap

Figure 6: .Influence of ligatures and coloration: Visual parameters affect the distance between the Cappella Sistina choir book and other mass sources.

In comparison, figure 6 compares the trees built on parameter sets either of uninterpreted notation and of signbased resolved meanings. Obviously, the layout of both trees is identical, while the edge lengths differ. In particular, the distance of the Vatican source in relation has grown. When taking a look at the sources, this result is evident. While [ModD 4] conforms mostly with the printed sources regarding the use of coloration and ligatures, the Cappella Sistina choir book makes a heavy use of coloration and ligatures, in melismatic sections of the tenor not unlikely complex multi-note ligatures. Notably, the variant *sesquialtera* signs don't change the layout of the tree, otherwise there would have been a difference between the trees derived from the signbased models – only **signbased.vis** takes the sign into account together with ligatures and coloration. Instead, it weighs much more if the model is signbased or notation agnostic.

Conclusion

First of all, I would like to start concluding about the analysis of transmission. The presented experiment shows a different usage of sequence alignment than retrieval scenarios. When reusing methods of phylogenetic analysis, the focus lies on the distinction of difference rather than on finding similarities. Therefore, other models need to be used. As well, it is obvious, that an unrooted tree constructed on global alignments must be read differently than a stemma because it is based on other conditions. But when trees are constructed based on different feature sets, the effect of certain assumptions can be made visible. Whereas a stemma usually relies on few significant variants, the showed trees take all detected differences into account.

Moreover, the results can clearly be summarized: Notation matters! The presented method of surrogate data analysis gives hints about the specific behaviour of a model. It shows, for the purpose of discriminating rather similar items, the model based on uninterpreted mensural notation provides better results than the models using resolved durations and ignoring visual variance. And while the surrogate data analysis shows no big difference for the latter, the comparison of trees favours a notation specific approach. These results not only demonstrate that it is possible to bypass the interpretative reading of music notation for the purpose of comparing differences, but as well it makes clear that notation itself can be a subject of research, e.g. in tracing changes in notational praxis. For this research, encoding is a pivotal part, but under different circumstances than e.g. for machine-readable editions or musical analysis. Encoding, used as a structured description of notation and its idiosyncrasies can serve the analysis of notation and its interpretation.

With that purpose in mind, the separation of semantic domains as provided in with MEI is a powerful tool. For the matter of encoding mensural notation, this is a highly intricate task, since it means to differentiate carefully the levels of interpretation. In the case of tracing the ambiguous relationship of sign and meaning, a procedural approach would be favourable, classifying the sign, representing its instructional value and illustrate resulting consequences in performance.

List of Sources

[Gio1526] Jacopo Gionta, Giovanni Giacomo Pasoti, and Valerio Dorico, eds. Missarum Josquin Liber secundus. Rom, 1526. http://data.onb.ac.at/rec/AC09229566

[VatS 14] MS Capp.Sist.14. Biblioteca Apostolica Vaticana. Rome, Italy. https://digi.vatlib.it/mss/detail/Capp.Sist.14

[VatS 41] MS Capp.Sist.41. Biblioteca Apostolica Vaticana. Rome, Italy. https://digi.vatlib.it/mss/detail/Capp.Sist.41

[ModD 4] Ms. Mus. IV. Archivio Capitolare. Modena, Italy.

[Pet1505] Ottaviano Petrucci, ed. Missarum Josquin Liber secundus. Venedig, 1505. http://diglib.hab.de/drucke/2-8-musica-2s/start.htm

[Pet1515] Ottaviano Petrucci, ed. Missarum Josquin Liber secundus. Ave maris stella. Hercules dux ferrarie. Malheur me bat. La mi baudichon. Una musque de buscaya. Dung aultre amer. Fossombrone, 1515. http://data.onb.ac.at/rec/AC09319724

[Pet1519] Ottaviano Petrucci, ed. Missarum Josquin Liber secundus. Ave maris stella. Hercules dux ferrarie. Malheur me bat. La mi baudichon. Una musque de buscaya. Dung aultre amer. Fossombrone, [1519]; 1515. http://data.onb.ac.at/rec/AC09170773

Works Cited

[1] Dumitrescu, Theodor, and Marnix van Berchum. "The CMME Occo Codex Edition: Variants and Versions in Encoding and Interface" in *Digitale Edition zwischen Experiment und Standardisierung*, eds. Peter Stadler and Joachim Veit. Beihefte zu editio. Tubingen: Niemeyer, 2009, 129-46.

[2] Schmidt, Thomas. "Making Polyphonic Books in the late Fifteenth and early Sixteenth Centuries" in *The Production and Reading of Music Sources*, eds. Thomas Schmidt and Christian Thomas Leitmeir (Epitome Musical). Turnhout: Brepols Publishers, 2018, 1-98.

[3] Apel, Willi. *Die Notation der polyphonen Musik: 900 - 1600.* 5. Au. Wiesbaden: Breitkopf & Härtel, 2006.

[4] Boorman, Stanley. "Notational Spelling and Scribal Habit", in *Datierung und Filiation von Musikhandschriften der Josquin-Zeit*, ed. Ludwig Finscher. Wolfenbütteler Forschungen. Wiesbaden: Harrassowitz, 1983, 65-109.

[5] Woodley, Ronald. "Minor coloration revisited: Okeghem's Ma bouche rit and beyond" in *Théorie et analyse musicales*, ed. Anne-Emmanuelle Ceulemans (Publications d'histoire de l'art et d'archéologie de l'Université Catholique de Louvain. Musicologica neolovaniensia. Studia). Louvain-la-Neuve, 2001, 39-63.

[6] Blackburn, Bonnie J. "The Sign of Petrucci's Editor" in *Venezia 1501: Petrucci e la stampa musicale*, eds. Giulio Cattin and Patrizia Dalla Vecchia (Serie III. Studi musicologici. B, Atti di convegni / 6). Venezia, 2005, 415-30.

[7] Busse Berger, Anna Maria. *Mensuration and proportion signs: Origins and evolution*. Oxford monographs on music. Oxford: Clarendon Press, 1993.

[8] Boorman, Stanley. "The Uses of Filiation in Early Music" Text 1 (1981), 167-84, http://www.jstor.org/stable/30234249

[9] van Kranenburg, Peter. "A Computational Approach to Content-Based Retrieval of Folk Song Melodies". Utrecht University, 2010. PhD dissertation. http://hdl.handle.net/20.500.11755/8436a210-ceeb-4c66-ba02-ffe5c7e66a42

[10] Mongeau, Marcel, and David Sanko. "Comparison of Musical Sequences" Computers and the Humanities 24, no. 3 (1990), 161-75, http://www.jstor.org/stable/30200223

[11] van Nuss, Jelmer, Geert-Jan Giezeman, and Frans Wiering. "Searching musical incipits by means of sequence alignment" presented at the Music Encoding Conference, Tours, France, May 16-19, 2017.

[12] Theiler, James, Stephen Eubank, André Longtin, and Bryan Galdrikian. "Testing for nonlinearity in time series: the method of surrogate data" Physica D: Nonlinear Phenomena 58, nos. 1-4 (1992), 77-94. https://doi.org/10.1016/0167-2789(92)90102-S.

[13] Plaksin, Anna. "Modelle zur computergestützten Analyse von Überlieferungen der Mensuralmusik: Empirische Textforschung im Kontext phylogenetischer Verfahren". Darmstadt: Technische Universität Darmstadt, 2019. PhD dissertation.

[14] Josquin. *Masses based on secular polyphonic songs: Critical commentary*, ed. Thomas Noblitt. New Josquin Edition 7. Utrecht: Vereniging voor Nederlandse Muziekgeschiedenis, 1997.

[15] Josquin. *Motets on non-biblical texts 2: Critical commentary*, ed. Bonnie J. Blackburn. New Josquin Edition 22. Utrecht: Vereniging voor Nederlandse Muziekgeschiedenis, 2003.

[16] Studier, James A., and Karl J. Keppler. "A note on the neighbor-joining algorithm of Saitou and Nei" Molecular biology and evolution 5, no. 6 (1988), 729-31. 10.1093/oxfordjournals.molbev.a040527.

Computer-Aided Analysis Across the Tonal Divide: Cross-Stylistic Applications of the Discrete Fourier Transform

Jennifer Diane Harding
Florida State University
jdharding@fsu.edu

Abstract

The discrete Fourier transform is a mathematically robust way of modeling various musical phenomena. I use the music21 Python module to interpret the pitch classes of an encoded musical score through the discrete Fourier transform (DFT). This methodology offers a broad view of the backgrounded scales and pitch-class collections of a piece. I have selected two excerpts in which the composers are very frugal with their pitch class collections—one in a tonal idiom, the other atonal. These constrained vocabularies are well suited for introducing the DFT's methodological strengths as they pertain to score analysis.

Introduction

The discrete Fourier transform (DFT) has recently gained traction in the music theory community as a mathematically robust way of modeling various musical phenomena. Theorists have used the DFT to model harmonic motion [1], set class similarity [2], meter [3], and the analysis of larger musical excerpts [4]. The work presented in this article falls into this last category. I use the music21 Python module to interpret the pitch classes of an encoded musical score through the discrete Fourier transform. This methodology offers a broad view of the backgrounded scales and pitch-class collections of a piece, what Dmitri Tymoczko refers to as "macroharmony"[5]. I have selected two excerpts in which the composers are very frugal with their pitch class collections—one in a tonal idiom, the other atonal. The exposition of the first movement of Mozart's String Quartet No. 4 in C Major, K. 157 is almost entirely diatonic. The theme from Messiaen's *Theme and Variations for Violin and Piano* adheres strictly to two of his modes of limited transposition. These constrained and highly symmetrical (in the case of Messiaen) harmonic vocabularies are well suited for introducing the DFT's methodological strengths as they pertain to score analysis.

The discrete Fourier transform and pitch class collections

The Fourier transform (FT) is a mathematical function that decomposes an input signal into its constituent sinusoidal components. In contrast, the *discrete* Fourier transform uses an array of discrete numbers, as opposed to a continuous signal, as its input. This makes it possible to apply the DFT to the pitch-class content of musical scores: the counts of pitch classes become the input array. The results of the DFT provide information about the aural saliency of the input, which roughly correlate with the qualia of the familiar interval cycles [6]: is it chromatically clustered? Is it more "whole-tone" sounding? Is it "fifthy?" Applying the DFT to macroharmonies allows us to see a broad overview of a composition's sound.

The output of the DFT comes in the form of six non-trivial Fourier components, denoted f_1, f_2, f_3, …f_6.[1] Each component can be imagined as a circle with twelve positions or nodes where pitch classes are conceptually located, which I refer to as the *component circle*.[2] The coefficient of each component reflects how far adjacent

1 The f_0 component is always equal to the cardinality of the set. Components f_7–f_{11} are mirror images of components f_1–f_5 and are therefore redundant.

2 The number 12 here comes from examining the twelve pitch classes. In music with quarter-tones, we would use 24 nodes. If we were examining meter, we might use four nodes for the four beats of a quadruple meter, or 16 nodes if we were looking at every sixteenth-note subdivision. The number of nodes is dependent upon the length of the input array.

integers (indicating pitch classes) are separated: on the f_1 component circle, each pitch class is one node apart from its neighbor; on the f_2 component circle, pitch classes are placed every two nodes, and so on. Figure 1 shows all six of the non-trivial Fourier components as component circles.

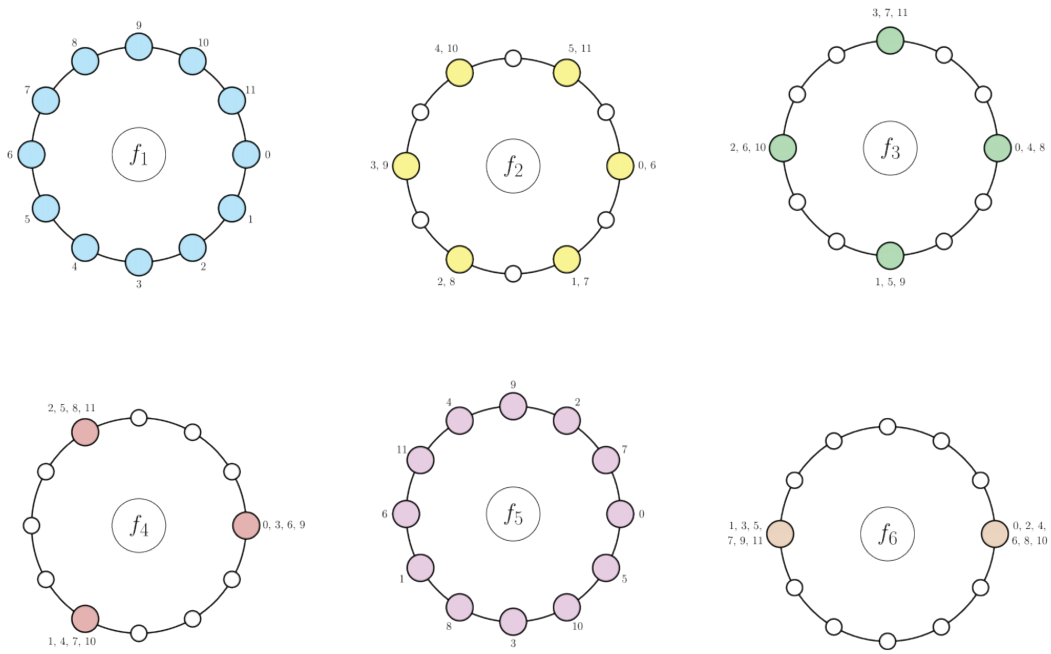
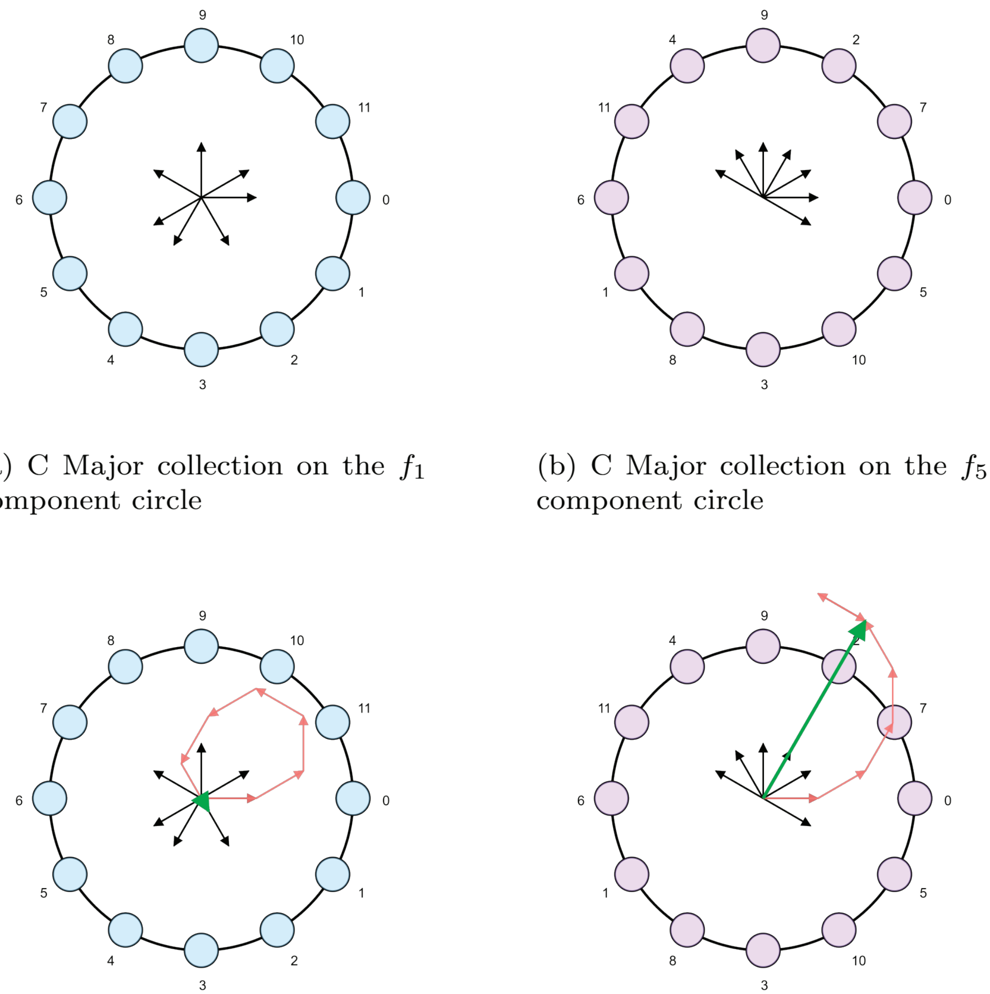


Figure 1: The six non-trivial Fourier components.

The values from a pitch-class array can be plotted on a component circle as vectors, each with a length or *magnitude* (equal to the corresponding value of the array) and a direction or *phase* (expressed as an angle).[3] The pitch class's position on the component circle determines the vector's phase. The vectors are added together by positioning them head-to-tail, resulting in a new vector from the origin of the circle to the end of the chain.[4] The magnitude and phase of this resultant vector for every Fourier component is the output of the DFT. This process is shown in Figure 2, mapping the C-major diatonic collection onto the f_1 and f_5 component circles. The resultant magnitude of the f_1 component indicates how chromatic the collection is. From the very low magnitude of 0.27, we can say that the diatonic collection is not very chromatically clustered (indeed we know that it is minimally chromatic for a septachord). In contrast, the f_5 component indicates how "fifthy" a collection is, and from the very high magnitude of 3.73, we can say that the diatonic collection is very fifthy (indeed, maximally so).

