2019

Music Information Retrieval for Irish Traditional Music Automatic Analysis of Harmonic, Rhythmic, and Melodic Features for Efficient Key-Invariant Tune Recognition

Pierre Beauguitte
Technological University Dublin

Recommended Citation

Beauguitte, P. (2019) Music Information Retrieval for Irish Traditional Music Automatic Analysis of Harmonic, Rhythmic, and Melodic Features for Efficient Key-Invariant Tune Recognition, Doctoral Thesis, Technological University Dublin. doi:10.21427/7hkk-p423

Music Information Retrieval for Irish Traditional Music

Automatic Analysis of Harmonic, Rhythmic, and Melodic Features for Efficient Key-Invariant Tune Recognition

by

Pierre Beauguitte

Supervisors: Dr. Bryan Duggan
Prof. John D. Kelleher

SCHOOL OF COMPUTER SCIENCE
Technological University Dublin

Thesis submitted for the degree of

Doctor of Philosophy

September 2019

Declaration

I certify that this thesis which I now submit for examination for the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate study by research of the Technological University Dublin and has not been submitted in whole or in part for an award in any other Institute or University.

The work reported on in this thesis conforms to the principles and requirements of the TU Dublin's guidelines for ethics in research.

TU Dublin has permission to keep, to lend or to copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature: ______________________ Date: ______________

Acknowledgements

Thank you to all my fellow PhD students in the TU Dublin School of Computer Science, Lucas, Jack, Eoin, Elizabeth, Ivan, Karim, Hao, Fei, Vihanga, Abhijit, Filip, Annika, Giancarlo, Niels, Shane, Andrei, Paul, Mariana, and Senja, as well as the postdoctoral researchers, Hector, Thibaut, Caroline, and Alex. Thanks also to the staff members, lecturers and researchers in TU Dublin: Sarah Jane Delany, Robert Ross, Paul Doyle, Patricia O'Byrne, John Butler, David Carroll, Denise Murray, and of course to my supervisors Dr. Bryan Duggan and Prof. John D . Kelleher. Thank you also to all the researchers I met along the way, in conferences and workshops in Ireland and abroad.

I am grateful to Emmanouil Benetos, Sean O'Leary, and Aggelos Pikrakis, for accepting to be, respectively, my external, internal, and transfer examiners.

I had the pleasure of taking part in the traditional music ensemble of the TU Dublin Conservatory of Music and Drama, and I want to thank its directors Odhrán Ó Casaide, Kieran Hanrahan and Tom Doorley, as well as Peter Browne, and the members of the ensemble, in particular Mark Redmond, Fionn Ó hAlmhain and Róisín Ward Morrow. Playing Irish music with such talented people was a privilege, and a thoroughly

enjoyable experience.

Thank you to my friends, Jérôme, Servane, Lucie, John, Sam and all the Lalala choir, Camille, Damien, Romain, Carine, Sarah, and Magnus.

Thank you to my family, my parents Annie and Jacques, and Virginie, Laurent, Isabelle, Lise, and Nicolas.

Finally, I want to thank Astrid, for her unflinching support and infinite patience. I have been extremely lucky to have her accompany me in this journey through the music of Ireland, and carry me through the toughest passages.

Abstract

Music making and listening practices increasingly rely on technology, and, as a consequence, techniques developed in music information retrieval (MIR) research are more readily available to end users, in particular via online tools and smartphone apps. However, the majority of MIR research focuses on Western pop and classical music, and thus does not address specificities of other musical idioms.

Irish traditional music (ITM) is popular across the globe, with regular sessions organised on all continents. ITM is a distinctive musical idiom, particularly in terms of heterophony and modality, and these characteristics can constitute challenges for existing MIR algorithms. The benefits of developing MIR methods specifically tailored to ITM is evidenced by Tunepal, a query-by-playing tool that has become popular among ITM practitioners since its release in 2009. As of today, Tunepal is the state of the art for tune recognition in ITM.

The research in this thesis addresses existing limitations of Tunepal. The main goal is to find solutions to add key-invariance to the tune recognition system, an important feature that is currently missing in Tunepal. Techniques from digital signal processing and machine learning are used and adapted to the specificities of ITM to extract harmonic and temporal features, respectively with improvements on existing key detection methods, and a novel method for rhythm classification. These features are then used to develop a key-invariant tune recognition system that is computationally efficient while maintaining retrieval accuracy to a comparable level to that of the existing system.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The widespread use, and constant improvement, of mobile technology is profoundly influencing the practice and enjoyment of music. One consequence of the growth of the smartphone market, in the past decade, is that apps based on techniques from the field of music information retrieval (MIR) have become popular with end users. Examples of music-oriented apps abound: Shazam, which allows users to identify commercial audio recordings, counts more than 100 million monthly active users (MAU);[1] Yousician, an e-learning software released in 2014, reported 8 million MAU in 2016.[2]

Research in MIR has been motivated, in part, by the need for reliable and efficient ways of processing audio data, made evident by the increasing digitisation of music. For example, retrieving information from raw audio signals is essential to build music search systems or annotate large

[1] `https://9to5mac.com/2018/01/26/shazam-apple-acquisition/`, last visited August 26, 2019

[2] `https://youtu.be/nIH-Wr2mJWw?t=462`, last visited August 26, 2019

collections (Downie, 2003). This led to the organisation, in 2000, of the first International Symposium on Music Information Retrieval (ISMIR). The symposium became a conference 2 years after, and is still one of the most important academic events in the field.

