MEI in MIR
The JOSQUINTAB dataset

Reinier de Valk
Moodagent A/S, Copenhagen

IMS Study Group 'Tablature in Western Music' meeting
Basel Lute Days
Basel, Switzerland, 14 September 2022

The project | JosquIntab

JOSQUINTAB: A DATASET FOR CONTENT-BASED COMPUTATIONAL ANALYSIS OF MUSIC IN LUTE TABLATURE

Reinier de Valk[1] Ryaan Ahmed[2] Tim Crawford[1]
[1] Department of Computing, Goldsmiths, University of London, [2] Digital Humanities, MIT
r.f.de.valk@gold.ac.uk, rahmed@mit.edu, t.crawford@gold.ac.uk

ABSTRACT

An enormous corpus of music for the lute, spanning some two and half centuries, survives today. Unlike other musical corpora from the same period, this corpus has undergone only limited musicological study. The main reason for this is that it is written down exclusively in lute tablature, a prescriptive form of notation that is difficult to understand for non-specialists as it reveals little structural information. In this paper we present JosquIntab, a dataset of automatically created enriched diplomatic transcriptions in MIDI and MEI format of 64 sixteenth-century lute intabulations, instrumental arrangements of vocal compositions. Such a dataset enables large-scale content-based computational analysis of music in lute tablature hitherto impossible. We describe the dataset, the mapping algorithm used to create it, as well as a method to quantitatively evaluate the degree of arrangement (goodness of fit) of an intabulation. Furthermore, we present two use cases, demonstrating the usefulness of the dataset for both music information retrieval and musicological research. We make the dataset, the source code, and an implementation of the mapping algorithm, runnable as a command line tool, publicly available.

1. INTRODUCTION

An enormous corpus of music for the lute, roughly spanning the sixteenth, seventeenth, and first half of the eighteenth century, and containing an estimated 60,000 individual pieces in circa 860 sources, printed and manuscript, survives today [27]. Although the lute was one of the most widely used instruments during these two and a half centuries, and this corpus is thus extremely rich in information about daily musical practice, up until the present day it has not undergone the same critical musicological study as other musical corpora from the same period. The main reason for this is that the notation used to write down lute music, lute tablature, is notoriously difficult to understand for non-specialists [18, 28]. This is because lute tablature is a purely *prescriptive* form of notation: like all forms of instrumental tablature [12], it merely provides the actions a player must take—in this case, where to place the fingers on the fretboard and which strings to pluck—rather than the sounds and musical edifice these actions produce, which a *descriptive* form of notation does [20, 29]. In practice, this means that lute tablature, as opposed to mensural forms of notation, conveys virtually no information about the structure, polyphonic or other, of the music it encodes, making it hard to comprehend *a prima vista*.

In this paper we propose to address the problem of the marginal position of music in lute tablature within musicological research by providing JosquIntab, a dataset of 64 automatically created *enriched diplomatic transcriptions*—literal transcriptions annotated with voice, key, and mensuration information—of a selection of sixteenth-century lute and vihuela (the lute's Spanish counterpart) *intabulations*, instrumental arrangements of vocal compositions, as well as a mapping algorithm for creating such a dataset, and an implementation thereof, runnable as a command line tool. The proposed dataset enables large-scale content-based computational analysis of music in lute tablature, which, in the absence of some notion of the music's polyphonic structure, has hitherto not been possible. The command-line tool allows researchers to extend the dataset by creating their own transcriptions.

In summary, the main contributions of this paper are:

- a dataset of 64 automatically created enriched diplomatic transcriptions of lute intabulations, each in the original and in an unornamented version;
- a mapping algorithm for creating such a dataset;
- an implementation of the mapping algorithm, runnable as a command line tool.

Furthermore, we provide:

- a method to quantitatively evaluate the degree of arrangement (goodness of fit) of an intabulation, which is incorporated in the mapping algorithm;
- two use cases, demonstrating the usefulness of the dataset for both music information retrieval (MIR) and musicological research.