3 0° is positioned due East, as on a polar coordinate system.

4 Since vector addition is both commutative and associative, the actual order in which the vectors are added does not matter.



(a) C Major collection on the f_1 component circle

(b) C Major collection on the f_5 component circle

(c) Resultant vector has a magnitude of 0.27 and phase of -30°

(d) Resultant vector has a magnitude of 3.73 and phase of 60°

Figure 2: The C Major diatonic collection mapped onto the f_1 and f_5 component circles

Of note is the fact that on the f_5 component circle, the resultant vector for the C-major diatonic collection does not pass through pc0, the tonic (Figure 2(d)). Instead, it passes exactly through pc2 (60°), the point of symmetry for the collection. Of course, in music we rarely see exactly one instance of each pitch class together like this. Rather, some pitch classes will be omitted while others are duplicated. If we account for the number of times each pitch class is present in a passage by using multisets—sets that allow for multiple instances of each element—the phase of the resultant vector will probably not pass quite so cleanly through pc2, but rather somewhere in its vicinity. A quantizing function snaps the phase to the nearest node, making the data more comprehensible. On the f_5 component, a diatonic collection will quantize to its scale-degree 2, meaning that we can use the phase as an indicator of the key of a largely diatonic passage. Or, more accurately, we can use the phase to identify the implied key signature.

Methodology

I built my computational apparatus in Python using music21,[5] a Python module developed by Michael Cuthbert and Christopher Ariza at MIT [7]. Music21 parses many different symbolic music file formats: MusicXML, MIDI, MEI, Humdrum, and Lilypond, to name a few. The file is converted to a *stream*, the fundamental object in music21. Streams store pieces of score information (note objects, articulations, barlines, etc.) in a hierarchical structure reminiscent of XML. Each object in the stream is located at a particular *offset*—or point in time from the beginning of the piece—as measured in quarter-note units.

To process the score, I create a window of a constant number of beats. As the window slides across the score, incrementing by beat, the program performs the DFT on the pitch content contained within that window.[6] This continues until the end of the score, creating a series of overlapping windows in much the same way that a camera takes multiple pictures and stitches them together to create a panoramic photo .

For some pieces of music, the process of sliding the window by a beat is computationally trivial because the meter is both constant and symmetrical. To slide the window forward by a beat, the beginning and end of the window offsets can simply be increased by the length of the beat. However, music with asymmetrical or changing meters pose more of a problem and require a much more flexible system. The solution was to use the music21 method `getContextByClass()` to retrieve the meter of every measure regardless of whether the `<time>` tag (used to indicate a meter signature) is present in the XML. Then, using the music21 object `MeterSequence` and its various partition methods, I create a list of the offsets that correspond with beat location which becomes the basis for incrementing the window. This helps to ensure that the portion of music being examined is always a meaningful unit.

The program has three possible strategies for how to count pitch classes in each window: *onset*, *duration*, and a *flat* set. The onset and duration options both return multisets. Onset counts how many times a pitch class is attacked within the window. If a note is initiated once and sustained, it will only be counted once, whereas if it is repeated, it is counted again for every repetition. The duration option returns the total duration of every pitch class within the window, even if a note was initiated before the window began. The flat set option counts each pitch class only once, regardless of how many times it appears on the musical surface. In essence it asks the binary question: “is this pitch class present?” These three options provide different ways of interpreting the pitch class data of the score, with each representing a different potential hearing of the music based on which parameters the listeners attend to.

Figure 3 shows Cipriano de Rore’s madrigal “Calami sonum ferentes” processed with each of these three approaches. The top two examples, using the onset and duration options, look fairly similar. This is a result of the voices tending to articulate each note just once as part of a melody rather than retaining the same pitch for multiple syllables. The f_5 component is quite jagged, indicating that while the diatonic collection is the primary background collection, chromaticism is suppressing its saliency. The graph showing the flat option confirms this, at times dropping to 0 in all components, which indicates that the entire chromatic aggregate is present in that window.

5 http://web.mit.edu/music21

6 The idea of combining the DFT with a sliding window was first proposed by Matthew Chiu at the 2019 Annual Conference of Music Theory Midwest. I am grateful for his correspondences on the topic.

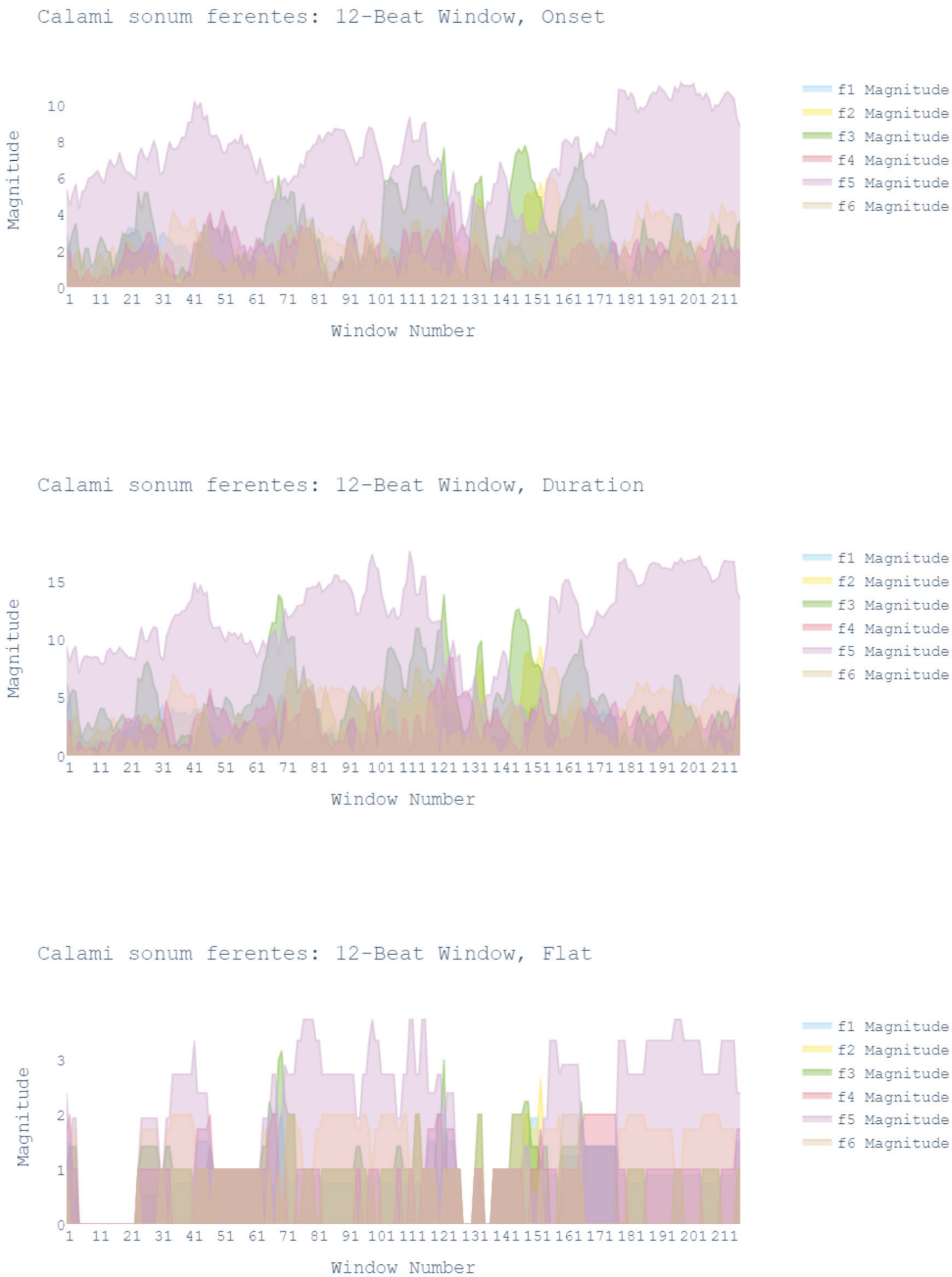


Figure 3: Three analyses of Cipriano de Rore's "Calami sonum ferentes."

Calculating the onsets is made complicated by the fact that in music21, the `Tie` class represents the visual and conceptual idea of tied notes. However, it retains the individual identity of the two different note objects rather than interpreting them as a single, long durational entity. Rather, music21 reads two individual notes. To compensate for this, the onset option relies on the music21 method `stripTies()`, which replaces all tied notes with a single note having a duration equal to all the tied constituents. This ensures that only one instance of the note is counted.

The duration option uses the music21 method `sliceByBeat()`, which splits a note at the beat offsets that music21 determines based upon the local time signature. Since the window increments every beat, this ensures that when the window boundary intersects a held note, the duration of the note within the window is accounted for and the portion outside the window is omitted. The flat option processes in the same way to avoid "missing" any notes whose durations began before the beginning of the window.

Once all the pitch-class arrays are collected, I weight the data with a logarithmic scale. The weighting procedure helps to shape the data to match our cognitive experience with the music. As the frequency of an individual pitch class increases, the information value of any given instance of that pitch class decreases. Because of its uniqueness, a single instance of a pitch class will have a relatively high influence. Musically speaking, this is equivalent to saying that the lone appearance of scale-degree sharp-4 will be more salient than repeated occurrences of scale-degree 1. The general profile of the array will stay the same—the higher frequency of scale-degree 1 will still have a greater effect than the single sharp-4, but the higher values will be flattened out.

Finally, I apply the DFT to each of the arrays and store the data in Pandas (a Python library used for data manipulation and analysis) data frames.[7] These tables are then used for queries and generating visualizations of the data.

Analysis

The graph in Figure 4 shows the magnitudes of all six Fourier components in the exposition to the first movement of Mozart's String Quartet No. 4 in C Major, K. 157, generated with a 16-beat sliding window using the duration valuation of pitch classes. The mass of lavender represents the f_5 component, and its prevalence indicates that perfect fifths or *diatonicity* (i.e., sticking to only the notes that are native to a particular key signature) is the most salient feature of the harmonic landscape.

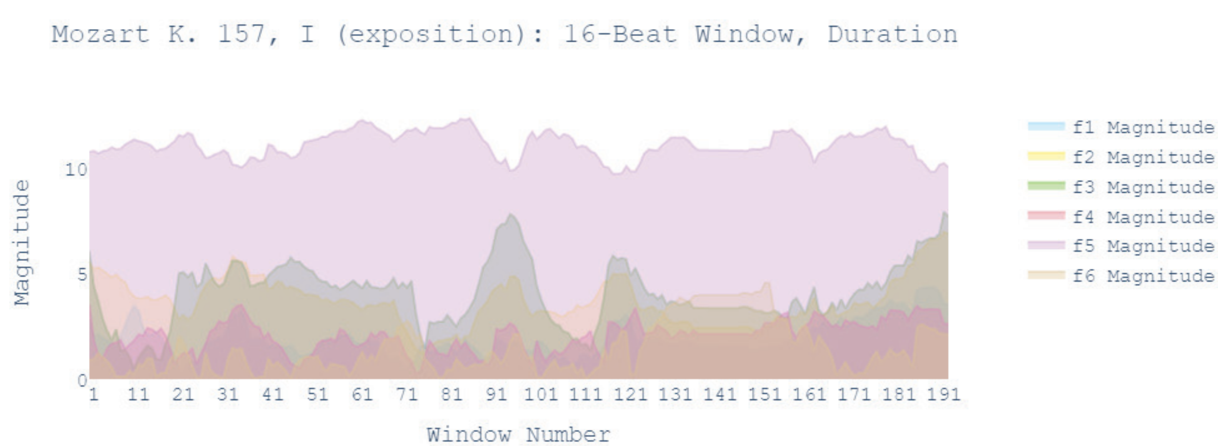


Figure 4: Magnitudes of the six Fourier components in the exposition of Mozart K. 157, I.

Figure 5 extracts the f_5 magnitude from Figure 4 and overlays both the raw phase data and the quantized phase. Since the phase of the f_5 component correlates with implied key signature, we can use this information to infer that the piece begins with 0 sharps or flats (indicated by the 60° position) and moves to 1 sharp (indi-

7 https://pandas.pydata.org

cated by the 90° position sometime around the 75th window and generally remains there for the rest of the exposition. In other words, we see evidence of the modulation to the dominant.

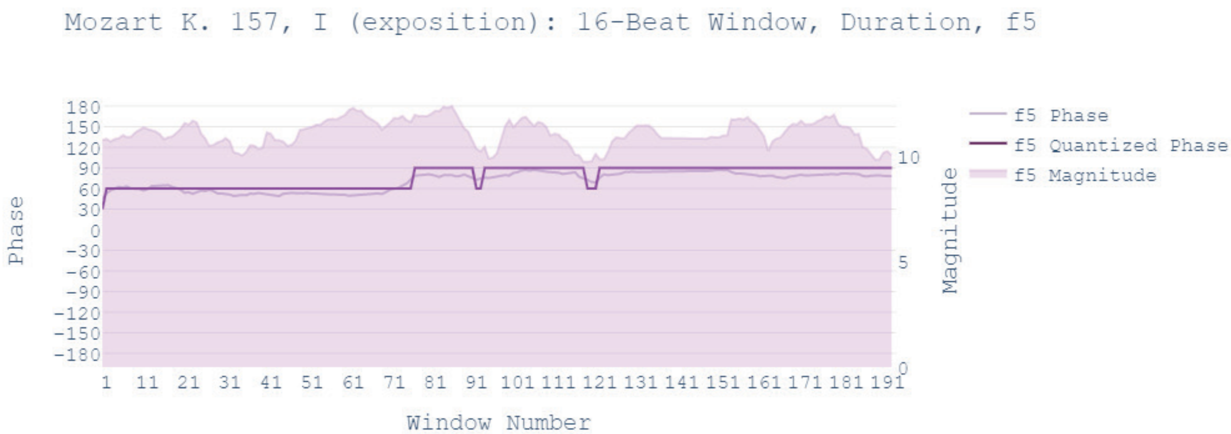


Figure 5: Magnitudes and phase of the f_5 component in the exposition of Mozart K. 157, I.

The theme from Messiaen's *Theme and Variations for Violin and Piano* adheres strictly to his second (octatonic) and third modes of limited transposition, a very different harmonic context than Mozart's. Figure 6 shows the magnitudes of the six Fourier components generated with a 16-beat sliding window using the onset valuation of pitch classes. The three-part construction of the theme is apparent based on which Fourier components are the most salient (have the highest magnitude). The first section is in *AA'* form, with the triple peaks of the f_6 component (in windows 1-11 and again around 31-41) showing the similarity of the beginnings of the two *A* repetitions most clearly.

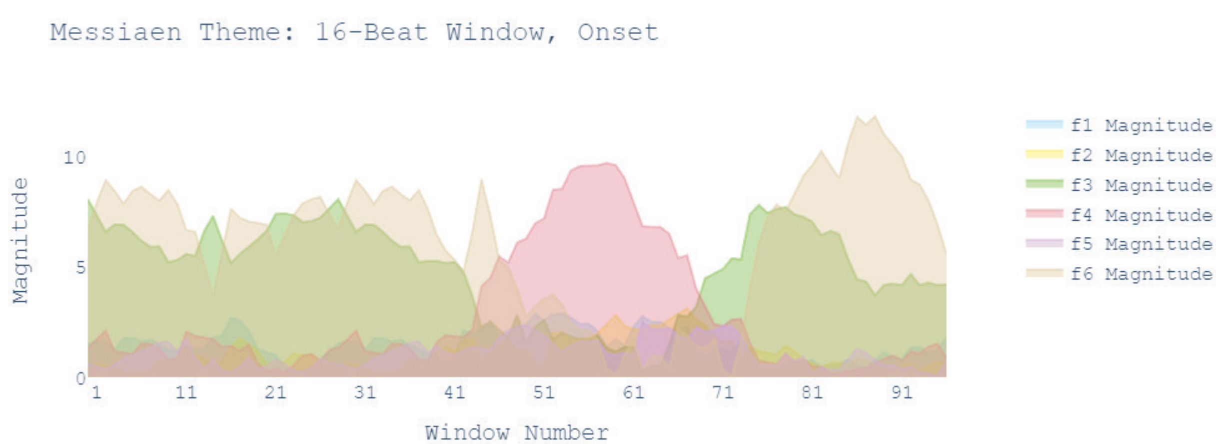


Figure 6: Magnitudes of the six Fourier components in the Theme from Messiaen's Theme and Variations for Violin and Piano.

The magnitude of the f_4 component is extracted in Figure 7 and shown along with the phase and quantized phase. The high magnitudes in the f_4 component and the phase at -60° indicate the presence of the Octatonic 01 collection (Messiaen's second mode of limited transposition), in much the same way that the position of phase can be used to determine the implied key signature when the f_5 component reaches a high magnitude.