Early in the history of MIR, Futrelle and Downie (2002) pointed out that most research focused on Western pop and classical music, and have highlighted the need for the field to expand to include other musical cultures. Tzanetakis et al. (2007) later coined the term *computational ethnomusicology*, and attempted to offer some guidelines for analysis of non-Western musics. In 2010, the first conference on Analytical Approaches to World Music (AAWM) was organised; the year after, an AAWM journal began, as well as an annual workshop on Folk Music Analysis (FMA). Researchers active in both AAWM and FMA have advocated in de Valk et al. (2017) for more collaboration between MIR researchers and digital music archives.

In particular, there is a growing literature on the use of technology for Irish traditional music (ITM), a musical tradition popular across the globe. Examples of such research projects include stylistic analysis of ITM recordings (Ali-MacLachlan et al., 2018; Köküer et al., 2019), composition of new tunes fitting the musical style with a generative recurrent neural network (Sturm et al., 2015; Sturm and Ben-Tal, 2018), and the creation of an ontology for ITM (Weissenberger, 2017).

The work presented in this thesis finds its place in this active landscape, and in particular around Tunepal, a query-by-playing tune recognition system for ITM (Duggan, 2010). Tunepal is now widely used among learners and practitioners of ITM, handling more than 20,000 queries per month (Duggan et al., 2016). Improving the retrieval accuracy of this system would not only benefit this user base, but also offer a better tool for annotating and exploring digital music archives. Currently, one of the main limitations of Tunepal is the fact that tune recognition is not key-invariant: a tune cannot be recognised if it is played in a different key to the one in which the corresponding score in the search space is written.

The main goal of this thesis is to propose computationally efficient solutions for key-invariant tune recognition in ITM. To this end, methods relying on techniques from digital signal processing and machine learning, and tailored to take the specificities of ITM into consideration, are developed. Both the harmonic and rhythmic dimensions of musical signals are considered, respectively with the tasks of key detection and rhythm classification. Tune recognition relies in part on automatic music transcription (AMT), a task in which significant progress has been made since the release of Tunepal. Using state-of-the-art AMT algorithms, and the new and improved MIR methods for ITM, a system for key-invariant tune recognition is proposed, with a focus on computational efficiency while maintaining retrieval accuracy to a comparable level to that of the existing system. This efficiency concern is particularly relevant given that Tunepal is typically used on mobile devices where the profile of computational resources can vary and may be limited.

1.1 Contributions

The first type of contribution to the field resulting from this thesis is the manual annotation of a number of corpora of ITM recordings. These annotations were used to evaluate the systems developed in the thesis. Furthermore, the annotations have been made publicly available, as has the code for the systems, and so the evaluations carried out in this thesis are replicable by other researchers. The annotations include:

- key annotations for the audio recordings of the Foinn Seisiún CDs (*FS*) published by Comhaltas Ceoltóirí Éireann and Grey Larsen's recordings, accompanying the book *300 Gems of Irish Music for All Instruments* (Larsen, 2013) (see Chapter 3);
- annotations on a collection of 500 recordings of tunes gathered from the *FS* collection and other online sources. Annotated fields contain the tune type, instruments recorded, tune id for the corresponding score on the website The Session, and transposition between the recording and the score. The choice of tunes in the collection was done to respect distributions of tune types across 4 reference collections (see Chapter 4 and 5);

- pitch track annotations for 30 tune recordings, made with the aid of the Tony software (Mauch et al., 2015) (see Appendix A).

The second type of contribution to the field resulting from this thesis is a set of novel methods tackling different tasks in MIR: key detection, rhythm inference, and key-invariant tune recognition. These novel methods are developed and evaluated in Chapters 3 through 6 of the thesis. More specifically:

- in Chapter 3, key detection is performed in a standard manner, by means of pre-defined key-profiles, and a set of new key-profiles adapted to ITM are introduced;
- a novel method for rhythm inference is introduced in Chapter 4, in which a logistic regression model is trained to predict a rhythm type from low-level audio features;
- a key-invariant tune recognition system is introduced in Chapter 5. New metrics are defined to deal with the specifics of this information retrieval system. Improvements on this system are proposed in Chapter 6 by taking into consideration the predictions from the models introduced in Chapter 4.

1.2 Publications arising from this thesis

The research contributions made to the field by the work reported in this thesis have been the basis for a number of international peer-reviewed publications. These include:

Chapter 3: Beauguitte, Pierre, Duggan, Bryan and Kelleher, John D., Key Inference From Irish Traditional Music Scores and Recordings. In *Proceedings of the 14th Sound and Music Computing Conference*, July 5-8, 2017, Espoo, Finland.

Chapter 4: Beauguitte, Pierre, Duggan, Bryan and Kelleher, John D., Rhythm Inference From Audio Recordings of Irish Traditional Music. In *Proceedings of the 8th International Workshop on Folk Music Analysis*, June 26-29, 2018, Thessaloniki, Greece.

Appendix A: Beauguitte, Pierre, Duggan, Bryan and Kelleher, John D., A Corpus of Annotated Irish Traditional Dance Music Recordings: Design and Benchmark Evaluations. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, August 7-11, 2016, New York City, USA.

Other publications made during the course of this research, but not explicitly referred to in this thesis, include:

- Beauguitte, Pierre, Duggan, Bryan and Kelleher, John D. (editors), *Proceedings of the 6th International Workshop on Folk Music Analysis*, 15-17 June, 2016, Dublin, Ireland.

- Beauguitte, Pierre, The AEPEM Collection: A Set of Annotated Traditional French Music Scores. In *Proceedings of the 7th International Workshop on Folk Music Analysis*, 14-16 June 2017, Málaga, Spain.
- Beauguitte, Pierre and Huang, Hung-Chuan, Content-based Music Retrieval of Irish Traditional Music Via a Virtual Tin Whistle. In *Proceedings of the 9th International Workshop on Folk Music Analysis*, July 2-4, 2019, Birmingham, UK.

1.3 Thesis summary and structure

The main body of this thesis is structured as follows.