In the spirit of open science and reproducible research, we aim to make the resources we contribute FAIR (findable, accessible, interoperable, and reusable) [32]. We do so by using *recommended formats*—MIDI and MEI—for the dataset,[1] and by making the dataset, the source code for the mapping algorithm, and the command line tool publicly available.

[1] https://www.loc.gov/preservation/resources/rfs/

© Reinier de Valk, Ryaan Ahmed, Tim Crawford. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Reinier de Valk, Ryaan Ahmed, Tim Crawford. "JosquIntab: A dataset for content-based computational analysis of music in lute tablature", 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 2019.

- A 'new' **dataset** [1]
  - "[A]utomatically created enriched diplomatic transcriptions of sixteenth-century lute intabulations [of Josquin compositions]"
  - "[E]nables large-scale content-based computational analysis of music in lute tablature"

[1] R. de Valk, R. Ahmed, and T. Crawford (2019). JosquIntab: A dataset for content-based computational analysis of music in lute tablature. In *Proceedings of the 20th ISMIR Conference, Delft, the Netherlands* (pp. 431-438). https://archives.ismir.net/ismir2019/paper/000051.pdf

The project | Transcriptions

- The dataset contains enriched diplomatic **transcriptions**
  - Voice, key, and mensural information is added
- These transcriptions are created **automatically**
- They are **‘intermediate’ or ‘working’ versions** – not end products!

Workflow |

- **Data (input) collection** (cleaning, correcting)
  - **Encodings** of lute and vihuela **intabulations** of vocal works by Josquin, taken from the Accessible Lute Music web page: https://www.lutemusic.org/ ('Gerbode')
  - **MIDI renderings** of scholarly editions of the **vocal models**, taken from the Josquin Research Project's (Stanford, CCARH) GitHub repo: https://github.com/josquin-research-project/ ('JRP')
- Design a **mapping algorithm** organising intabulations polyphonically by superimposing them on their model
- **Data (output) creation**: apply the mapping algorithm

Mapping algorithm |

```
function map(M, G, o, v) returns list
  ornamentations O, voices V := []
  for i in range(|M|) if M_i != null:
    if |M_i| == 1 and dur(M_i) <= o
      and G_i == null:                           # Step 1
      add M_i[0] to O; add null to V
    else:
      unmapped notes U := []
      free voices F := [j for j in range(v)]
      for note n in M_i:                         # Step 2
        if n in G_i:
          add voice(n) to V
          remove voice(n) from F
        else:
          add n to U; add null to V
      if |U| > 0:                                # Step 3
        pairs A := [(pitch(f_{i-1}), f) for f in F]
        compute cheapest mapping U→A
        replace elements in V; empty U
      if O[|O|-1] == M_{i-1}:                    # Step 4
        compute cheapest connection O[|O|-1]→M_i
        replace elements in V; empty O
  return V
```

Mapping algorithm | Steps

- **Input**: per piece, a tuple of files
  - Machine-readable **encoding** (`.tc`) of an intabulation
  - Machine-readable **rendering** (`.mid`) of an edition of the model
- The algorithm **'maps' the tablature notes** onto voices using the vocal piece as a guide
  - **Exact matches** form the core polyphonic structure
  - All remaining notes — **mismatches** — are handled by means of a number of predefined rules

Mapping algorithm | Steps

- **Output**: per piece
  - The transcription as MIDI — for further **computational** usage
  - The transcription as MEI (`.xml`) — for **human** usage
  - The mapping details (`.csv`) — for **statistical** analysis, e.g., calculation of the *goodness-of-fit* of the intabulation

$$m = \frac{|N| - (o|M_o| + r|M_r| + f|M_f| + |M_a|)}{|N|}$$

Example | Intabulation (`.tc`)

Francesco Spinacino, Adiu mes amours. *Intabulatura de lauto, Libro primo* (1507), f. 32v