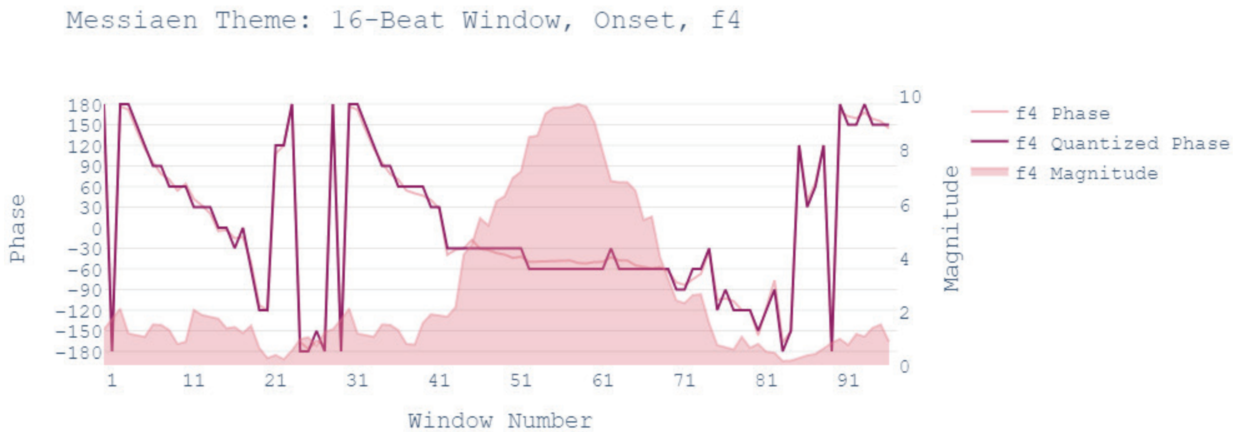


Figure 7: Magnitude and phase of the f_4 component in Messiaen's Theme.

Messiaen's third mode of limited transposition is a bit more subtle to tease out. Mode 3 can be thought of as a scale built of repeated <half-half-whole> step patterns or, more productively for our use, as a combination of three of the four augmented triads (interval-class 4-cycles, represented by f_3). This further means that it includes one of the two whole-tone scales (2-cycles, represented by f_6) and half of the other. Looking at the first and third sections of the piece (Figure 6), Fourier components f_3 and f_6 are the most salient. While the magnitude is high, the phase of f_6 (shown in Figure 8) stays at 0°, the angle indicating the even whole-tone collection. However, if this were just a whole-tone collection, the vectors in the f_3 component would cancel each other out (Figure 10), effectively removing f_3 from the graph, and the magnitude of f_6 would be even higher than it is. Clearly this is not the case as the magnitude for the f_3 component is relatively high, close to that of f_6. The f_3 phase hovers around 90° (Figure 9), the location of three pitch classes (3, 7, and 11) that are not included in the even whole-tone collection as shown in Figure 10. The presence of these three pitch classes dramatically bolsters the f_3 component, but has the opposite effect on the f_6 component, substantially lowering its magnitude.

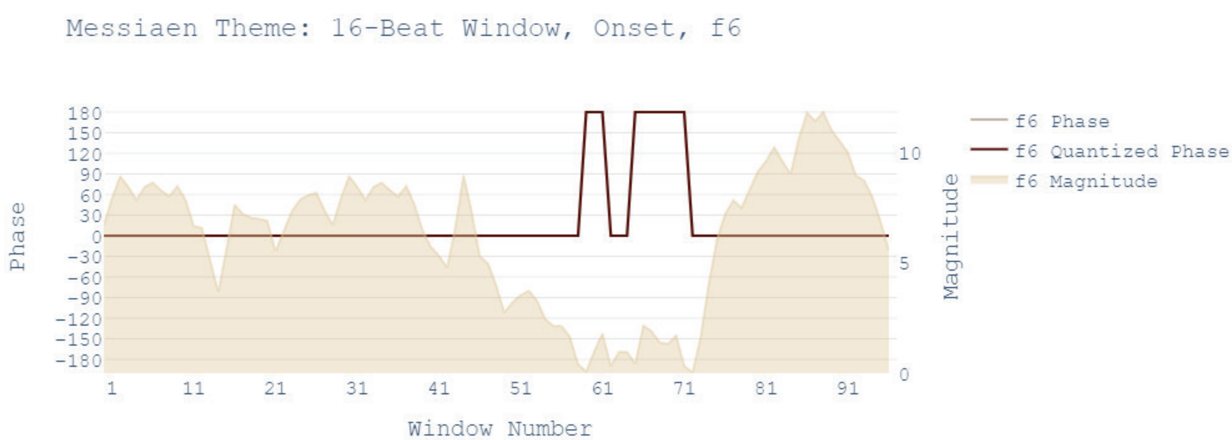


Figure 8: Magnitude and phase of the f_6 component in Messiaen's Theme.

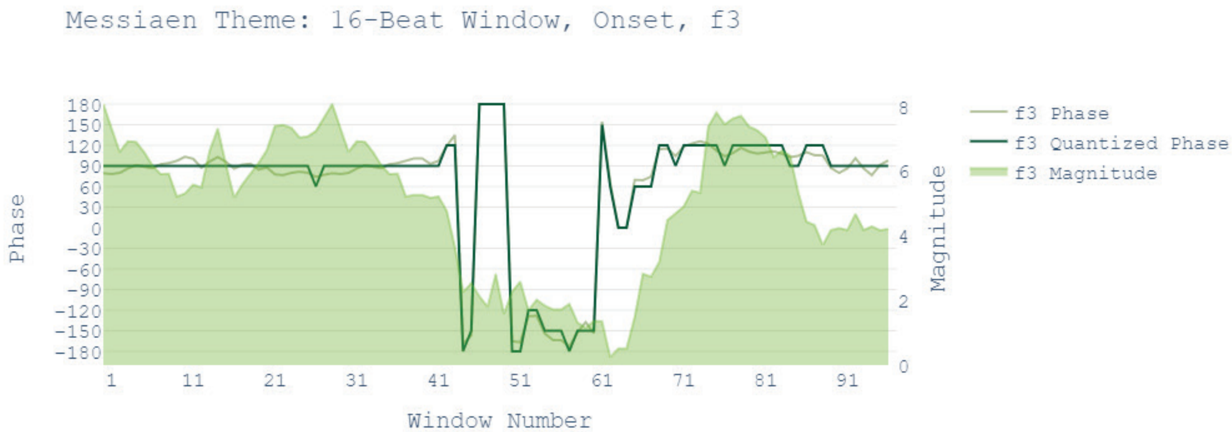


Figure 9: Magnitude and phase of the f_3 component in Messiaen's Theme.

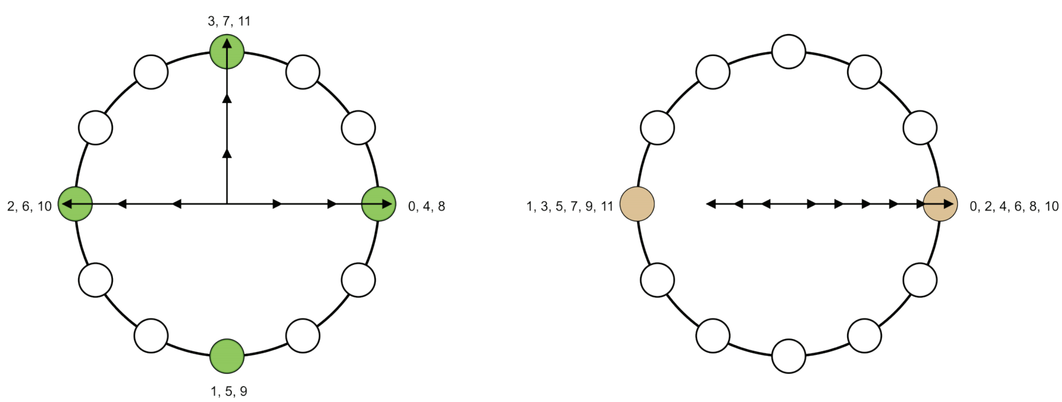


Figure 10: Third mode of limited transposition mapped onto the f_3 and f_6 component circles.

Discussion and conclusion

Examining macroharmonies through the DFT clearly displays a broad overview of a piece's sonic landscape, and it does so in a completely quantifiable and mathematically precise way. These two brief analyses offer a limited view into this methodology and its capabilities; many other variants exist. The overview can be as blurred or as granular as the analyst wishes by adjusting the size of the sliding window. Music based in other collections that equally divide the octave is evaluated with as much ease as our 12-note chromatic world, opening the door to examine other understudied repertoires simply by changing the length of the input array. Different modes of listening can be modeled by using the different pitch class evaluation strategies (onset, duration, and flat set).

The DFT can also function as a key-finding algorithm—a long-studied topic in both music cognition and computation. It is typically not enough to examine just the f_5 component to determine a key, as in the simple[MB1] Mozart example here. Instead, components f_2 and f_3 also play strong roles in tonal music, fluctuating based on sounding harmony while f_5 remains stable [8]. The DFT provides a mathematically robust way to model harmonic and tonal activity by tracking motion through $f_{2/3/5}$ space [1].

Perhaps most significantly, this is a single methodology that is equally applicable to music from a wide variety of genres, time periods, and styles. As long as the music can be encoded with discrete pitches and rhythms, this approach will have something to say about it. Many music-analytical techniques are bound to a particular repertoire. The DFT is stylistically and historically agnostic, allowing for a more unified understanding of music writ large.

Works cited

[1] Yust, Jason. “Harmonic Qualities in Debussy’s ‘Les sons et les parfums tournent dans l’air du soir’” *Journal of Mathematics and Music* 11, nos. 2-3 (2017), 155-73.

[2] Tymoczko, Dmitri. “Set-Class Similarity, Voice Leading, and the Fourier Transform.” Journal of Music Theory 52, no. 2 (2008), 251-72.

[3] Amiot, Emmanuel. *Music through Fourier Space: Discrete Fourier Transform in Music Theory*. Cham, Switzerland: Springer International Publishing, 2016.

[4] Chiu, Matthew. “A systematic approach to macroharmonic progressions: Durufle’s Requiem through Fourier Space” presented at the Thirtieth Annual Conference of Music Theory Midwest, Cincinnati, OH, May 10-11, 2019.

[5] Tymoczko, Dmitri. *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice.* Oxford: Oxford University Press, 2011.

[6] Quinn, Ian. “General Equal-tempered Harmony: Parts 2 and 3” Perspectives of New Music 44, no. 2 (2006), 114-58.

[7] Cuthbert, Michael Scott, and Christopher Ariza. “Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data” in *Proceedings of the International Society for Music Information Retrieval* (ISMIR 2010), 637-42.

[8] Yust, Jason. “Probing Questions about Keys: Tonal Distributions through the DFT” in *Mathematics and Computation in Music, Proceedings of the Sixth International Conference* (MCM 2017). Vol. 10527 of *Lecture Notes in Artificial Intelligence*, eds. Octavio A Agustin-Aquino, Emilio Lluis-Puebla, and Mariana Montiel. Cham, Switzerland: Springer International Publishing, 167-79.

Preventing Conversion Failure across Encoding Formats: A Transcription Protocol and Representation Scheme Considerations

Emilia Parada-Cabaleiro
Universidad Complutense de Madrid (Spain)
eparada@iccmu.es

Álvaro Torrente
Universidad Complutense de Madrid (Spain)
atorrente@iccmu.es

Abstract

Conversion issues across musical symbolic representations, such as musicXML, MEI, and humdrum, are well known. Often, these depend on methodological choices undertaken during the generation and processing of the data. For a better under-standing of this topic, we present a transcription protocol, result of trial and error transcription attempts performed with Finale engraving software, which aims to prevent conversion errors (Verovio 2.1.0 and VHV were taken into account for conversion) from musicXML (export format from Finale) to MEI and **kern (symbolic representations also evaluated).

Introduction

Original written sources, both manuscripts and prints, represent an enormous part of western music heritage; these, since only the original remains for some repertoires, possess an invaluable historical value. To encourage preservation, analysis, and performance, the transcription of these sources is of great interest in musicology, whose goal is often the creation of diplomatic, critical, and practical editions [1]. Despite these sources have raised great interest also in Music Information Retrieval (MIR), and although the understanding of this repertoire would incredibly benefit from a Musicology–MIR collaborative approach, the connection between these two disciplines is limited by the application of generally accepted practice. Musicologists' strong predisposition towards specific notation software [2 , pp. 43–66] and the lack of maintenance of MIR tools no longer sponsored by the creator's institutions [3], impair, e.g., an effective translation across encoding formats [4, 5]. We present a case study based on the 18th century Italian opera—handwritten versions of Demofoonte by different composers were considered—which analyses Musicology–MIR methodological incompatibilities by evaluating the transcription techniques traditionally used in musicology, as well as examining two typical symbolic data representations: MEI [6], rendered with Verovio;[1] and the Humdrum representation scheme **kern [7], engraved through Verovio Humdrum Viewer (VHV).[2]

Transcription protocol to guarantee musicXML quality

Engraving software configuration

1. To prevent data loss and 'artefacts' generation, metadata, as e.g., composer, should be indicated in the score manager.
2. To reduce misinterpretation risks, all the parameters defined in the engraving software, such as dynamics or repeat marks, should be introduced, when possible, prioritising the software's default options, rather than using text indications.

1 https://www.verovio.org
2 https://verovio.humdrum.org

Transcription of 18th century hand-written sources

3. Missing repeat expressions, such as Fine, are typical in 18th arias, e.g., it might be pointed out Da Capo al Fine without having previously indicated Fine. In order to avoid ambiguities which would impair computational processes, e.g., the function repeat.expander in music21 [8], partial instructions should be completed.
4. Incomplete measures are typically associated in the presented repertoire to repeat expressions. To prevent duration discrepancies [9], these should be corrected in the transcription, by indicating repeat brackets if needed. For divisi (voices written in different layers of a staff) and redundant information, such as duplicated rests, should be hidden instead of omitted.
5. Mid-note dynamics, i.e., dynamics that start at the middle of a long note, are also typical editorial choices [10]. To avoid that mid-note dynamics are automatically associated to the consecutive note in **kern, these should be linked to the note's attack and then manually shifted in the engraving software. Although this does not codify the mid-note dynamic in its exact position, guarantees that it is not connected to the wrong note.
6. Mixed dynamics, i.e., dynamics made up of by the combination between standard letters (e.g., pp) and text expressions (e.g., poco), contrary to what expected, should not be introduced in the engraving software as new dynamic marks. In order to prevent information loss, each element (letter and text) should be individually associated to the same note.
7. Similar to unclosed ties [9], the confusion between ties and slurs is also an unexpected behaviour that would lead to ambiguous interpretation; this engraving mistake should be carefully prevented.

MEI and HUMDRUM conversion and rendering issues

1. The Segno ( ), i.e., the symbol used as a 'navigation marker' after an Al Segno is found, is lost in the conversion from musicXML to MEI and **kern. Similarly, the repeat expressions Al Segno and Da Capo are also lost in **kern.
2. Repeat brackets, i.e., numbered brackets to indicate a different ending for the first play and its repetition, are lost in the conversion from musicXML to **kern. Furthermore, the use of the repeat option 'go to measure' in the engraving software introduces, in both MEI and **kern, 'artefacts', since interpreted as a textual indication.
3. In MEI, dynamics' positions are expressively defined by the attribute time stamp (tstamp), which gives the specific horizontal alignment between the expression and the measure; this enables the codification of mid-note dynamics. Differently, in **kern, dynamics are automatically linked to the attack of the note to which they belong to; this impairs the assignment of mid-note dynamics, which to be correctly displayed should be indicated later on in the **dynam spine (cf. Figure 1).

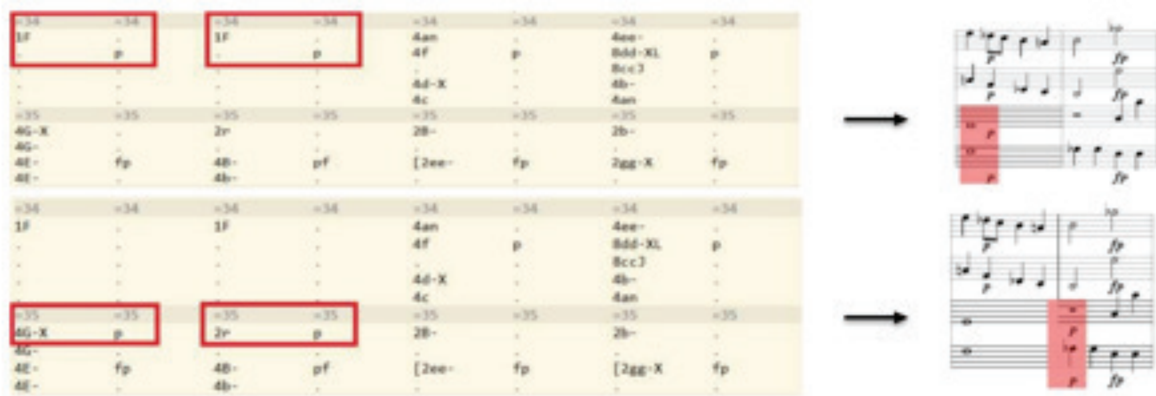


Figure 1: At the left, above: mid-note dynamic p wrongly linked in **kern to the attack of the consecutive note, i.e., the first note of the next measure (number 35); below: correct codification of the mid-note dynamic to the whole note in measure 34. The musical samples engraved with VHV are displayed at the right, above: wrongly codified; below: correct. This example was taken from the aria Misero Pargoletto, composed by Leonardo Leo in 1735.

4. Splitting mixed dynamics prevents their loss during conversion; cf. (iv) in the section 'Transcription Protocol'. Yet, to enable a correct rendering, the individual elements should be reassembled again after conversion. Although this guarantees a correct rendering in MEI, mixed dynamics in **kern (e.g., molto f or pf) might not be engraved.
5. In the translation to **kern, despite no conversion failure, tremolos are totally lost.

Conclusion

Through these guidelines we aim to minimise incompatibilities between Musicology and MIR, linked to transcription methods typical of music editions. Furthermore, to encourage Musicology–MIR collaborative approaches, we also pointed out encoding limitations that could be addressed in future. Although some elements were lost during conversion, MEI and **kern syntax present also advantages w.r.t. musicXML, as, e.g., the possibility to codify mixed dynamics as unique instances.