Chapter 2 provides definitions and context for the work presented in the subsequent chapters. After defining concepts and terms from music theory useful for the present purposes, it focuses on the history and practice of ITM. Then a survey of related work in MIR and ITM is presented.

Motivated by a first tentative architecture for key-invariant tune recognition, Chapter 3 presents some improvements on the state of the art for key detection, on both audio and symbolic music.

After focusing only on the harmonic content of musical signal in Chapter 3, Chapter 4 focuses on the temporal dimension. Using low-level audio features and machine learning models, the proposed method predicts rhythm categories from an audio signal. A new dataset of annotations on ITM recordings is introduced.

In Chapter 5, the task of key-invariant tune recognition is approached with a new architecture, as the key detection method from Chapter 3 proved to lack robustness in this musical context. The new method, based on pitch class histogram alignment, is presented, new metrics are defined, and the performance of the proposed system is compared to existing methods.

In Chapter 6 an improved key-invariant tune recognition system is presented, which integrates the rhythm prediction system from Chapter 4 into the key-invariant tune recognition system from Chapter 5. The evaluation of this system indicates that the integration of rhythm predictions into the system improves the system in terms of both computational efficiency and retrieval accuracy.

Finally, concluding remarks and discussions of future work are given in Chapter 7.

The thesis also includes a number of appendices. Appendix A introduces a corpus of manual pitch track annotations of ITM recordings, along with a benchmark study of existing automatic music transcription algorithms. The work presented in this appendix represents preliminary work carried out during this research project that informed the selection of the automatic music transcription systems that are used in the present thesis. Appendix B gives 4 confusion matrices for the final experiments of Chapter 3. Finally, Appendix C gives URLs to the datasets and implementations realised in the context of the work presented in this thesis.

Chapter 2

Background

This chapter gives some definitions and context for the work presented in this thesis. Section 2.1 introduces some definitions and terminology from music theory. Section 2.2 is a short presentation of the history and current practice of Irish traditional music (ITM). Finally, relevant research in the field of music information retrieval (MIR) is presented in Section 2.3.

2.1 Elements of music theory

Some terms and concepts of music theory are introduced here. Even though ITM often does not follow the harmonic and tonal structures of Western pop and classical music, the theory and vocabulary are still useful for the discussion.

The atomic component of a melody is a *note*, that has both a pitch and a duration. Other properties such as loudness and timbre are not considered. The pitch of a note is its perceived frequency, that can be different from the physical frequency of the sound. In the 12-tone equal temperament (12-TET) used in Western music, the octave (*i.e.* the interval between a frequency f and its double $2f$) is split in 12 equal steps, called *semi-tones*. The MIDI scale offers a convenient representation of pitch as integer numbers, with the convention that the middle A, or A_4 (440 Hz) is represented by the number 69, and the unit is the semitone. Consequently the middle C, or $C4$, being 9 semi-tones below A_4, is represented by 60. In general, a pitch of fundamental frequency f is represented by the MIDI note:

$$p = 69 + 12 \times \log_2 \left(\frac{f}{440 \text{ Hz}} \right) \tag{2.1}$$

It is often useful to consider pitch *classes*, with respect to octave equivalence ($f \sim 2f$, or modulo 12 in MIDI notation). Pitch classes can be seen as elements of $\mathbb{Z}_{12}$, where 0 represents C.

Another useful unit of pitch is the *cent*, defined such that a semitone corresponds to 100 cents. The octave then corresponds to 1200 cents, and the number of cents between two notes at fundamental frequency a and b is:

$$n = 1200 \times \log_2 \left(\frac{b}{a} \right)$$

Melodies are sequences of notes. In a score, the notation support for

Western music, note durations are quantised as fractions of the beat: a crotchet represents one beat, a quaver half a beat, and so on. However, melodies can also be represented without any reference to a beat, for example as succession of notes in the format (pitch, onset, duration).

Monophony refers to the situation where only one melody is being played, by one or several players. Polyphony means that multiple melodies, or one melody and some accompaniment, are played simultaneously. Heterophony is an intermediary situation, where multiple players interpret the same melodic line, but not in the exact same manner. It can be described as a “complex” monophony.

Most of Western music is built around the diatonic scale, *i.e.* the arrangement of seven degrees separated by one or two semitones, as represented by the polygon on Figure 2.1. A *mode* is then determined by choosing a degree as a starting point, or *tonic*. Choosing *Do* results in the Ionian mode, with the intervals T-T-S-T-T-T-S (T and S standing for tone and semitone, respectively). *Re* results in Dorian, T-S-T-T-T-S-T. As there are 7 degrees in the diatonic scale, 7 such modes exist, but only the 4 used in Irish music are represented on Figure 2.1.

A mode can be either major or minor, depending on the nature of the third, that is the number of semitones between the first and third degrees of the mode. A major third has 4 semitones, a minor third only 3. Ionian and Mixolydian are major modes. Dorian and Aeolian are minor modes. The musical *key* of a piece is defined by both the tonic note and the nature of the third. The notation is simply the pitch class of the tonic followed by M for major or m for minor (*e.g.* *G*M, *B*♭m).

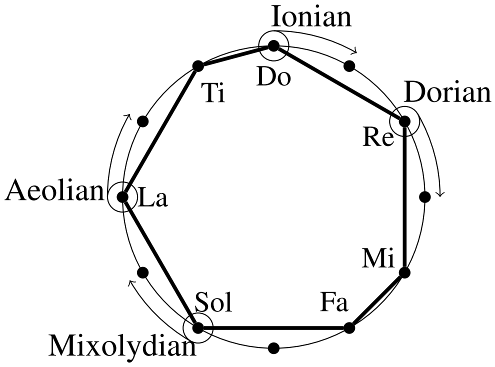


Figure 2.1
Circular representation of a diatonic scale, and the 4 modes used in ITM

2.2 Irish traditional music

The history of ITM has been the subject of a large number of books and publications already (Ó Canainn, 1993; Breathnach, 1996; Ó hAllmhuráin, 2004; O'Shea, 2008; Vallely, 2011), so this section does not intend to provide a comprehensive overview of the topic, but rather focuses on a few aspects relevant for the work presented in this thesis.