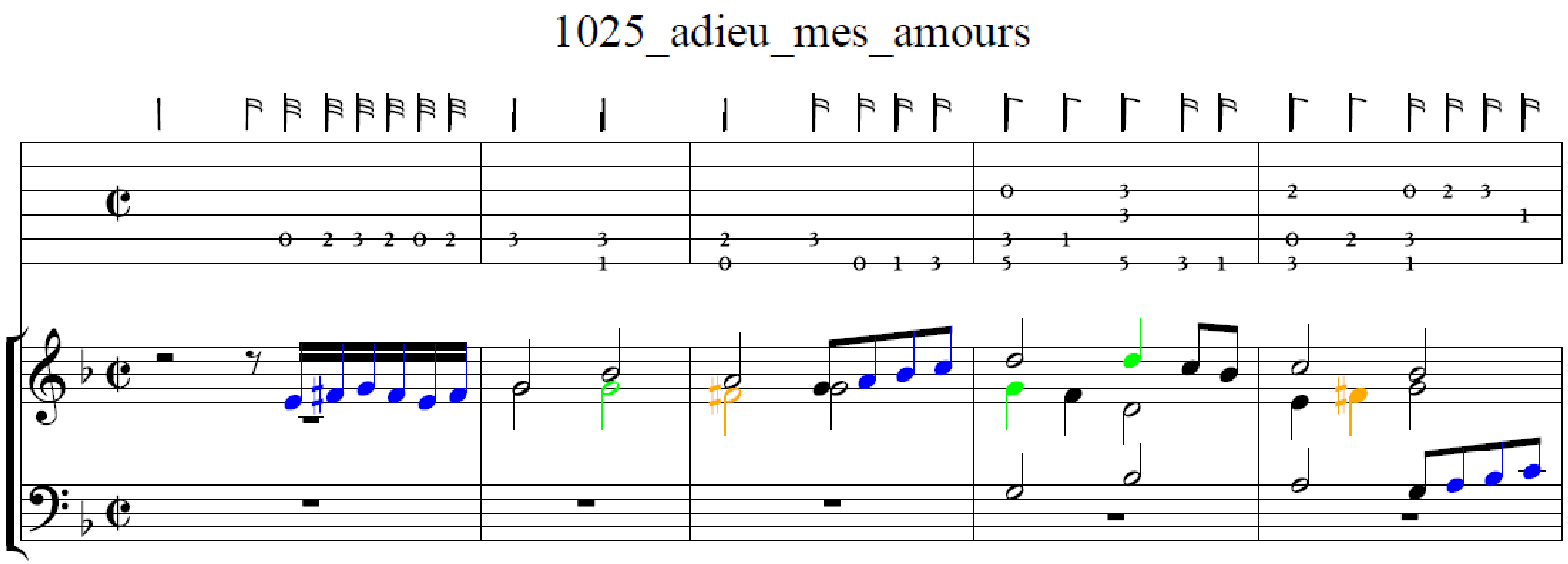
Example | Model (`.mid`)

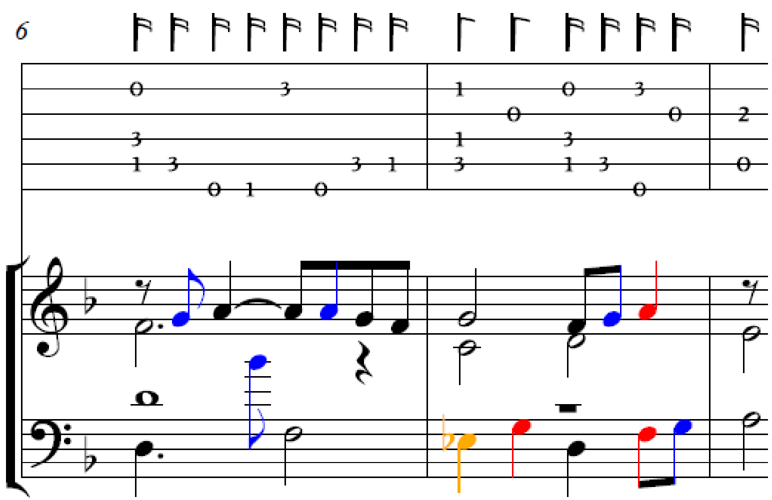
Josquin des Prez, Adieu mes amours. The Josquin Research Project.
https://josquin.stanford.edu/cgi-bin/jrp?a=notationEditText&f=Jos2803