Acknowledgements

This work is a result of the Didone Project, which has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, Grant agreement No. 788986

Works Cited

[1] Broude, Ronald. "Musical works, musical texts, and musical editions" Scholarly Editing: The Annual of the Association for Documentary Editing 33 (2012), 1–29. http://www.scholarlyediting.org/2012/essays/essay.broude.html

[2] Wiering, Frans. "Digital critical editions of music: A multidimensional model" in Modern Methods for Musicology, eds. Tim Crawford and Lorna Gibson. London, UK: Routledge, 2016.

[3] Gardiner, Eileen, and Ronald G. Musto. The digital humanities: A primer for students and scholars. Cambridge, UK: Cambridge University Press, 2015.

[4] Nápoles López, Néstor, Gabriel Vigliensoni, and Ichiro Fujinaga. "The effects of translation between symbolic music formats: A case study with Humdrum, Lilypond, MEI, and MusicXML" presented at the Music Encoding Conference, Vienna, Austria, May 29-June 1, 2019.

[5] Parada-Cabaleiro, Emilia, Anton Batliner, and Björn Schuller. "A diplomatic edition of il Lauro Secco: Ground truth for OMR of white mensural notation" in Procceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR 2019), 557–64.

[6] Roland, Perry, Andrew Hankinson, and Laurent Pugin. "Early music and the music encoding initiative" Early Music 42 (2014), 605–11.

[7] Huron, David. "Music information processing using the Humdrum Toolkit: Concepts, examples, and lessons" Computer Music Journal 26 (2002), 11–26.

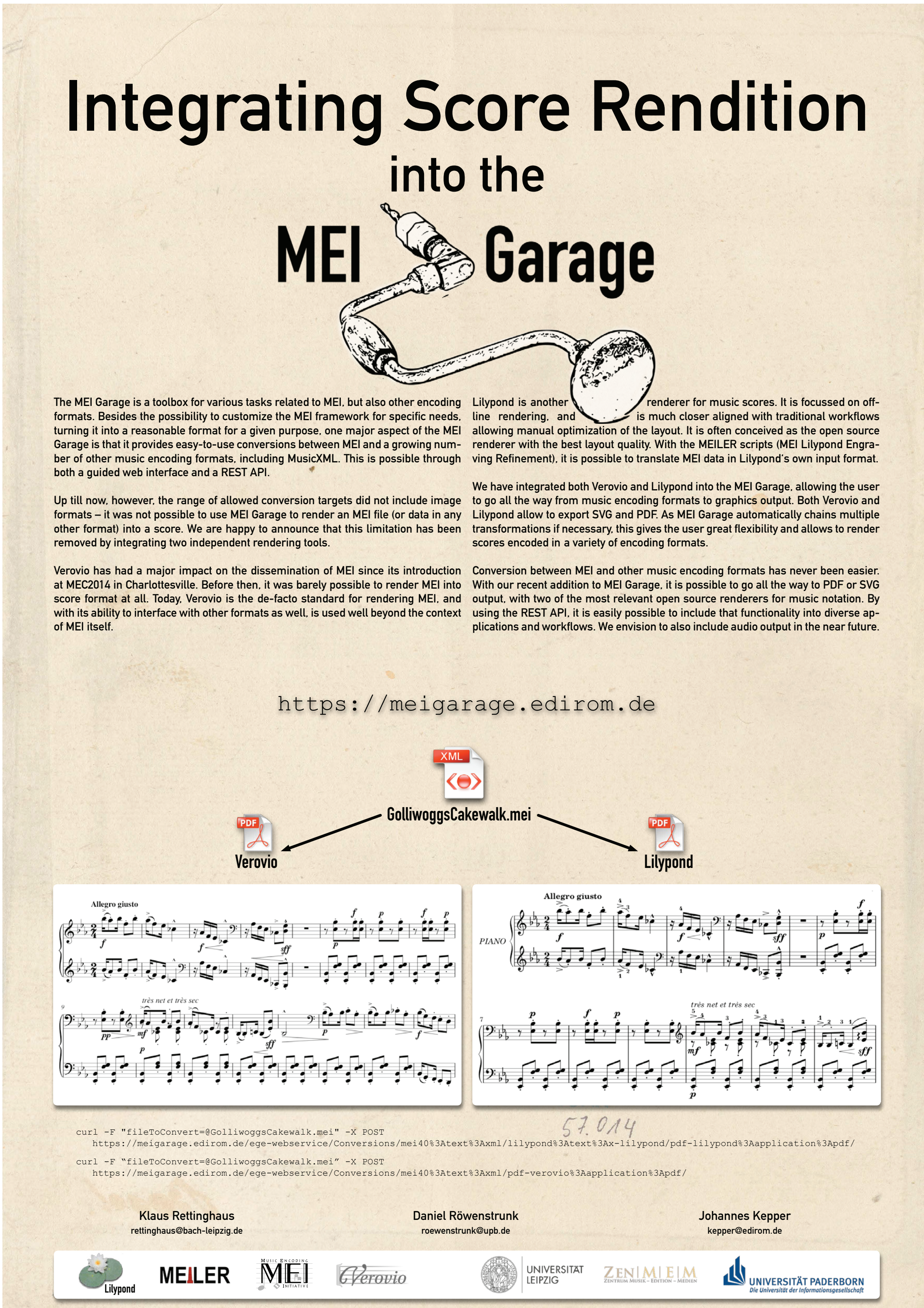
[8] Cuthbert, Michael Scott, and Christopher Ariza. "music21: A toolkit for computer-aided musicology and symbolic music data" in Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010), 637–42.

[9] Nápoles López, Néstor, Gabriel Vigliensoni, and Ichiro Fujinaga. “Encoding matters” in Proceedings of the 5th International Conference on Digital Libraries for Musicology (DLfM ’18), 69–73. https://doi.org/10.1145/3273024.3273027.

[10] Günter, Raphael. “Ludwig van Beethoven Klaviertrios”. Munich, Germany: G. Henle Verlag, 1992.

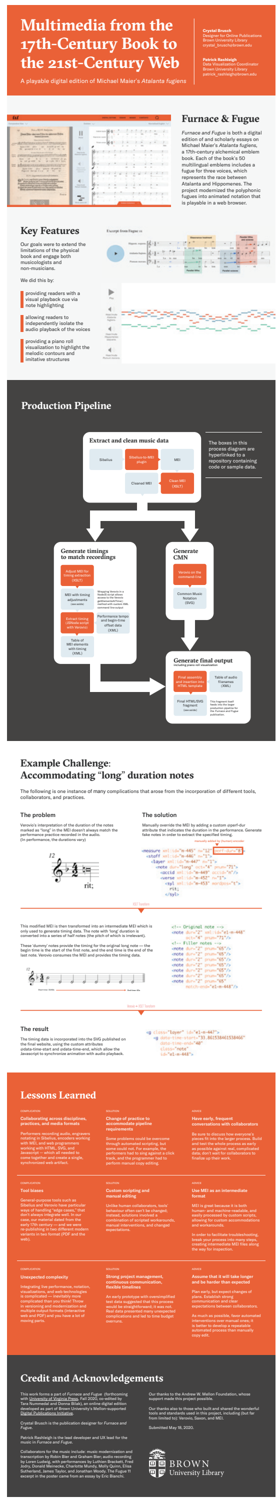
Integrating Score Rendition in the MEI Garage

Klaus Rettinghaus, Daniel Röwenstrunk, Johannes Kepper



Multimedia from the 17th-Century Book to the 21st-Century Web – A Playable Digital Edition of Michael Maier's "Atalanta fugiens"

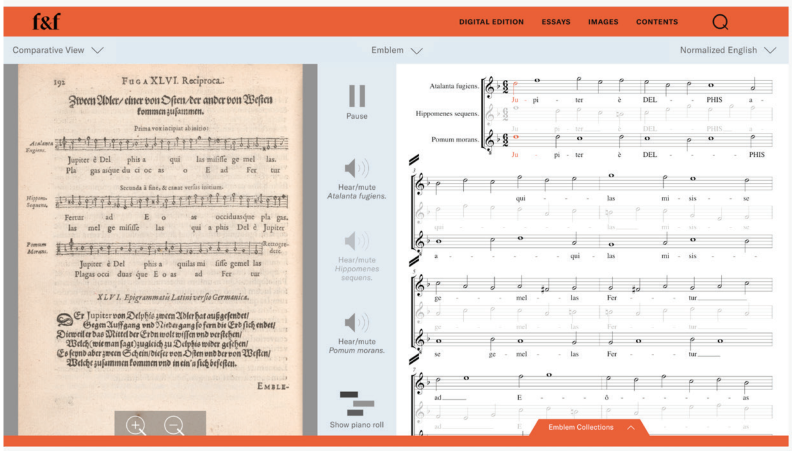
Patrick Rashleigh, Crystal Brusch



Multimedia from the 17th-Century Book to the 21st-Century Web

A playable digital edition of Michael Maier's *Atalanta fugiens*

Crystal Brusch
Designer for Online Publications
Brown University Library
crystal_brusch@brown.edu

Patrick Rashleigh
Data Visualization Coordinator
Brown University Library
patrick_rashleigh@brown.edu

Furnace & Fugue

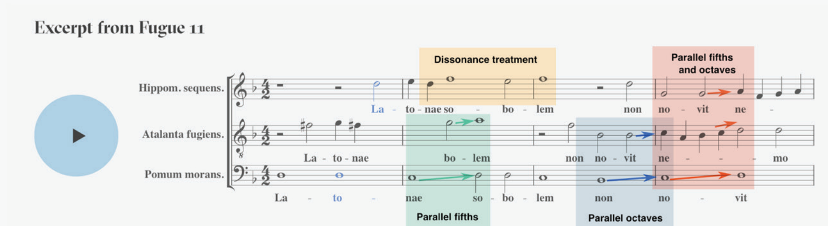
Furnace and Fugue is both a digital edition of and scholarly essays on Michael Maier's *Atalanta fugiens*, a 17th-century alchemical emblem book. Each of the book's 50 multilingual emblems includes a fugue for three voices, which represents the race between Atalanta and Hippomenes. The project modernized the polyphonic fugues into animated notation that is playable in a web browser.

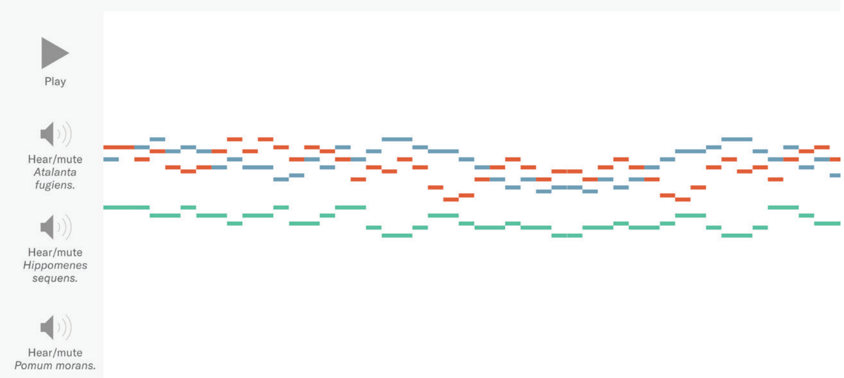
Key Features

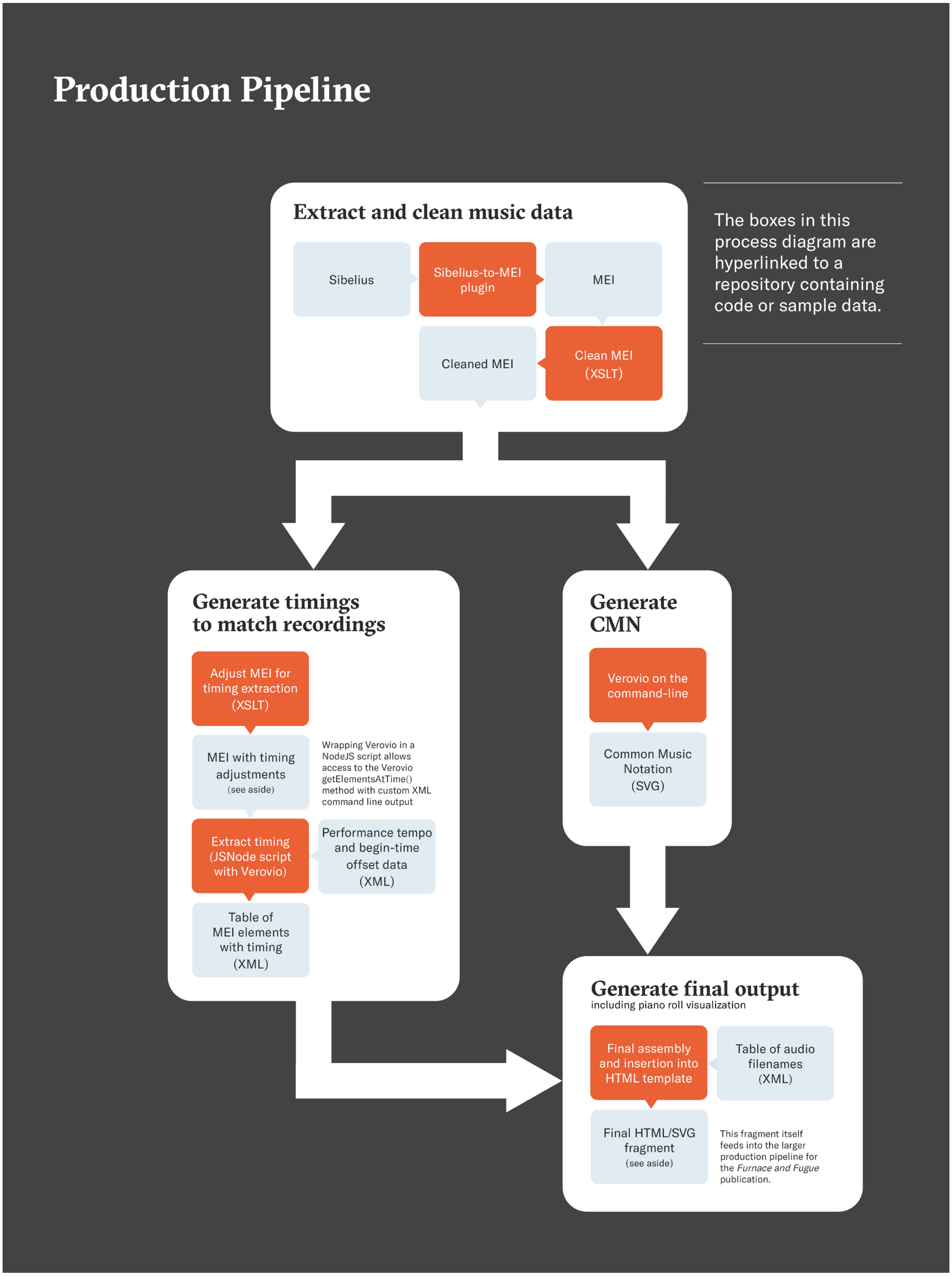
Our goals were to extend the limitations of the physical book and engage both musicologists and non-musicians.

We did this by:

- providing readers with a visual playback cue via note highlighting
- allowing readers to independently isolate the audio playback of the voices
- providing a piano roll visualization to highlight the melodic contours and imitative structures



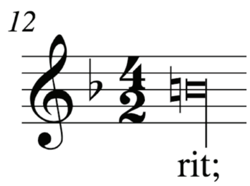



Production Pipeline
Extract and clean music data
Sibelius
Sibelius-to-MEI plugin
MEI
Cleaned MEI
Clean MEI (XSLT)
The boxes in this process diagram are hyperlinked to a repository containing code or sample data.
Generate timings to match recordings
Adjust MEI for timing extraction (XSLT)
MEI with timing adjustments (see aside)
Wrapping Verovio in a NodeJS script allows access to the Verovio getElementsAtTime() method with custom XML command line output
Extract timing (JSNode script with Verovio)
Performance tempo and begin-time offset data (XML)
Table of MEI elements with timing (XML)
Generate CMN
Verovio on the command-line
Common Music Notation (SVG)
Generate final output
including piano roll visualization
Final assembly and insertion into HTML template
Table of audio filenames (XML)
Final HTML/SVG fragment (see aside)
This fragment itself feeds into the larger production pipeline for the *Furnace and Fugue* publication.

Example Challenge: Accommodating "long" duration notes

The following is one instance of many complications that arose from the incorporation of different tools, collaborators, and practices.

The problem

Verovio's interpretation of the duration of the notes marked as "long" in the MEI doesn't always match the performance practice recorded in the audio.
(In performance, the durations vary)

The solution

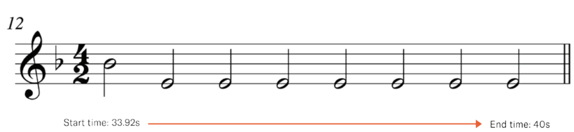
Manually override the MEI by adding a custom @perf-dur attribute that indicates the duration in the performance. Generate fake notes in order to extract the specified timing.

manually added by (human) encoder

```
<measure xml:id="m-445" n="12" perf-dur="8">
  <staff xml:id="m-446" n="1">
    <layer xml:id="m-447" n="1">
      <note dur="long" oct="4" pnum="71">
        <accid xml:id="m-449" accid="n"/>
        <verse xml:id="m-452" n="1">
          <syl xml:id="m-453" wordpos="t">
            rit;
          </syl>
```

XSLT Transform

This modified MEI is then transformed into an intermediate MEI which is only used to generate timing data. The note with 'long' duration is converted into a series of half notes (the pitch of which is irrelevant).