2.2.1 Tunes, types and structure

The nature of the music is melodic and modal, as opposed to the more common harmonic and tonal aesthetics of Western pop and classical music. In common with similar aurally transmitted musical traditions, ITM is subject to variations and ornamentation in its interpretation. The melodies, or *tunes*, are usually short and consist of two or sometimes more repeated parts. Each part is typically 8 bars long, and repeated twice in most tunes.

A large part of ITM was played to accompany dances. If nowadays the music is more often played on its own than to accompany dancers, the dance types are still present as the different tune categories. The most popular are reels, jigs (of which there exists different types: double, single, slip, hop), hornpipes, slides, polkas, and rarer types include highlands, strathspeys, mazurkas, waltzes, and barndances. The majority of melodies follow simple rhythmic structures, mostly consisting of quaver movements.

In the first volume of the collection *Ceoil Rince na hÉireann* (1963), collector, author, and musician Breandán Breathnach lists the typical tempi for different tune types. These are given in Table 2.1, along with the quaver duration these tempi imply, as well as the duration of a tune part (8 bars) in the corresponding time signatures. On a set of manual annotations of tunes (Köküer et al., 2019), the duration of notes is measured to be between 80 and 220ms 90% of the time, with a mode at 130ms. Compared to the values listed in Table 2.1, these are much shorter, indicating faster tempi.

2.2.2 Collections

Collections of Irish music have been published since the eighteenth century, with the work of John and William Neale, *A Collection of the Most*

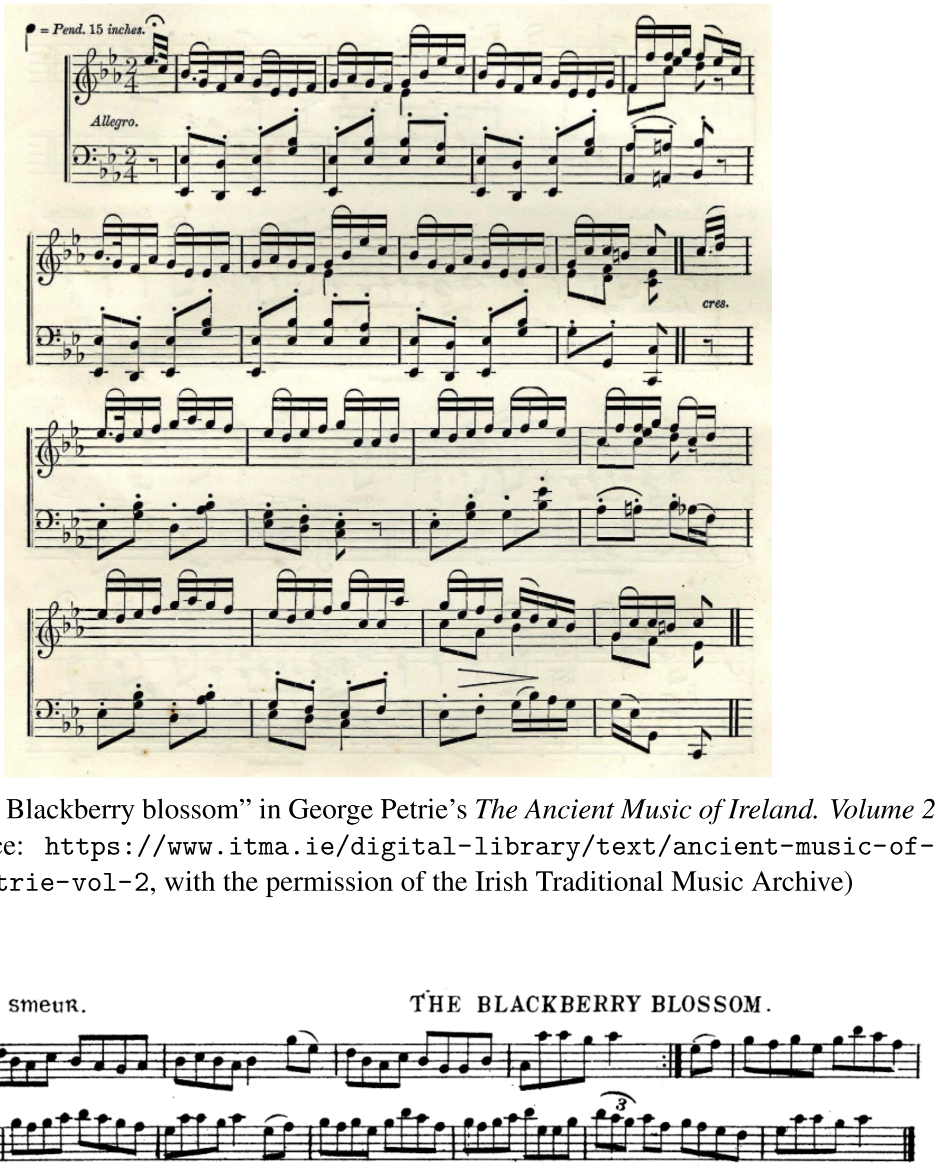
tune type	tempo (BPM)	quaver (s)	8 bars (beats)	8 bars (s)
double jig ($\frac{6}{8}$)	♩. = 127	0.157	16	7.56
single jig ($\frac{6}{8}$)	♩. = 127	0.157	16	7.56
hornpipe ($\frac{4}{4}$)	♩ = 180	0.333	32	10.67
slip jig ($\frac{9}{8}$)	♩. = 144	0.417	24	10
reel ($\frac{4}{4}$)	♩ = 224	0.268	32	8.57

Table 2.1
Durations in ITM

Celebrated Irish Tunes, published in 1726. At the turn of the century in 1792, a harp festival was held in Belfast, where Edward Bunting was appointed musical scribe. This led to the publication, in 1796, of his *General Collection of the Ancient Irish Music*. Later, in the middle of the nineteenth century, George Petrie published in *Ancient Music of Ireland* (1855) the melodies he had collected from musicians around Ireland.