Example | Transcription (`.xml`)

1025_adieu_mes_amours

Example | Transcription (`.xml`)





- *m*: **weight** to evaluate at different granularity levels
  - (57-(17+3+3+3)) / 57 = **0.54**
  - (57-(0+0+3+3)) / 57 = **0.89**
  - (57-(0+0+0+3)) / 57 = **0.95**

The dataset |

- Goal: transcribe ***all*** **intabulations** of Josquin compositions
- Intabulated for **lute/vihuela** are, according to the *NJE* [2]
  - 18 chansons, 17 motets, 17 (partial) mass sections, and 8 complete masses = a total of 60 works — in over **200 intabulations**
- At the moment, JosquinTab contains **64 pieces** (WiP!)
  - JRP provides editions of all vocal models; Gerbode encodings of **64 intabulations** (9 chansons, 21 motet parts, 12 mass section parts)
  - https://github.com/reinierdevalk/data/tree/master/josquintab/

[2] W. Elders and E. Jas, editors. *Josquin des Prez: The sources*. The Collected Works of Josquin des Prez, 1. Koninklijke Vereniging voor Nederlandse Muziekgeschiedenis, Utrecht, 2013.

The dataset |

Vocal model	Voices	I	Vocal model	Voices	I
Absalon fili mi	4	1	*Missa Hercules Dux Ferrarie*, Sanctus–Pleni sunt–Osanna	4, 2, 4	1, 1, 1
Benedicta es, pt. 1–3	6, 2, 6	4, 3, 3			
Fecit potentiam	2	1	*Missa Pange lingua*, Benedictus	2	1
In exitu Israel de Egypto, pt. 1–3	4	1, 1, 1	*Missa Sine nomine*, Cum sancto	4	1
Inviolata, pt. 1–3	5	1, 1, 1	*Qui belles amours*	4	2
Memor esto verbi tui, pt. 1–2	4	1, 1	*Adieu mes amours*	4	3
Pater noster, pt. 1–2	6	2, 2	*Comment peult avoir joye*	4	1
Preter rerum seriem, pt. 1–2	6	3, 2	*Faulte d'argent*	5	1
Qui habitat in adjutorio, pt. 1–2	4	2, 2	*Je ne me puis tenir*	5	3
Stabat mater, pt. 1–2	5	2, 1	*La plus des plus*	3	1
Missa De beata virgine, Cum sancto–Credo–Crucifixus	4, 5, 5	2, 1, 1	*Mille regretz*	4	2
			Plus nulz regretz	4	1
Missa Faysant regretz, Qui tollis–Sanctus–Pleni sunt–Osanna	4, 4, 3, 4	1, 1, 1, 1	*Si j'ay perdu*	4	1

CLI tool |

- An implementation of the mapping algorithm can be run as a **command line tool** called `TabMapper`
  - https://github.com/reinierdevalk/tabmapper/
- Researchers can use the tool to **make transcriptions themselves**, for their own purposes
  - A caveat is that, for reliable output, the tuple of input files must be **clean and correct** — which they are not always out of the box
- **Demo**: let's see how this works!

Thank you! |

- **Resources**
  - Paper: http://archives.ismir.net/ismir2019/paper/000051.pdf
  - Data: https://github.com/reinierdevalk/data/tree/master/josquintab/
  - CLI tool: https://github.com/reinierdevalk/tabmapper/
  - Gerbode: https://www.lutemusic.org/
  - JRP: https://github.com/josquin-research-project/
- **Contact**

  - reinierdevalk@gmail.com; redv@moodagent.com
  - t.crawford@gold.ac.uk