These 'dummy' notes provide the timing for the original long note — the begin time is the start of the first note, and the end time is the end of the last note. Verovio consumes the MEI and provides the timing data.


```
<!-- Original note -->
<note dur="2" xml:id="e1-m-448"
      oct="4" pnum="71"/>
<!-- Filler notes -->
<note dur="2" pnum="65"/>
<note dur="2" pnum="65"/>
<note dur="2" pnum="65"/>
<note dur="2" pnum="65"/>
<note dur="2" pnum="65"/>
<note dur="2" pnum="65"/>
<note dur="2" pnum="65"
      match-end="e1-m-448"/>
```

Verovio + XSLT Transform

The result

The timing data is incorporated into the SVG published on the final website, using the custom attributes @data-time-start and @data-time-end, which allow the Javascript to synchronize animation with audio playback.

```
<g class="layer" id="e1-m-447">
   <g data-time-start="33.861538461538466"
      data-time-end="40"
      class="note"
      id="e1-m-448">
```

Lessons Learned

COMPLICATION

Collaborating across disciplines, practices, and media formats

Performers recording audio, engravers notating in Sibelius, encoders working with MEI, and web programmers working with HTML, SVG, and Javascript -- which all needed to come together and create a single, synchronized web artifact.

SOLUTION

Change of practice to accommodate pipeline requirements

Some problems could be overcome through automated scripting, but some could not. For example, the performers had to sing against a click track, and the programmer had to perform manual copy editing.

ADVICE

Have early, frequent conversations with collaborators

Be sure to discuss how everyone's pieces fit into the larger process. Build and test the whole process as early as possible against real, complicated data; don't wait for collaborators to finalize up their work.

COMPLICATION

Tool biases

General-purpose tools such as Sibelius and Verovio have particular ways of handling "edge cases," that don't always integrate well. In our case, our material dated from the early 17th century — and we were re-publishing in two different modern variants in two format (PDF and the web).

SOLUTION

Custom scripting and manual editing

Unlike human collaborators, tools' behaviour often can't be changed; instead, solutions involved a combination of scripted workarounds, manual interventions, and changed expectations.

ADVICE

Use MEI as an intermediate format

MEI is great because it is both human- and machine-readable, and easily processed by custom scripts, allowing for custom accommodations and workarounds.

In order to facilitate troubleshooting, break your process into many steps, creating intermediate MEI files along the way for inspection.

COMPLICATION

Unexpected complexity

Integrating live performance, notation, visualizations, and web technologies is complicated — inevitably more complicated than you think! Throw in versioning and modernization and multiple output formats (interactive web and PDF) and you have a lot of moving parts.

SOLUTION

Strong project management, continuous communication, flexible timelines

An early prototype with oversimplified test data suggested that this process would be straightforward; it was not. Real data presented many unexpected complications and led to time budget overruns.

ADVICE

Assume that it will take longer and be harder than expected

Plan early, but expect changes of plans. Establish strong communication and clear expectations between collaborators.

As much as possible, favor automated interventions over manual ones; it is better to develop a repeatable automated process than manually copy edit.

Credit and Acknowledgements

This work forms a part of *Furnace and Fugue* (forthcoming with University of Virginia Press, Fall 2020, co-edited by Tara Nummedal and Donna Bilak), an online digital edition developed as part of Brown University's Mellon-supported Digital Publications Initiative.

Crystal Brusch is the publication designer for *Furnace and Fugue*.

Patrick Rashleigh is the lead developer and UX lead for the music in *Furnace and Fugue*.

Collaborators for the music include: music modernization and transcription by Robin Bier and Graham Bier; audio recording by Loren Ludwig, with performances by Luthien Brackett, Fred Jodry, Donald Meinecke, Charlotte Mundy, Molly Quinn, Elisa Sutherland, James Taylor, and Jonathan Woody. The Fugue 11 excerpt in the poster came from an essay by Eric Bianchi.

Our thanks to the Andrew W. Mellon Foundation, whose support made this project possible.

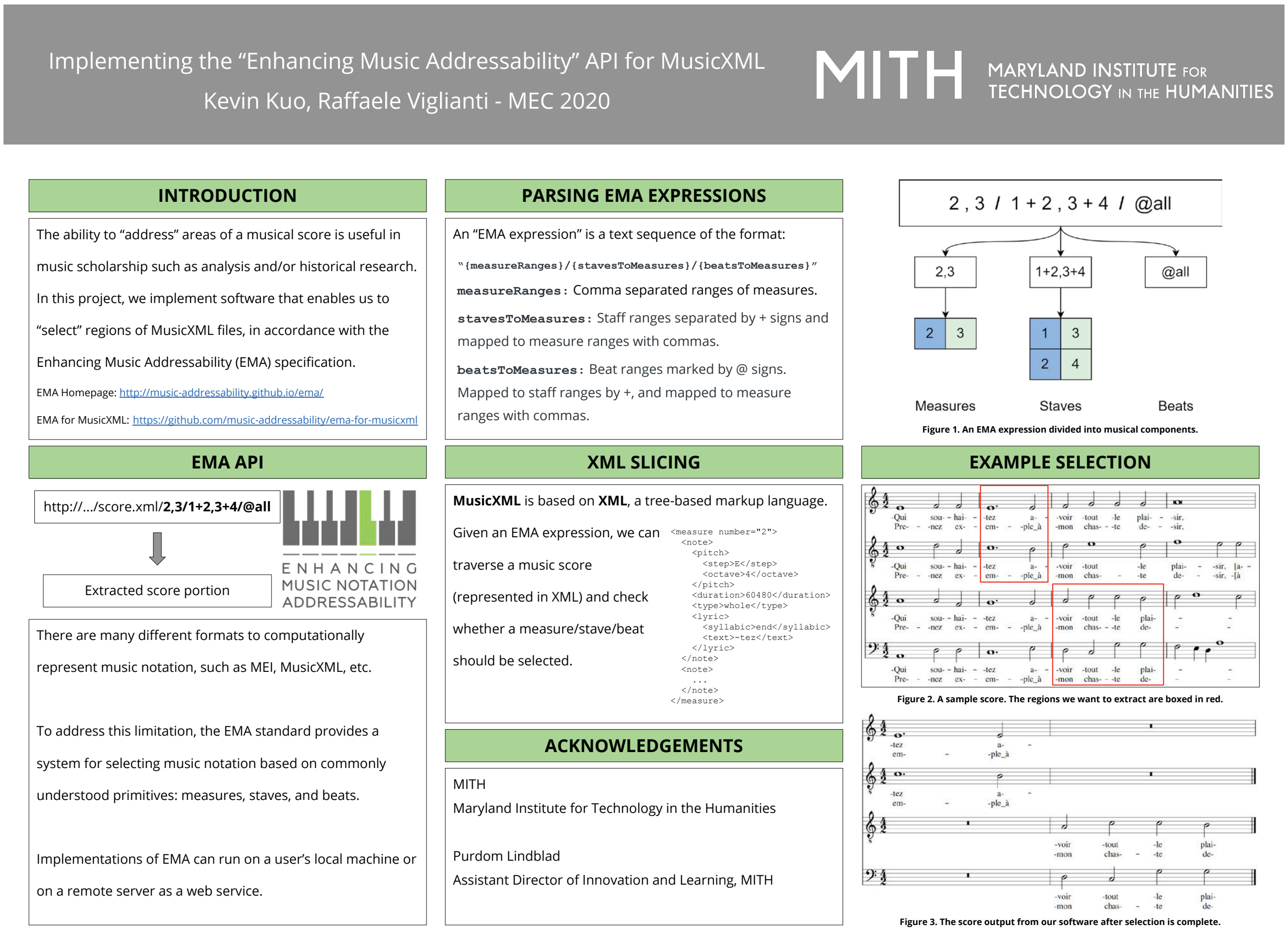
Our thanks also to those who built and shared the wonderful tools and standards used in this project, including (but far from limited to): Verovio, Saxon, and MEI.

Submitted May 18, 2020.

BROWN
University Library

Implementing the Enhancing Music Addressability API for MusicXML

Kevin Kuo, Raffaele Viglianti



INTRODUCTION

The ability to "address" areas of a musical score is useful in music scholarship such as analysis and/or historical research. In this project, we implement software that enables us to "select" regions of MusicXML files, in accordance with the Enhancing Music Addressability (EMA) specification.

EMA Homepage: http://music-addressability.github.io/ema/

EMA for MusicXML: https://github.com/music-addressability/ema-for-musicxml

EMA API

http://.../score.xml/**2,3/1+2,3+4/@all**

↓

Extracted score portion

ENHANCING
MUSIC NOTATION
ADDRESSABILITY

There are many different formats to computationally represent music notation, such as MEI, MusicXML, etc.

To address this limitation, the EMA standard provides a system for selecting music notation based on commonly understood primitives: measures, staves, and beats.

Implementations of EMA can run on a user's local machine or on a remote server as a web service.

PARSING EMA EXPRESSIONS

An “EMA expression” is a text sequence of the format:

“`{measureRanges}/{stavesToMeasures}/{beatsToMeasures}`”

`measureRanges`: Comma separated ranges of measures.

`stavesToMeasures`: Staff ranges separated by + signs and mapped to measure ranges with commas.

`beatsToMeasures`: Beat ranges marked by @ signs. Mapped to staff ranges by +, and mapped to measure ranges with commas.

XML SLICING

MusicXML is based on **XML**, a tree-based markup language.

Given an EMA expression, we can traverse a music score (represented in XML) and check whether a measure/stave/beat should be selected.

```
<measure number="2">
  <note>
    <pitch>
      <step>E</step>
      <octave>4</octave>
    </pitch>
    <duration>60480</duration>
    <type>whole</type>
    <lyric>
      <syllabic>end</syllabic>
      <text>-tez</text>
    </lyric>
  </note>
  <note>
    ...
  </note>
</measure>
```

ACKNOWLEDGEMENTS

MITH

Maryland Institute for Technology in the Humanities

Purdom Lindblad

Assistant Director of Innovation and Learning, MITH

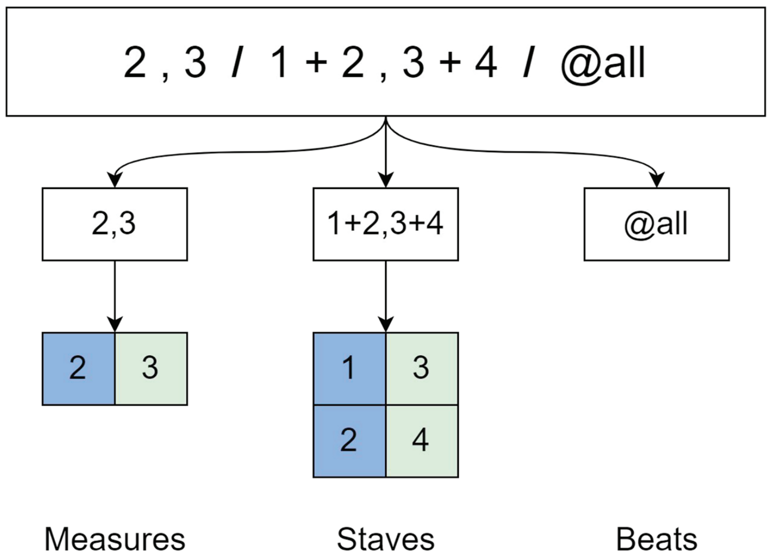


Figure 1. An EMA expression divided into musical components.

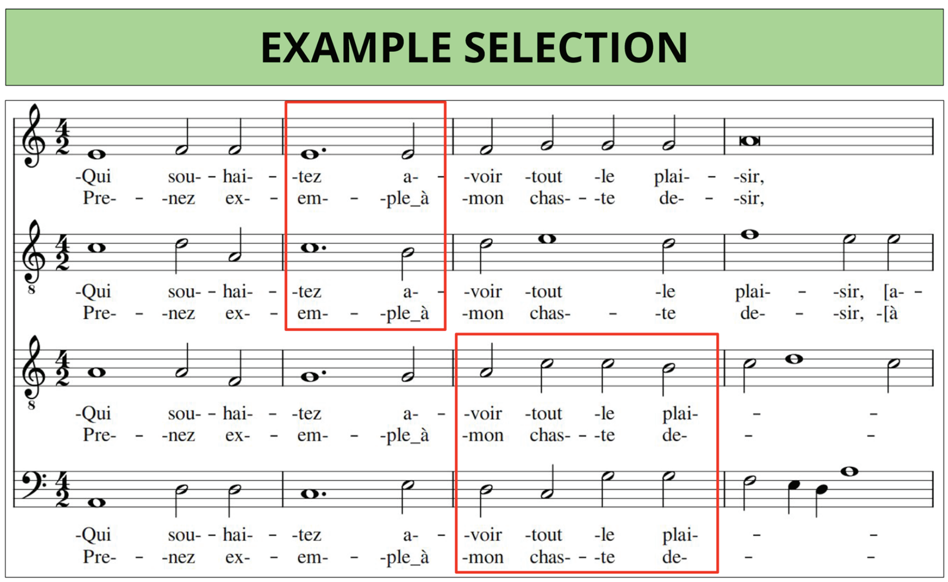


Figure 2. A sample score. The regions we want to extract are boxed in red.

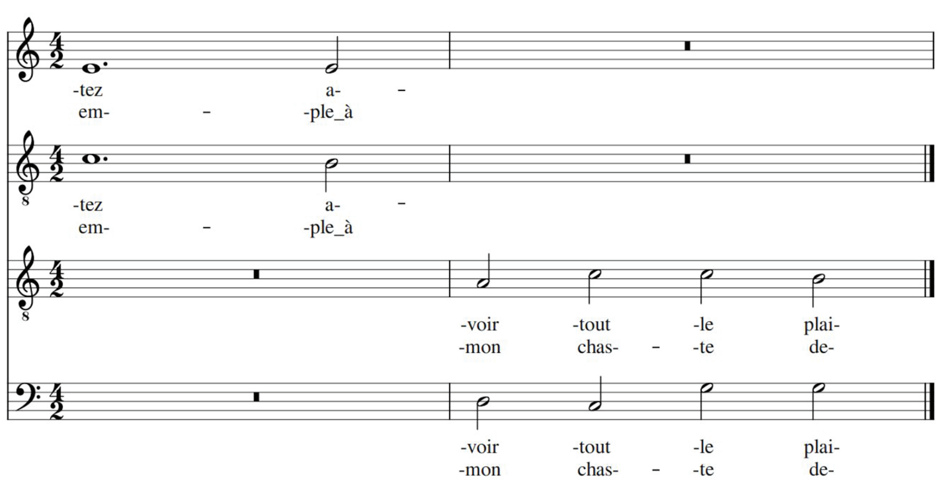


Figure 3. The score output from our software after selection is complete.

Next Steps for Measuring Polyphony – A Prototype Editor for Encoding Mensural Music

Karen Desmond, Andrew Hankinson, Laurent Pugin, Juliette Regimbal, Craig Sapp, Martha E. Thomae

Project background

The thirteenth and fourteenth centuries saw an unprecedented increase in the production of manuscripts transmitting music repertoires with a new diversity of styles, genres, and subject matter, copied in both music-only anthologies, and in miscellaneous collections that interweave song, text, and illuminations. At the same time, techniques for specifically notating rhythmic duration emerged, a notation called "mensural" or "measurable." Almost all polyphonic music (music composed for two or more parts) from 1300-1600 is notated in mensural notation, the rules of which changed little from c. 1350. Yet modern print editions distance today's readers from the original experience of this music: first, by translating the original notation into modern notation; and second, by sorting and classifying this repertoire according to conventions associated with the printed book (that is, presenting it in volumes or series ordered by composer, genre, or country), rather than its presentation in the original manuscript.

This poster presents an NEH-funded project to develop a prototype editor for encoding mensural notation (PI: Karen Desmond, Brandeis University). It builds on the Measuring Polyphony project (measuringpolyphony.org), a website that presents digitisations of polyphonic motets copied in late medieval manuscripts in mensural notation. Coding of the editor prototype began in January 2020, and a workshop directly before the 2020 Music Encoding Conference evaluated the prototype in terms of its interface and design, accessibility, and interoperability, and advised on a plan for the project's full implementation. This poster includes links to videos that outline the main functionality of the prototype and a summary of the project goals, impact, and next stages of development.

Project goals

An online mensural music editor will allow a variety of modern readers (students and experts, musicologists, music theorists, those interested in the history of music notation, the history of counterpoint, medieval palaeography or manuscript studies in general) to both access and contribute transcriptions of polyphonic music directly linked to digital images of the medieval manuscripts. The GUI prototype will allow users with no expertise in music encoding to encode large amounts of music data in mensural notation directly linked to digital images of the medieval manuscripts, thereby rapidly increasing medieval music repertoires available for study.

Currently there is no editor, either commercially available or available as an open-source web application, that allows users to notate music in its mensural form. This is a significant problem in general for scholars of early music. In order to produce music examples in mensural notation to include in publications, for example, images must be generated within graphics software such as Adobe Illustrator, a time-consuming and non-intuitive process. Graphics software cannot capture any data about the music's sounding aspects, that is, what the graphemes mean in terms of their pitch and duration. This project addresses this need through the development of an open-source web-based editor designed to capture the shapes and the meaning of the mensural notation, following the encoding standards developed by the Music Encoding Initiative.