Concerning both Bunting and Petrie, it has been noted (Ó Canainn, 1993; O'Shea, 2008) that their collection process was both selective (in that it purposefully left out some airs that, according to the collector, did not belong to the collection) and incorporative (in that it included pieces that originated from other neighbouring traditions).

Classical notation was used to write the melodies, thus discarding any microtonality or non-tempered scales that native musicians might have used. Tomás Ó Canainn notes "the tendency of most collectors to change modal tunes into minor airs, with a sharpened seventh, thereby changing their whole character" (Ó Canainn, 1993, p. 18). Melodies were also arranged with piano accompaniment, that stemmed from the musical knowledge of the collector in classical harmony, but did not



(a) Reel “The Blackberry blossom” in George Petrie’s *The Ancient Music of Ireland. Volume 2* (1882) (source: `https://www.itma.ie/digital-library/text/ancient-music-of-ireland-petrie-vol-2`, with the permission of the Irish Traditional Music Archive)

(b) Reel “The Blackberry blossom” in Francis O’Neill’s *The Dance Music of Ireland* (1907) (source: `https://imslp.org/wiki/Special:ReverseLookup/259251`)

Figure 2.2
Two notations of the reel “The Blackberry Blossom”

belong to the musical idiom of native musicians.

The collection of tunes *The Dance Music of Ireland* published by Francis O’Neill in 1907 is different in this regard: melodies were written without accompaniment or indications for interpretation, and was aimed at an audience of musicians already familiar with the Irish musical idiom, and thus able to make “an informed interpretation using an individual choice of phrasing, embellishment and variation from a familiar stylistic palette” (O’Shea, 2008, p.20). By contrast, Bunting and Petrie had arranged the melodies to suit more classical ears. Figure 2.2 shows two notations of a same reel, “The Blackberry Blossom”, in George Petrie’s *The Ancient Music of Ireland - Volume 2* (1882) and in Francis O’Neill’s *The Dance Music of Ireland* (1907). Key and metre are different, but more important is the piano arrangement in the former, applying classical tonal harmony to an Irish melody.

A more recent collection of tunes is Breandán Breathnach’s *Ceoil Rince na hÉireann*, a series of 5 books published between 1963 and 1999. With the advent of the Internet in the early 1990s, online collections have appeared. A notable one is that released by Heinrik Norbeck,[1] first published in 1997 and regularly updated, for a current total of over 2500 tunes.

Lastly, The Session[2] is a very popular website hosting discussions about ITM, listings of regular sessions (see below Section 2.2.6) around the world, and a user-curated collection of tunes, comprising more than 30,000 settings, or variants, of more than 15,000 tunes. Some of these tunes clearly originate from outside Ireland: for example, a tune called *The Crested Hens* is listed as one of the most popular waltzes of the collection, when it is actually a French *bourrée*, a dance much faster than

[1] `https://norbeck.nu/abc/`

[2] `https://thesession.org`

the waltz, composed by hurdy-gurdy player Gilles Chabenat. However the melody was recorded as a waltz by the Irish-American band Solas on their eponymous record in 1996, and was subsequently appropriated by ITM practitioners. Some musicians also add their own compositions to The Session's tune collection. However challenging this is to the notion of *traditional* repertoire, it is not fundamentally different from the incorporative nature of historical collections discussed above in this section. Because it is the most comprehensive collection of tunes, it will be used as a reference in Chapter 5.

2.2.3 Notation

The collections mentioned above used classical musical notation, thus ignoring some subtleties of intonation or microtonality, which are used in ITM (Molloy, 2017; Vallely, 2011). In particular, the third and seventh degree of the scale are often mobile, giving a modal colour to the music. In some major tunes for example, the seventh is natural when approached from above, but flat when from below. These nuances were often discarded by collectors. The increasing use of tempered instruments (see below Section 2.2.4) also led to ITM being closer to well tempered music.

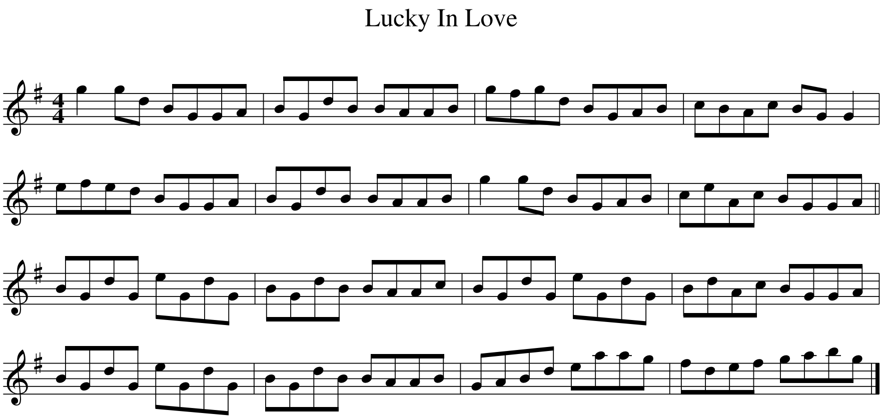
Other systems of notation have been used by ITM practitioners, often as mnemonic and teaching devices. Tablatures are common for fretted string instruments as well as accordions. O'Shea (2008) also mentions an old ABC notation for fiddle resembling a tablature: the letters ABCD indicate which finger to use, and the numbers 1234 represent the string to bow.

A very popular standard today is the ABC notation, first invented in the USA (Williams, 2013), then formalised and developed by Chris Walshaw.[3] Its appeal relies to a large extent on it being a text-based format, both machine- and human-readable. A large number of collections of Irish traditional music have been digitised in that format, and are used by Tunepal, as presented below in Section 2.3.5. An example is given of Figure 2.3. The header of a tune contains a list of fields including title, rhythm, metre, key (`T, R, M, K` respectively). The `L` fields defines the length of a note. In the example given on Figure 2.3 it is an eighth note, or quaver. Following the header is the tune itself, represented in a straightforward syntax. One letter indicates one note (here a quaver, as defined by the `L` field). A number following a letter multiplies its length: g2 is a quarter note. Casing indicates the octave: upper case notes are in the first octave of the instrument, lower case in the second octave. Other conventions allow the representation of all other necessary indications such as ornaments, repeats, and alterations.

[3] http://abcnotation.com

```
T: Lucky In Love
R: reel
M: 4/4
L: 1/8
K: Gmaj
g2 gd BGGA | BGdB BAAB | gfgd BGAB | cBAc BG G2 |
efed BGGA | BGdB BAAB | g2 gd BGAB | ceAc BGGA ||
BGdG eGdG | BGdB BAAc | BGdG eGdG | BdAc BGGA |
BGdG eGdG | BGdB BAAB | GABd eaag | fdef gabg |]
```

(a) ABC notation



(b) Standard notation

Figure 2.3
Two notations of the reel "Lucky in love"

2.2.4 Instruments

This section presents briefly the main instruments found in ITM. The list aims at being comprehensive, but has no pretension of being exhaustive. Unless other works are cited, Vallely (2011) is the source for the information.

The harp, if not a very common instrument in sessions and among ITM practitioners in general, is perhaps the most emblematic of the tradition, and of Ireland itself, of which it is the national symbol. Harps have been present in Ireland since at least the 11th century. Older instruments used strings made of brass or iron. Modern Celtic harps are strung with gut or nylon. They are in general fully chromatic thanks to semitone levers.

The traditional Irish flute is a side-blown wooden flute with open tone holes. Some are equipped with keys that can make the instrument chromatic. This kind of flute started to be used in Ireland only in the 18th century, when their popularity was also rising in Europe. Boehm system flutes remain somewhat marginal in ITM.

Whistles have been found to be present in Ireland as early as the 12th century. Tin whistles are now very popular, and played by a large number of ITM practitioners, even though it might not be their main instrument. It is a six-holes fipple flute, most often in *D* but existing in all other keys. In the second half of the 20th century, musicians and makers experimented to create *low*-whistles in lower registers. They have since become quite widespread.

Different types of bagpipes have been present in Ireland since at least the 11th century. Existing representations indicate that they were mostly mouth-blown and equipped with 2 drones, similarly to other bagpipes in Europe. It is only at the beginning of the 18th century that a distinct Irish kind of bagpipe appears, the uilleann pipes or union pipes. In the current shape of the instrument, the chanter (melody pipe) has a register of 2 octaves, larger than other bagpipes. The bag is inflated using bellows. It has a set of drones (usually 3) and a set of *regulators*, allowing for chordal accompaniment. An interesting specificity of this type of bagpipe is the ability to play *staccato*, thanks to a closed or “tight” fingering.

The violin, or *fiddle* in the context of ITM, is one of the most popular instruments. Although older types of bowed string instruments have been found in Ireland, dating as far back as the 11th century, the modern fiddle is the standard orchestra violin originating from Italy in the 16th century. The style of playing differs from classical playing by both the bowing technique and the use of non-tempered scales or sliding notes. O’Shea (2008) notes that the preference for the term “fiddle” over “violin” is a recent phenomenon, tied to the revival and commodification of ITM.

The bodhrán is a frame drum, found in its modern form in Ireland from the 1830s. It is held with one hand inside the frame, sometimes pushing on the skin to adjust its pitch, and played with a stick held in the other hand.

The concertina is a six-sided free reed instrument of English and German origins. It became popular in Ireland in the second half of the 19th century. Contrarily to most accordions, both hands are used for playing melody, and there is no bass or chord accompaniment.

The banjo is a plucked string instrument, with a frame-drum-like resonator. Its origins are African American, and Irish musicians that had emigrated to the USA began using it for ITM at the turn of the 20th century. Modern instruments are fretted, and often played with a plectrum, but older ones were fretless and strummed with the fingers.

The guitar, the mandolin and the bouzouki are plucked string instruments descending from the lute, and integrated in ITM relatively recently, in the 20th century. The latter is originally from Greece, and was first used in ITM only in the mid 1960s. Musicians have devised alternative tunings more suitable to Irish music accompaniment (DADGAD for guitar, GDAD for Irish bouzouki).

Finally, modern drum set and piano have been used in *céilí* dancing, a form of traditional dancing revived by the Gaelic League at the end of the 19th century.

Voice is present in Irish music, with a few different genres and techniques. *Sean-nós* ("old-style" in Irish) singing is a solo and ornamented style of singing. Tomás Ó Canainn claims that "no aspect of Irish music can be fully understood without a deep appreciation of sean-nós singing" (Ó Canainn, 1993, p. 49). However, these songs do not fall into the definition of dance tunes (Section 2.2.1) and are therefore out of the scope of this thesis.

Ballads and folk songs are common in Irish music. Lyrics can be in English or Irish language, and sometimes a mix of both in *macaronic* songs. Some are written in a dance-like rhythm, like the song *Rocky Road to Dublin* which has a $\frac{9}{8}$ time signature like a slip jig. Again, this repertoire does not fall into the dance tunes category, and are not in the scope of this thesis.