Software and standards leveraged in this project

MEI

Verovio

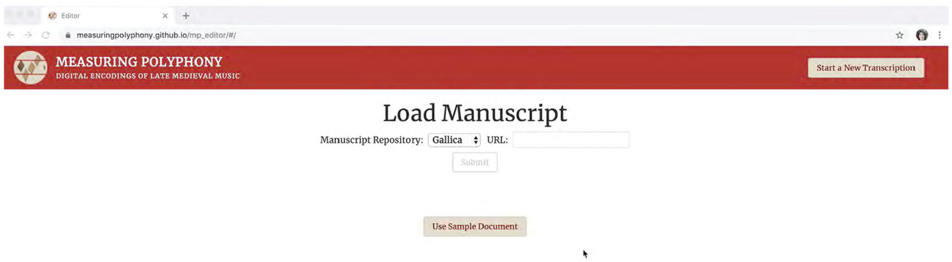
iiif

divajs

humdrum.org

The Measuring Polyphony Mensural Editor (click to play)

measuringpolyphony.github.io/mp_editor



Project impact and next steps

One envisioned outcome of the next stages of this project is the development of pedagogical modules in which students could learn about medieval notations: perhaps eventually providing a collective space to work on collaborative editions directly from, and linked to, the manuscript images. One pressing research question is whether the MEI data that captures the graphic information on the note shapes should be formally separated from the data necessary for score alignment and audio rendering. In some cases, the interpretation of the mensural shapes has only one correct answer, and follows specific rules established for mensural notation. But in other cases, the interpretation may be dependent on notational dialects used in a particular geographic location, or on a particular scribal practice, or the interpretation may be unclear. Methods by which these "flavors" of notation might be abstracted in our data model and captured via the editor were discussed in the pre-MEC workshop.

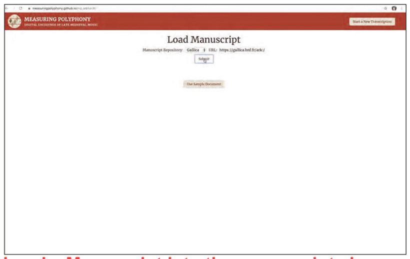
The major music editing projects of the twentieth century divorced the musical text from its parchment and ink origins, segregating them into composer- and genre-ordered print collections, converting their notation to modern equivalents, and secreting away detailed philological considerations into cryptic appendices. Since the late 1990s, the availability of high-resolution images of most manuscript sources of medieval polyphony, in particular through the groundbreaking open-access DIAMM initiative (Digital Image Archive of Medieval Music, diamm.ac.uk) and through library-based repositories such as Gallica (gallica.bnf.fr), has prompted new investigations of medieval music within its original material contexts, as is also the case within medieval studies in general (the "new" philology). It is envisioned that the final implementation of this project will allow experts and non-experts to move seamlessly between manuscript image and hear audio realizations of the compositions found there (for projects that do similar things, but which are focused on text rather than music, see the projects French Renaissance Paleography, paleography.library.utoronto.ca and Digipal, digipal.eu). Users will learn how to directly contribute transcriptions of the hundreds of as yet unedited (or poorly edited) pieces, facilitating new corpus studies of musical style. In addition, since the mensural notation will be digitally encoded, this project will inform new understandings of regional and scribal notational practices, and how changes in notation and representation engender changes in musical style.

NATIONAL ENDOWMENT FOR THE HUMANITIES

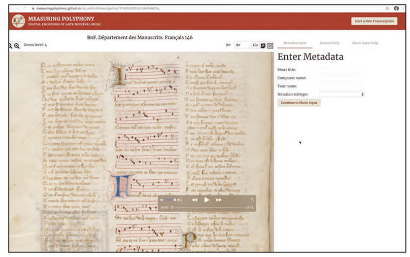
Brandeis UNIVERSITY

The support of the National Endowment for the Humanities and Brandeis University is gratefully acknowledged.

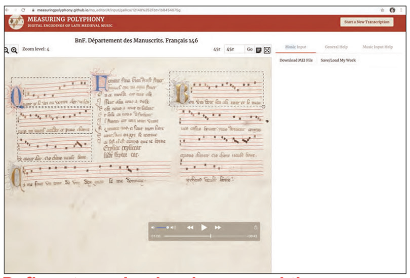
Help videos (click to play)



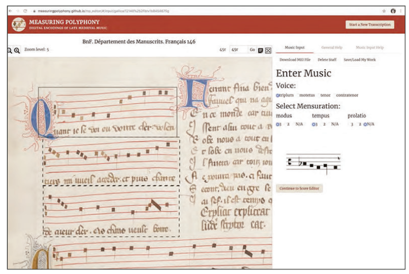
Load a Manuscript into the manuscript viewer



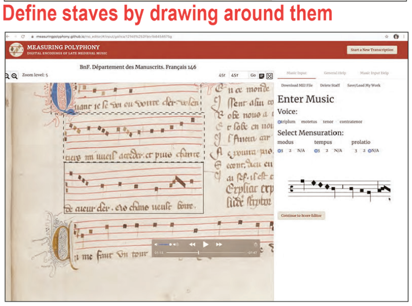
Enter some basic metadata



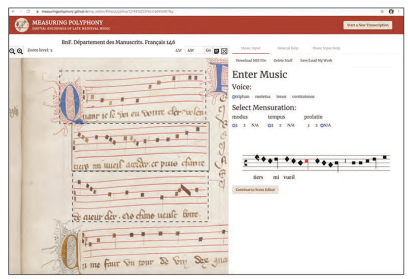
Define staves by drawing around them



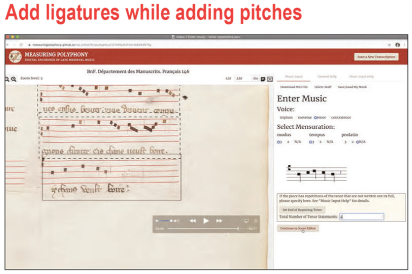
Enter the pitches and rhythmic values



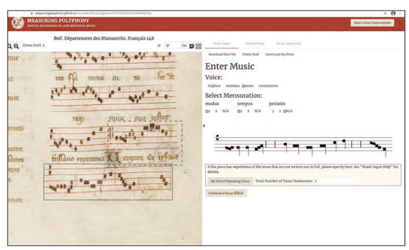
Add ligatures while adding pitches



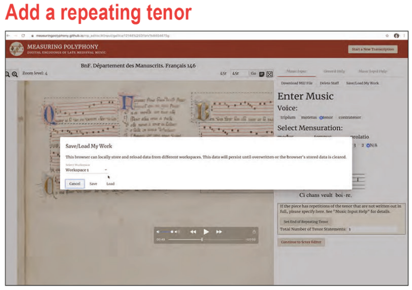
Enter underlaid text



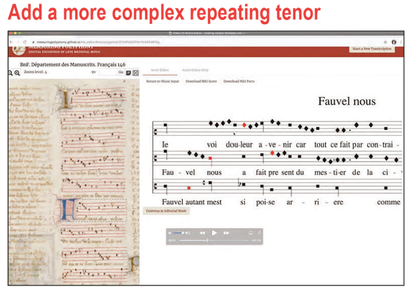
Add a repeating tenor



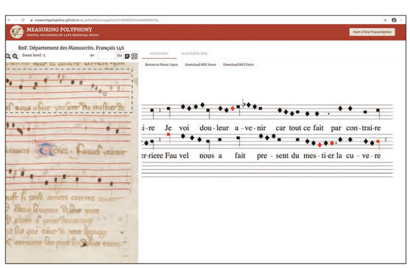
Add a more complex repeating tenor



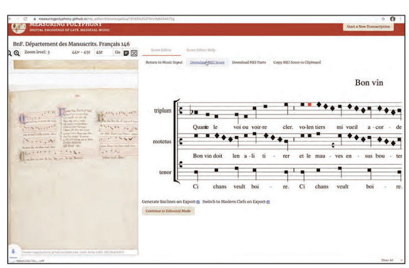
Save your work as you go



Score up the parts and make corrections



Simple editing of the score



Downloading the MEI files

Traversing Eighteenth-Century Networks of Operatic Fame

Estelle Joubert
Fountain School of Performing Arts, Dalhousie University
estelle.joubert@dal.ca

Abstract

This paper employs a digital project entitled "Visualizing Operatic Fame" to delve into three major issues in graph theory and network science: searching and pathfinding, influencers and hubs, and clusters and communities.

Introduction

"The public is the toughest and finest critic in the world, and yet, a clumsy folk tune is enough to amuse it for an entire year," writes J.F. Schulze with some exasperation in the *Deutsches Magazin* in 1798 [1]. The late eighteenth century witnessed a remarkable capacity for the public (rather than rulers) to act as critic and supporter of the performing arts. This new context also witnessed the rise of the classics and the much-debated musical canon, which was taking shape at the same time as Schulze's remarks. One might come away thinking that fame is fickle, subjective, and fleeting, thereby defying systematic analysis. And yet, the broader processes of collective ascriptions of value, and the emergence of the musical canon suggests that there may yet be some structure inherent these complex cultural processes. The question that lies at the heart of my current project is: how did music become famous at the time of the emergence of the classic? Did musical works acquire fame in view of criticism or performance? Are there broader patterns of attaining fame particular to specific artforms or, in the case of music, genres?

The problem of canon formation and the classic, I believe, is really a network problem: we're dealing with a dynamic network of people and things, and for the purposes of my project, people and things related to eighteenth-century opera. Librettists write operatic texts, sometimes in response to commissions or royal events; composers compose music, singers perform (and sometimes adapt arias), publishers print full operatic scores and collections of famous arias, manuscript copies of operas in full or part move via agents such as diplomats, travelers, or other composers; critics review opera performances and opera prints available to them, often via royal libraries or lending libraries. Some opera critics even comment on other critics' assessments, forming long and complex inter-related chains of aesthetic assessments leading to canon formation. In effect, all of these "actors," together contribute to processes of operatic fame. And, to be more precise, as physicist Albert-László Barabási puts it, "Networks are only the skeleton of complexity, the highways for various processes that make our world hum. To describe society we must dress the links of the social network with actual dynamical interactions between people" [2, p. 225].

My efforts to "dress the links of eighteenth-century musical networks" find their home in a large-scale team SSHRC-funded project called "Visualizing Operatic Fame". A project of this scope requires various kinds of expertise, and I gratefully acknowledge the contributions of my various team members: Austin Glatthorn (Postdoctoral Research Fellow), James Summerby-Murray (Technical Lead), Hilary McSherry (MA Candidate in Musicology); Shawn Henry (Graduate Research Assistant), Thomas Carberry (Undergraduate Research Assistant), Paul G. Doerwald (Technical Consultant).[1] *Visualizing Operatic Fame* asks the question: what factors contributed to operas being established as musical works during the latter half of the eighteenth century? Answering this question necessitates bringing together evidence from a wide range of sources (reviews, scores, performance calendars, catalogues and so forth). For computational purposes, this means a variety of data types. Each source type contains various kinds of data: for example, performance events (found primarily in

1 See the project website: http://operacanon.io This research was generously supported by an Insight Grant from the Social Sciences and Humanities Research Council of Canada.

theater calendars) includes the name of the opera, composer, date performed, names of performers (or at least the name of the troupe), and the performance space, usually, the name of the theatre.

Since my project involves a range of data, it features distributed data governed by relationships. This emphasis on relationships also determines the type of database. Most databases are relational databases, meaning that data is stored in highly structured tables; these tables can be linked using ids. By contrast, graph databases – the most famous example being facebook – emphasizes relationships between the nodes and are much easier to query. I decided to use a graph database, as the types of queries available to me in a relational database, simply didn't answer my questions. For example: early on in the project I endeavored to find out which operas were the most famous in a given timeframe. My results were typically a list of operas, which I could visualize in a bar chart. Among the limitations of relational databases is that all entries are treated equally: if one were to survey opera reviews, for example, there is no way to distinguish between an opera review that is a paragraph in length versus one that is many pages in length. Each result in a relational database is weighted the same, and relationships are difficult to foreground or query.

Visualizing Operatic Fame is a graph database powered by neo4j (the leading commercial graph database platform) and uses Cypher as query language. More recently, we have turned to using neo4j Bloom as our visualization tool, as it is now free for desktop use. The tool is designed to represent relationships not only between individuals (composers, opera singers, music publishers) but also objects related to operatic fame (scores, reviews, images of actors). The data-model has been tweaked a number of times so as to sharpen queries and visualizations.

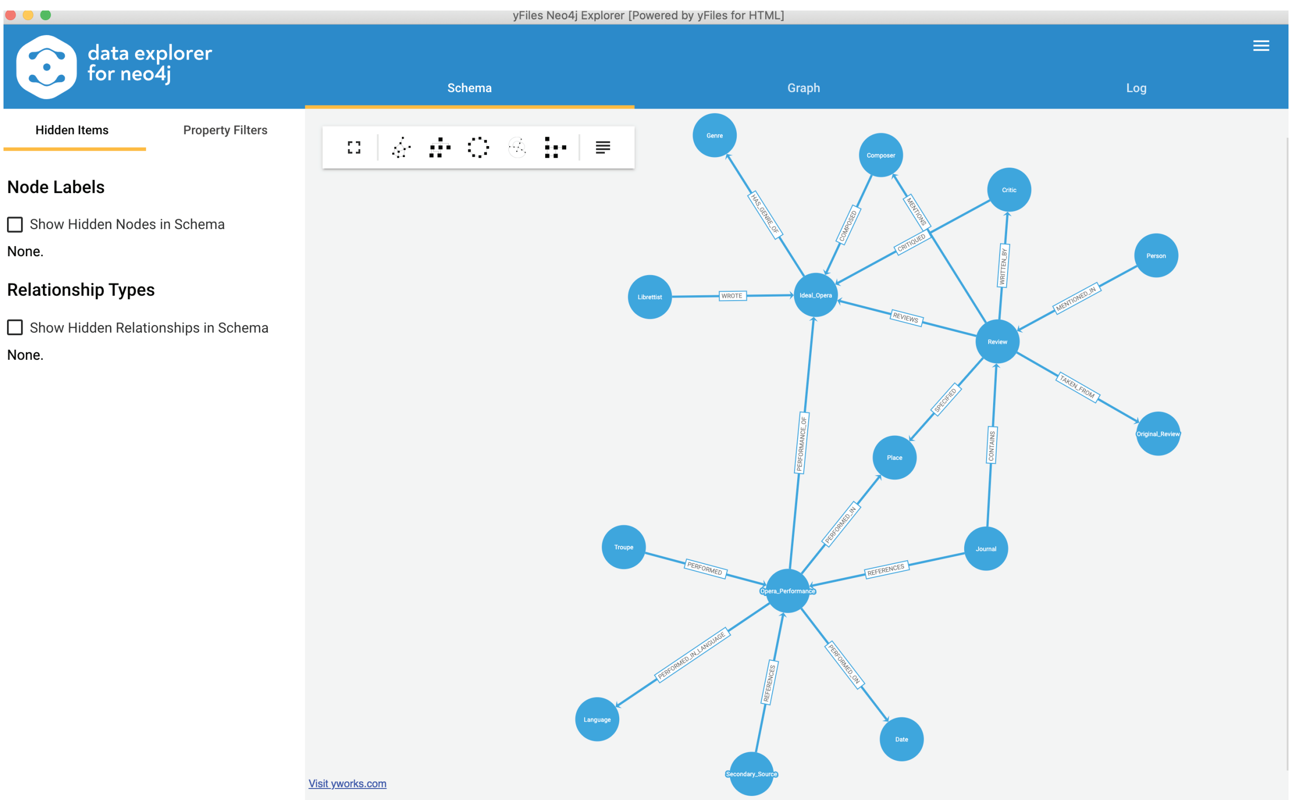


Figure 1: Current data model

Currently, we have, for instance, people such as critic and composer, and objects such as journal as nodes (incidentally, these designations are known as labels for each node). Edges (these are arrows that connect nodes) are directional in neo4j and multiple connections between two nodes are possible. One of the central questions in theories of the musical canon is whether works become famous in view of multiple performances, or whether canon is really a function of music criticism. To get to the bottom of this debate, my data model has an "ideal opera" node, which allows me to distinguish between opera performances (its own node) and opera reviews (also its own node). Put another way, I am able to distinguish between a review of a performance and the review of an operatic work, more generally. We have endeavored to be quite precise about the relationships, as it is also possible to search by relationship type. For example, in our data model a composer "composed" operas, whereas a librettist "wrote" a libretto and a critic "critiqued" an opera. This allows searches that distinguish between the librettist's and the composer's relationship to an opera, potentially routing a query through a different path in the network. Although this representation of our data model does not show them, each of the nodes also have distinct properties, which facilitate querying. For instance, my Opera Performance node has 10 properties, some of which (city, coordinates, performance date) will allow us to map the results later on.