Dance tunes can be sung with a technique called *lilting*, or vocalisation of melodies using non linguistic syllables, sometimes referred to as “diddley-dee”. Lilting can be used as a memory aid for learning a melody, or as a substitute for instruments. Although this could be included in a study of dance music of Ireland, the work in this thesis is limited to instrumental music, and lilting is not represented.

2.2.5 Keys and tonalities

Some accounts of earlier ITM practice suggest that the keys in which tunes were played were not fixed. For example, Breathnach (1996) claims that *D* was used to denote the bottom note of the uilleann pipes, flute or whistle, regardless of the actual pitch. Ó Canainn (1993) also indicates that fiddle players would detune their instrument so that the *D* string matches the bottom note of the pipes. That note depended on the length of the chanter, until the Taylor brothers, pipes makers in Philadelphia, standardised the concert pitch pipes, in *D*, at the end of the 20th century.

Most of the instruments used in ITM are limited to a certain key, or a small number of keys. Some, like the concertina, keyed flute or keyed pipes are fully chromatic, but these notes are used as accidentals or to borrow from other keys rather than to play in entirely different keys. It is common for musicians to own different instruments allowing them to play in different keys.

Sets of tunes are constructed in a way that most often accommodates these limitations of the instruments, in that the tunes of a set will generally use the same tone set, even if they modulate between different modes (*e.g.* *E* Aeolian, *G* Ionian and *A* Dorian all use the same notes). It is interesting to compare this to the sets or *medleys* in the fiddle music tradition of Cape Breton in Canada, where often the mode changes while the fundamental remains the same (Doherty, 1996). This means that the tone set changes, which is perfectly feasible on a chromatic instrument like the violin.

According to Fintan Vallely, "it is the rule-of-thumb practice to assign a tune the key corresponding to its ending note" (Vallely, 2011, p. 388). However, Breandán Breathnach states that "many airs do not close on the expected key note", and "that by the ending or final note is not simply meant that last note of an air as it would appear in a transcribed version, but the final note of rest or repose on which the melody can be fittingly brought to a close" (Breathnach, 1996, p. 8). From this it appears that defining unambiguously the key of a tune is not straightforward. The modal nature of ITM blurs even more the notion of key, as the 4 different modes used in the repertoire are built from the same tone set (*e.g.* *G* Ionian, *A* Dorian, *D* Mixolydian, and *E* Aeolian with *G* as Do). Furthermore, many tunes only use 6 or even 5 degrees of the scale, adding more ambiguity.

2.2.6 The *session*

Vallely defines a session as “a loose association of musicians who meet, generally, but not always, in a pub to play an unpredetermined selection, mainly of dance music, but sometimes with solo pieces such as slow airs or songs” (Vallely, 2011, p. 610). This phenomenon is relatively recent: it began in the middle of the 20h century in the United Kingdom when Irish workers gathered in pubs to play together, as “a means of celebrating Irishness as a shared difference” (O’Shea, 2008, p. 43). Since then, sessions have become global: The Session has a register of 1859 regular sessions,[4] predominantly in Europe and North America, but also on the other continents. Sessions can be more or less open to newcomers, and the ratio of professional and amateur musicians can vary a lot. Frequently, the publican pays a few session leaders, who decide on the rules of the session. A process of commodification, increased by the growing tourism industry in Ireland, is often discussed, as the example of session in Doolin, County Clare, discussed in O’Shea (2008). This performative aspect of some sessions goes against the claim made by Williams that “Irish music does not require an audience” (Williams, 2013, p. 10) .

When played in a session, Irish music can often be adequately qualified as heterophonic (Williams, 2013). All players of melodic instruments (typically greater in number than rhythmic and harmonic instruments) play the same tune together, but the result is often far from unison

[4] website visited on July 7th, 2019

for several reasons. First of all, different instruments can play the same melody in different octaves (*e.g.* flute and tin whistle). Additionally, due to the acoustic limitations of certain instruments, or as an intended variation, some notes of a tune can be played in a different octave. The low B (B_3) for example cannot be played on most traditional flutes. Consequently flute players often play a B_4 instead, while a banjo player would play the note B_3 and a whistle player would play an octave higher than the flute, B_5. Yet, all would be considered as playing the same tune. Another important aspect is the amount of variations present in Irish music. Because of personal or regional stylistic differences, the abundance of different sources (notations, archive or commercial recordings), and of the music being transmitted aurally (thus relying on the memory of the musician and therefore subject to interpretation), many different versions of a same tune may exist. Although musicians will often try to adapt to each other in order to play a common version during a session, it is not uncommon to hear some differences in the melodies. Finally, tunes are almost always ornamented differently by each individual musician depending on their style and personal preferences.

2.2.7 Recording and ITM

The advance of recording technology has influenced ITM practice over the last century, and this section illustrates some important aspects of this relationship.

The first commercial recordings of Irish music were made in the USA in the 1920s, notably by 3 fiddlers known as The Sligo Masters, Michael Coleman, James Morrison and Paddy Killoran, who had emigrated from Ireland and settled in New York. It had a widespread success in Ireland, and had a notable effect on practice: many musicians copied the sets and style of the disc, and this led to a blurring of regional differences.

The physical limitations of the recording format has dictated, to an extent, the structure of the sets of tunes: 2 or 3 tunes played *segue*, repeated 2 or 3 times each (O'Shea, 2008; Vallely, 2011). It is worth comparing this with other traditions, such as old-time music in the USA, where tunes can be repeated up to 50 or 60 times (Williams, 2013), or Cape Breton fiddle music, where tunes in a medley are repeated only once or twice (Doherty, 1996).