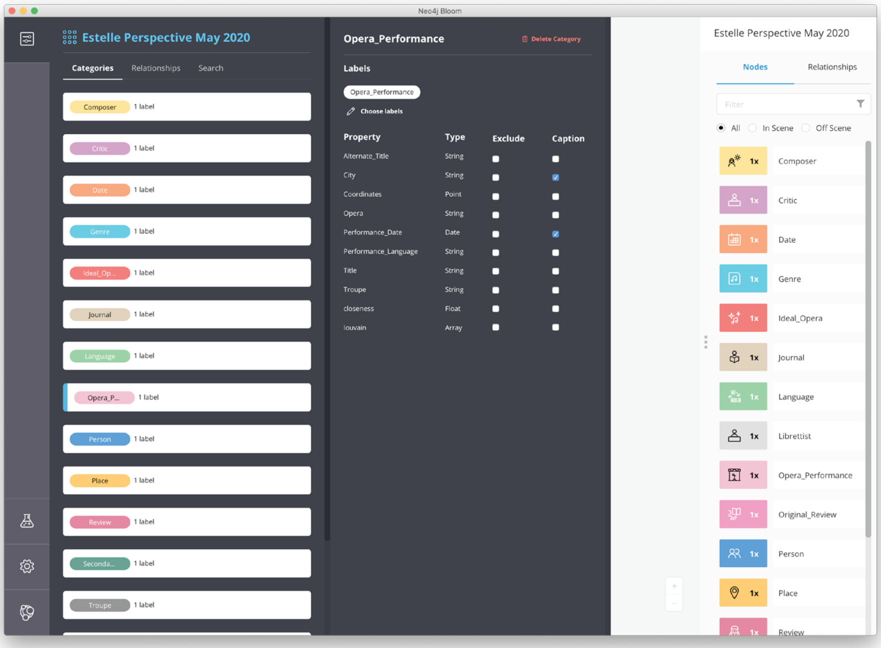


Figure 2: neo4j Bloom interface, showing node properties (left)

This also shows the datatype for each property, strings being the most common. My talk will use examples from this database, but it will focus more broadly on three common types of queries that one might encounter in graph theory: searching/pathfinding, influencers/hubs and clusters and communities.

Traversing the Network: Searching and Pathfinding

Given that my database contains over 37,000 nodes, it is obviously unworkable to call them all up at the same time. Instead, it is more productive to choose an entry-point into a graph and expand relationships as desired. I'll begin with Maria Antonia Walpurgis, Saxon Electress and composer. She is important in the history of the musical canon as her operas were the first musical works to receive reviews that included in-text music examples during the 1750s, in turn enabling detailed commentary on the score itself. Switching to neo4j Bloom, a visualization tool, the first step is to search for the composer node, Maria Antonia.

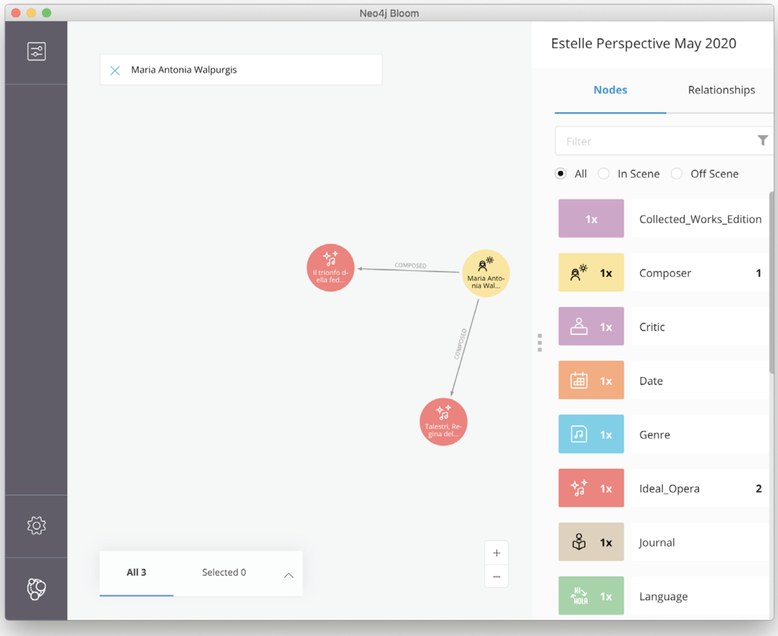


Figure 3: Searching for the composer node: Maria Antonia

Clicking on the node allows one to see the various kinds of relationships, and it is possible to reveal some or all of these (I will reveal, in this case, the nodes associated with her most famous opera, *Talestri*). We now have two Maria Antonia nodes (a second one, as librettist, appears, as she also wrote the text to her opera). The full score node, in blue, can be expanded yet again, to reveal a review of that particular score by the critic Johann Friedrich Agricola.

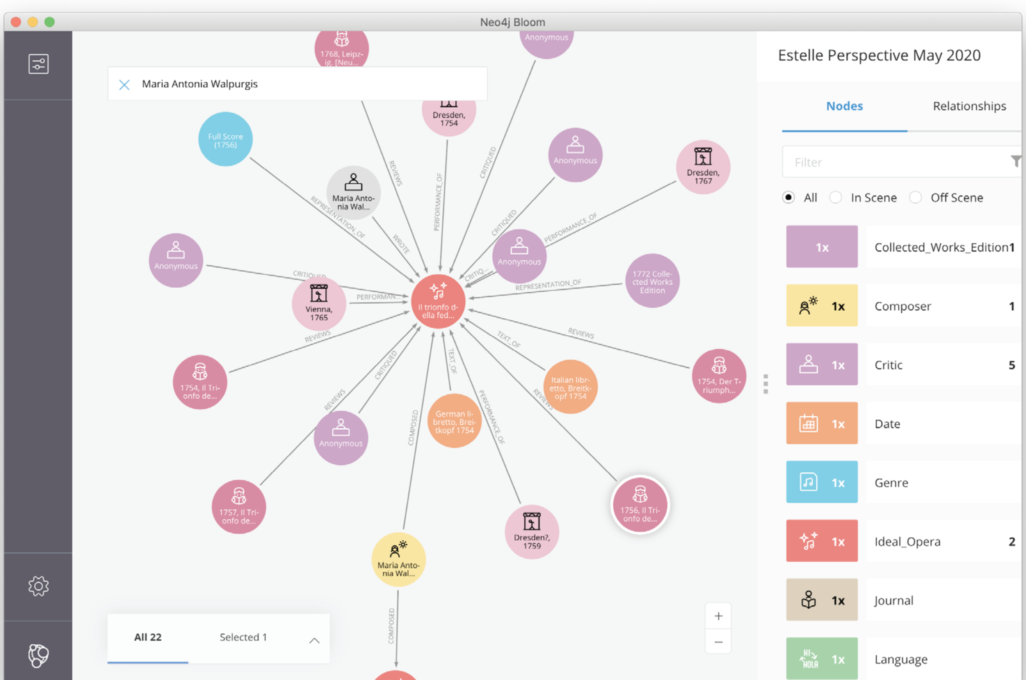


Figure 4: Expanding pathways connected to Maria Antonia's *Talestri*

A number of strengths of a graph databases come to the fore here: first, reviews are not merely ascribed to a composer, or even a work. Instead, criticism can be directly ascribed to a specific performance of a work, the printing of a libretto, or the printing of a particular score. These results seem to reinforce the material dimensions of opera criticism; unlike later Kantian ideals associated with aesthetic autonomy and canon formation, visualizing how operatic criticism was generated uncovers just how directly reviews (and consequently also aesthetic judgements) were formed in response to the material conditions of operatic performance and mobility. One curious finding is that amongst the performances and prints that received criticism, the collected works edition of 1772 received no critical attention. Visualizing these connections thus offers a much more nuanced picture of which objects related to Maria Antonia's operas actually effected fame, and which ones did not. In a sense, it prevents assumptions about what we often presume must have generated fame, and what demonstrably (or empirically) did.

In addition to exploring the network starting with a particular node, it is also possible to search for pathways between two nodes. One of the most popular pathway searches in graph theory is the so-called "shortest path" search. This is particularly useful when trying to ascertain whether two people, or a place and person, for instance, are connected. For example, are there any direct connections between Gluck and Johann Adam Hiller? From here on forward, my queries have been done in the neo4j browser, as precise querying is easier than in neo4j Bloom.

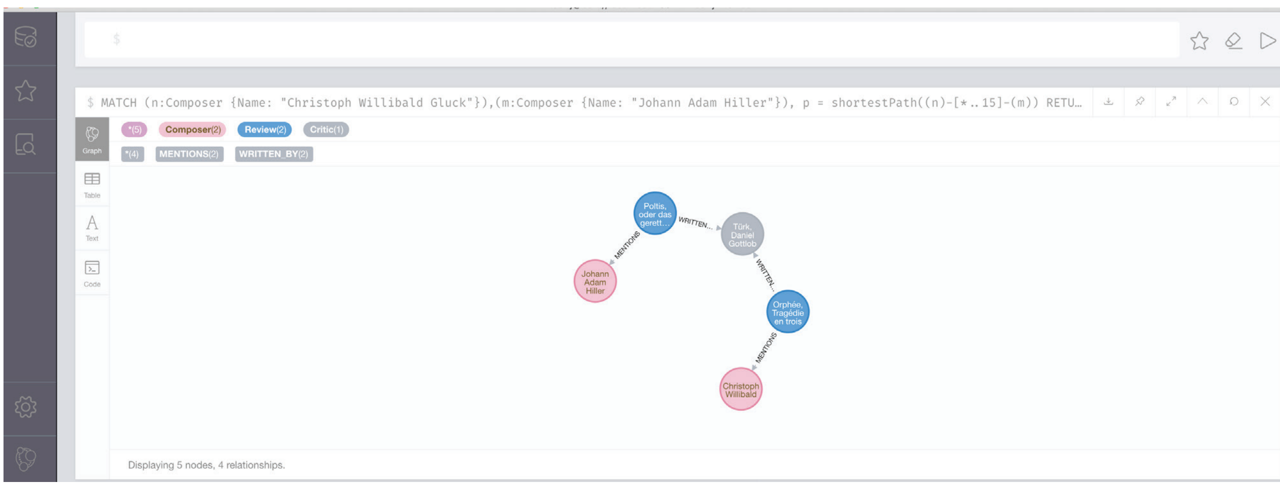


Figure 5: Shortest path search from Gluck to Hiller

Here, the result is fairly simple: there was at least one connection (this one is the shortest path; there may be other longer paths). As it turns out, Gluck and Hiller were both reviewed by Daniel Gottlob Türk. Now, for a more complex example: did Mozart have connections to Berlin? We know that he traveled there once in 1789, but we're now interested in how prominent Mozart's music was there and how Berliners came to know the composer's music.

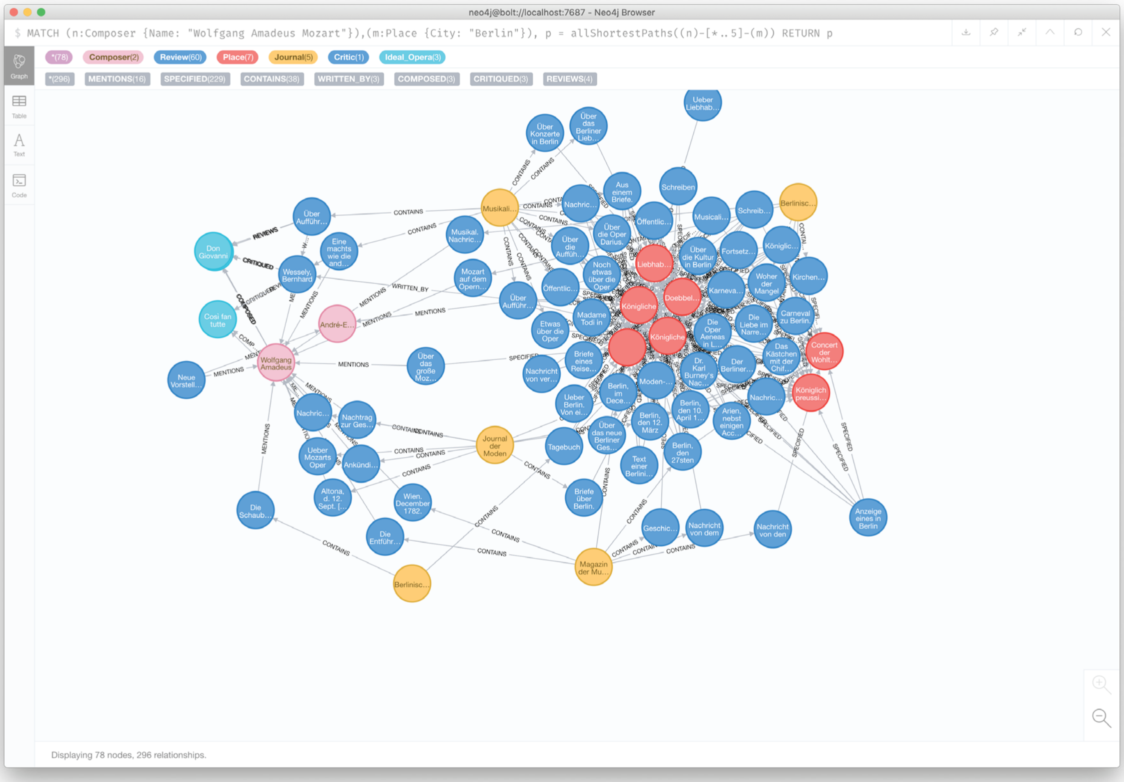


Figure 6: Paths between Mozart and Berlin

There are indeed many paths between Mozart and Berlin. The Mozart composer node is on the left-hand side; connected to it are two of his operas, *Don Giovanni* and *Così fan tutte*. Most of the connections in Berlin are reviews connected to five journals: the *Berlinisches Archiv*, *Journal der Moden*, *Magazin der Musik*, the *Musikalisches Wochenblatt* and *Berlinische Musikalische Zeitung* (these are the yellow nodes). The red node without a label is the city of Berlin, and we also have connections to some of Berlin's musical institutions, most notably the Königlich preussische Hofkapelle (the royal Prussian court chapel), the Königliche Oper (royal opera) and Döbbelin theatre troupe (these are also red nodes). Judging by the short titles of some of the reviews "Über Konzerte in Berlin" (Concerning concerts in Berlin) or "Öffentliche Musik in Berlin" (Public music in Berlin), one might surmise that some of these reviewers are describing eighteenth-century concerts, which often contained separate arias from operas, among other musical numbers.[2] While some reviews offer direct coverage of an opera performance of *Don Giovanni* and *Così fan tutte*, it seems much more likely that Berliners might have gained familiarity with Mozart's music through excerpts in public concert life and via reviews of his music (alongside publications of his scores, of course; the database does not yet list the vast quantities of sheet music circulating during this period). This search for pathways between Mozart and Berlin not only yielded a fairly complex portion of the network. It also hints at our next topic in graph theory: influencers and hubs.

Influencers and Hubs

In his ground-breaking work on network science, Albert-László Barabási examines the underlying structures of complex networks in a wide range of disciplines, including the movement of fish in oceans, the spread of disease, transportation as well as various social networks. Almost all networks, he argues, include an uneven distribution of nodes, the clustering of nodes, and ultimately the formation of hubs [4]. Music networks are no exception. For, as we are well aware, some composers received many more performances and much more critical attention than others. A search for hubs – nodes of particular importance – in the current dataset re-

2 For a discussion on programming concerts in the eighteenth and nineteenth centuries, see [3].

veals the following: the top seven influential nodes are: Johann Adam Hiller, Johann Gottlieb Naumann, Christoph Willibald Gluck, Johann Friedrich Hönicke, Antonio Bianchi, Wolfgang Amadeus Mozart, and Friedrich Ludwig Brandes (these are in orange, Figure 7). Neo4j has grouped these in clusters, surrounded by nodes representing their popular works (in pink) and reviews that generated fame in the public mind (in green).

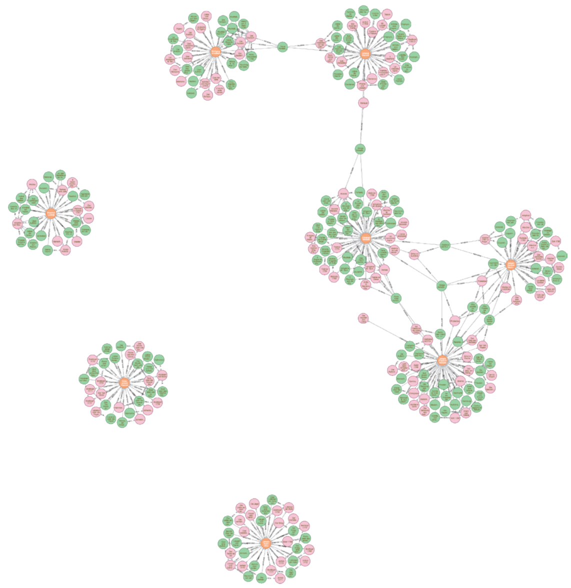


Figure 7: Composer hubs

It is also possible to get a view of these hubs in the network, and this reveals much more context surrounding these influencers. Focusing in on Gluck, for example, we notice that his composer node is close to that of his librettist, Calzabigi. The works that are most prominent (pink nodes) are *L'arbre enchanté*, *Iphigénie en Aulide*, *Alceste* and *Armide*. *Orfeo et Euridice*, a work often taught in music history surveys, seems to have far fewer connections. I would seem that *Alceste*, has many more connections to music criticism. Notably, the German version of *Alceste* (likely performed in Vienna in the early nineteenth century), did not feature prominently in criticism. What is missing here, and indeed, is a next step for this project, is the dimension of time. For fame can indeed be fleeting, and queries such as this one show only a representation of the entire period, 1750-1815. If one were to query by decade or indeed, create a dynamic time-lapse representation, one would like see that Gluck's fame was established with the discourse surrounding *Alceste*, that his fame remained relatively stable, and that toward the late eighteenth and early nineteenth century (that is, after the composer's death), his fame was sustained by performances rather than continued criticism. Performances of works such as *L'arbre enchanté*, *Iphigénie en Aulide* and *Armide* were staples in early nineteenth century Vienna, when there was a craze for French opera (often performed in German), following the arrival of Napoleon's troupes in the city in 1806. Scrolling a bit to the right, one also gets a sense of how Gluck is connected to the broader network of eighteenth-century opera: he is connected to Lully through a review entitled "Gluck und Lulli" but he is also connected to Handel and Rameau through criticism (Figure 8).

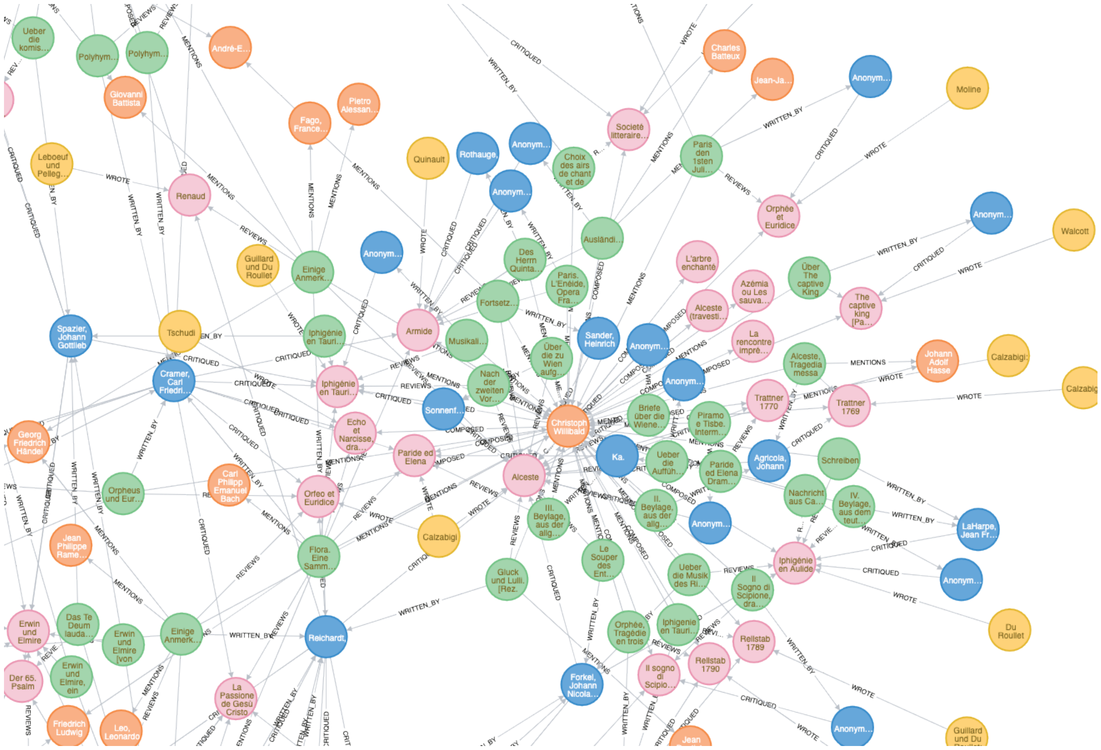


Figure 8: Close-up of composer hubs

Moving to the lower left-hand corner, we also see the prominent hub of a lesser-known composer and musician: Antonio Bianchi.

Figure 9: Structures of composer hubs

And here, the obvious question is: do the hubs of canonic composers look different in structure compared to those who are lesser known. And, for scholars interested in machine learning and AI, the obvious next step is: can one predict fame, especially for musicians today? Neo4j does offer some predictive algorithms in their graph data science playground, though they are relatively new and still experimental. At first glance, Bianchi seems to have composed a similar number of operas, compared to Gluck and most of his operas are connected to a review, an ideal opera and a librettis.

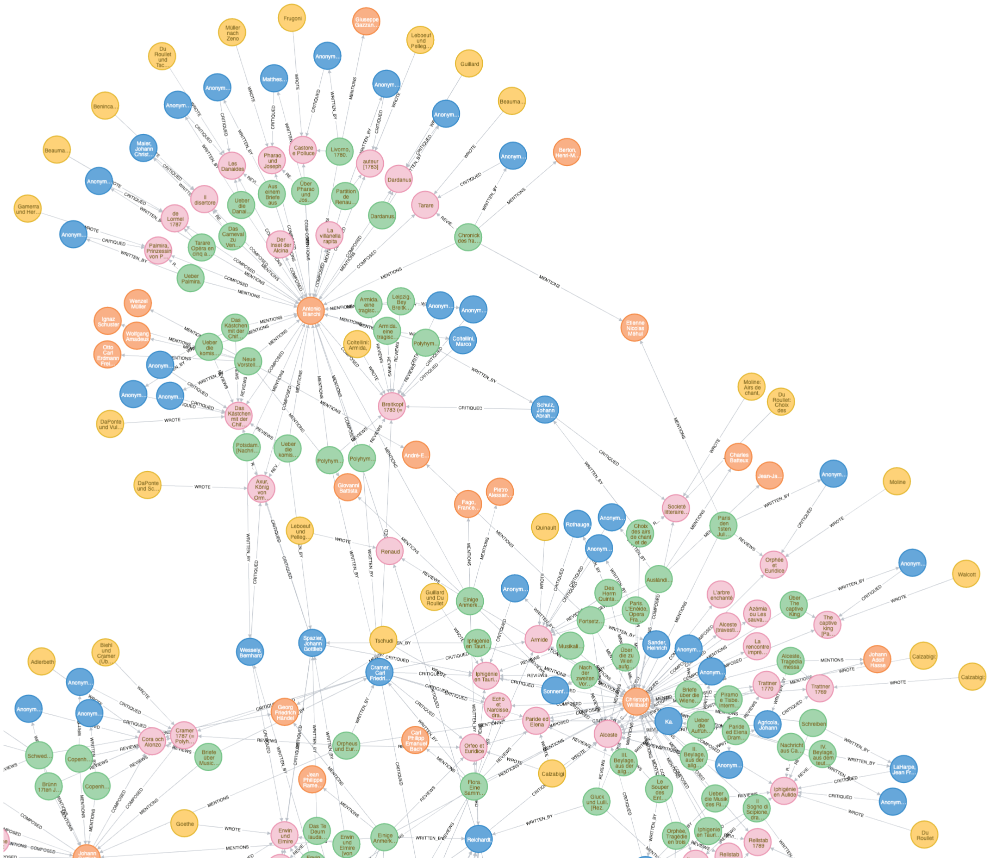


Figure 10:

Yet somehow the Bianchi hub simply looks much cleaner than the Gluck hub; fewer pathways traverse through the node, and the context seems "less messy" if you will. While it is far too soon to connect messy networks structures to lasting fame – clearly much more work remains to be done on this – this idea of patterns of fame surrounding hubs holds much promise.

Clusters and Communities

Hubs bring to prominent nodes (individuals or works) to foreground, while community detection in graph theory is concerned with similar things naturally grouping together. In a social network, some nodes naturally have more connections than others, and musical communities form around performance events, genres, debates in music criticism, societies in particular cities and star singers, to name only a few. It is fair to assume that communities play a substantial role in generating musical fame and by extension, canon formation. Much more challenging, however, is to uncover these communities in a given network and analyze patterns within those communities. Let's begin with a relatively simple example: a search for troupes connected to performances of Mozart's *Die Zauberflöte* (Figure 11).

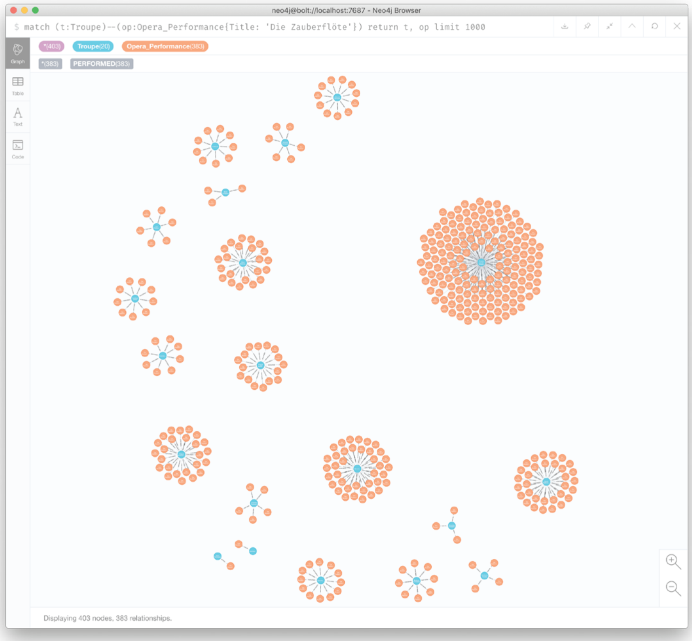


Figure 11: Performances of Mozart's *Die Zauberflöte*

The query returns a range of clusters of performances, each stemming from a theatre troupe or resident theatre company. The largest cluster, of course, is the Theater auf der Wieden in Vienna, where the Mozart's famous opera was premiered. But, as is evident, the opera was also performed at the Kärtnertortheater in Vienna, the National Theatre in Berlin and the Prague National Theater (these are three second largest communities). Subsequently, we have even smaller centers such as Mannheim and Innsbruck as well as traveling troupes such as the Schuch and Vollotini companies. Queries such as this one tell us about the relative size of performance clusters for a famed opera, which is a useful for a broader bird's eye-view perspective, perhaps prior to zeroing in on a particular group.

Yet along with communities comes the challenge that social networks do not only have bi-directional edges, but in some cases, especially in global music mobility, there are asymmetrical power-relationships at play. Here, I turn to an example from the work of one of my students, Sr. Ilaria Culshaw, and her paper for my research seminar on operatic mobilities [Figure 12].

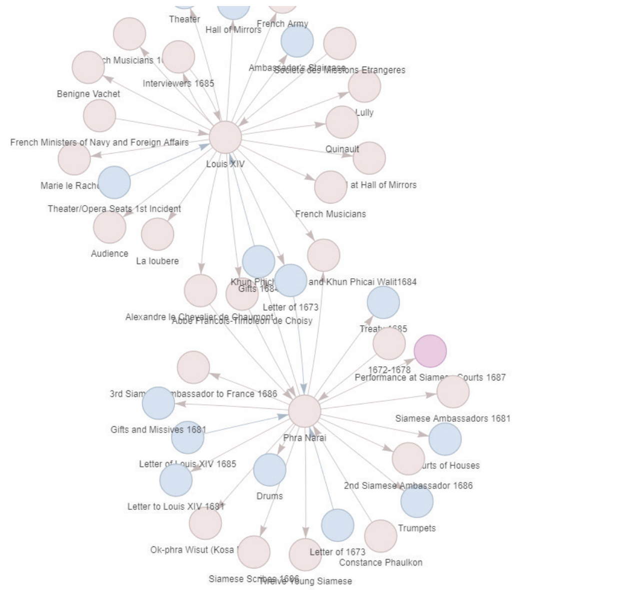


Figure 2 The main hubs which form around the two monarchs.

	Louis XIV	Phra Narai
Total Number of Connections	22	22
Number of Connections representing the monarch's own agency	18	15
Number of connections representing the monarch being acted upon	4	7

Figure 12: Sr. Ilaria Culshaw's visualization of French-Siamese relations using graph9

Culshaw graphed (using our custom graph visualization software called graph9) French-Siamese musical encounters between Louis XIV and the Phra Narai in the late seventeenth century [5]. Her work was based on primary source documents analyzed in a study by my colleague David R. M. Irving in his article, *Lully in Siam* (present-day Thailand).[3] One of the figures in her essay represents the two communities, each with their respective monarch in the centre. That all the arrows point back to each of the monarchs arguably illustrates an important point about these societies: power is concentrated at the centre and in a single person in monarchical societies. In this particular diplomatic visit, cultural exchange could only be facilitated by people and things represented here by a few common nodes: some ambassadors from both parties and a letter of 1673. With so few commonalities, it is perhaps unsurprising that this cross-cultural encounter went awry, like so many others in the early modern period. This example illustrates two notable features of graph theory: that two communities can exist entirely on their own, at great geographic distance, and that it only takes a few (sometimes even one) connecting node to suddenly collapse the distance between two far-away communities. Of course, important nodes that connect two otherwise separate communities appear, disappear, and reappear over time. Second, at least on French soil there was an asymmetrical power-relationship at play. More sophisticated graphing tools and algorithms able to cope with weighted nodes would bring these asymmetrical relationships into relief.

While monarchical societies have a particular structure, so do democratic societies, and my database reflects society at a time in which liberal-democratic ideals were just beginning to take hold. Public perceptions of fame reigned supreme by the eighteenth-century, as Schulze, the critic commenting on the year-long pop-

3 https://github.com/ejoubert/graph9

ularity of a clumsy folk-tune suggests. Yet even within the public sphere, musical communities might differ in their structure, make-up and impact on generating musical renown. One of the advantages of graph theory is that it has the potential to both detect community formation and, with appropriate algorithms, detect patterns distinct to individual communities. The neo4j graph data science playground is a relatively new plug-in intended to facilitate the exploration of datasets with complex algorithms. The idea behind the playground is that one does not need to deal with the code directly but can still benefit from the power of often-called algorithms through an easier-to-use interface. However, the tool is still experimental. I was able to get some preliminary results for the community-detection (Figure 13).

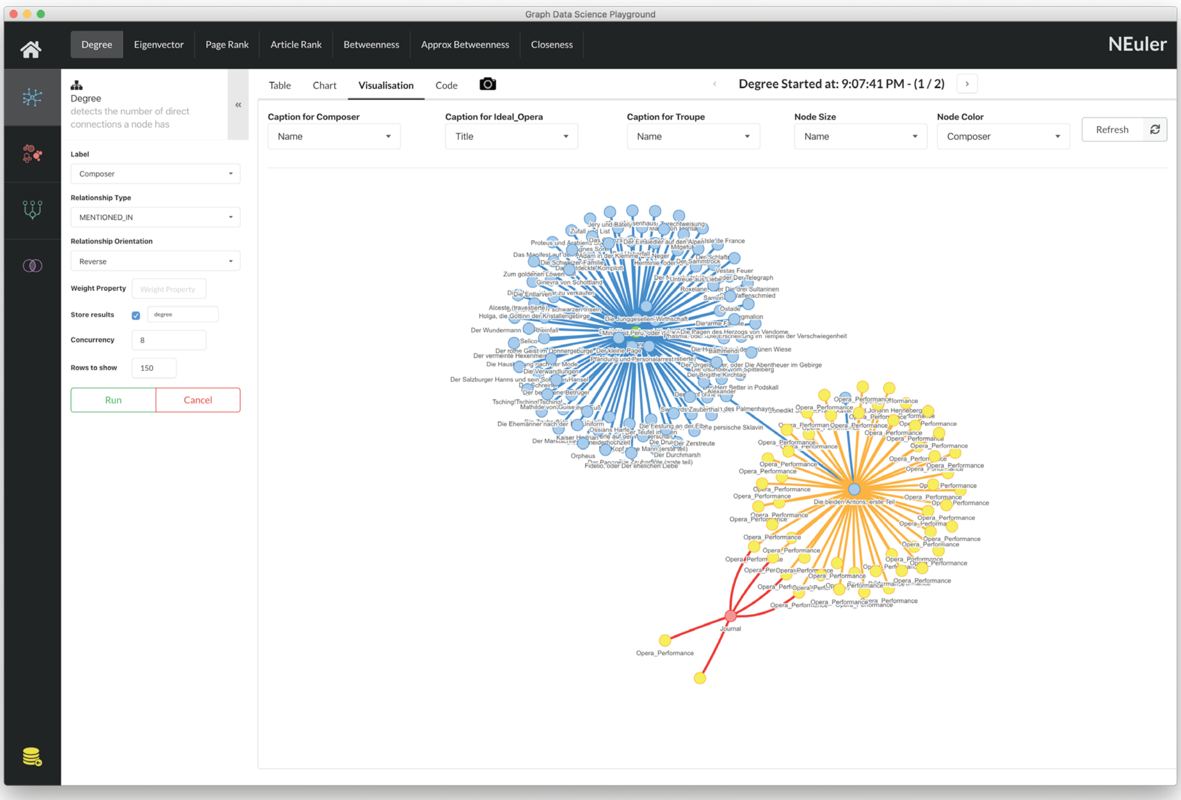


Figure 13: Emerging communities using neo4j Graph Data Science Playground

The largest community seems to be centered, unsurprisingly, around genre: the German opera node. A subset of that community is centered on an opera that opened the Theater auf der Wieden just two years before Mozart's *Die Zauberflöte* was premiered there: Benedict Schack's *Die beiden Antons*. In fact, this suggests that the community for *Die beiden Antons* is likely stronger (or perhaps larger) compared to that of Mozart's opera. Future research for this project will be focused on pattern detection to see how various communities (once identified) are structured, and how that structure has an impact on canon formation. Some interesting questions might include: did communities for symphonic music look different than communities concerned with opera? Do the subcommunities have similar structural patterns or do some of them differ (and why?). Does the behaviour of musical communities change over the course of time? While graph theory has already provided valuable insights into how operatic fame is generated in my dataset, many more insights remain to be discovered traversing these networks of operatic fame.

Works Cited

[1] Schulze, J.F., “Publikum” *Deutsches Magazin* 15 (1798), 332-6.

[2] Barabási, Albert-László, *Linked: The New Science of Networks.* Cambridge, MA: Perseus Publishing, 2002.

[3] Weber, William, *The Great Transformation of Musical Taste*. Cambridge: Cambridge University Press, 2008.

[4] Barabási, Albert-László, *Network Science.* Cambridge: Cambridge University Press, 2016.

[5] Irving, David R. M., “Lully in Siam: Music and Diplomacy in French-Siamese Cultural Exchanges, 1680-1690” *Early Music* 40 n. 3 (2012), 393-420.