More recently, O'Shea (2008) reports the attitude of some practitioners that defend their repertoire, and oppose to being recorded in sessions for fear of having their tunes "stolen". Similar tensions were revealed in Su and Duggan (2014): it describes a live-monitoring system, where music played at a session is analysed, and the list of recognised tunes is automatically published on a website. Some musicians disliked the intrusiveness of the system, and feared its possible consequences on the practice. The commodification of the session, now an essential part of the tourism industry in Ireland, multiplies the occurrences of unwanted and/or unauthorised recordings of live music.

2.3 Related work

This section gives a concise literature review of MIR research relevant to the topic of this thesis. Section 2.3.1 gives an overview on the task of Query by Singing/Humming and defines the task of tune recognition. This discussion will lead to the question of musical key identification, which is the focus of Section 2.3.2. Then, Section 2.3.3 presents existing work aiming at rhythm classification. Section 2.3.4 presents some MIR research focusing specifically on ITM. Finally, Section 2.3.5 presents the Tunepal app, which is the basis on which most of the work presented in this thesis builds upon.

2.3.1 Query by Singing/Humming, tune recognition

The ability to search a digital music library by content, *i.e.* by playing or singing a melody excerpt, as opposed to textual search by metadata (such as title or artist name), has been a core motivation of MIR research since the early establishment of the field (McNab et al., 1996; Lemström and Perttu, 2000; Downie, 2003).

Query by Singing/Humming (QbSH) is one of the tasks defined by the Music Information Retrieval Evaluation eXchange (MIREX)[5] initiative. Created in 2005, MIREX aims at fostering research and development in MIR by defining precise tasks, establishing evaluation metrics, and providing evaluation datasets. Although the metrics are not suited

[5] https://www.music-ir.org/mirex/wiki/MIREX_HOME

to all MIR research, especially when applied to music genres outside of the Western pop and classical music canon, they remain a standard in the field. A common practice is to report both the MIREX metrics as well as others, tailored to the specific study case (Serrà et al., 2010; Benetos and Holzapfel, 2015). Other tasks and metrics defined by MIREX will be discussed where relevant in the remainder of this thesis.

2.3.1.1 Problem definition

In the QbSH task, a query is an audio recording of a sung or hummed melody excerpt, and the search space consists of either symbolic representations (MIDI, ABC, or equivalent) or audio recordings of pieces of music. A typical approach is to transcribe the query into a symbolic representation using an automatic music transcription (AMT) algorithm, as well as the search space items if they are in audio format. Then, measures of melodic similarity are used to find the item closest to the query.

Companies such as ACRCloud[6] or SoundHound[7] have made QbSH tools commercially available. The popular app Shazam[8] does not allow sung or hummed queries, and in fact addresses a task known as audio fingerprinting, where the actual audio signal of a recording is recognised.

The audio-to-audio QbSH situation is similar to another MIREX task, audio cover song identification. In such cases, some approaches do not

[6] https://www.acrcloud.com/music-recognition

[7] https://www.soundhound.com

[8] https://www.shazam.com/

make use of an intermediate symbolic representation, but perform the matching on lower-level audio features, like chromas (Ellis and Poliner, 2007; Serrà et al., 2010; Salamon et al., 2013; Lee et al., 2018).

In the task of QbSH, queries are audio recordings, but other types of queries are sometimes considered. Pikrakis et al. (2016) addresses the symbolic-to-audio situation: queries are in a symbolic format (MIDI), and searched within automatic transcriptions of audio recordings. A gesture-to-audio paradigm is proposed in Wallace (2018), where a query is a pitch contour traced by moving a smartphone in the air, recorded via the accelerometers of the device, and the search space consists of automatic transcriptions of a collection of recordings of Norwegian folk music.

The task of tune recognition addressed in this thesis corresponds to an audio-to-symbolic situation, where the query is an audio recording and the search space comprises melodies in symbolic format (ABC, introduced in Section 2.2.3). Because queries are played on an instrument rather than sung or hummed, the term query-by-playing is used.

An important feature of QbSH is key-invariance, *i.e.* the ability to identify a query regardless of possible transpositions. Stasiak (2014) proposes a transposition invariant algorithm, but its alignment method relies on the assumption that the queries start at the beginning of melodies. As will be explained below in Section 2.3.5, this restriction is not appropriate for the problem tackled in this thesis. Using intervals instead of pitch in the symbolic representation does achieve key-invariance, but has the effect of amplifying the cost of errors or variations in the query (Janssen et al., 2017). The symbolic-to-symbolic method proposed in Martiniano and Silla (2017) relies on knowing the key of the query, which is not readily available in the case of tune recognition as defined above. Finding an efficient and robust way to perform key-invariant tune recognition for ITM is the main motivation for the work in this thesis. The proposed methods will be presented in Chapters 5 and 6.

2.3.1.2 Automatic music transcription

When an intermediate symbolic representation is used for QbSH, an automatic music transcription (AMT) algorithm is used. AMT is most often concerned with either monophonic or polyphonic music, and Benetos et al. (2013) claim that the problem can be considered solved for the former case, although new methods are still proposed (Kim et al., 2018). For polyphonic music, AMT methods generally attempt at extracting all the notes from the polyphonic signal, but there also exists a task called melody extraction, aiming at finding only the main melodic line. A survey of such approaches is given in Salamon et al. (2014).

The result of an AMT algorithm can be either a continuous pitch track, giving an estimate pitch for each audio frame, or a set of notes with onset, pitch, and duration. Both can be useful for QbSH, depending on the melodic similarity measure chosen. The following list introduces some state-of-the-art algorithms in AMT and melody extraction.

pYIN (Mauch and Dixon, 2014) stands for probabilistic YIN, and is based on the frequency estimation algorithm YIN (de Cheveigné and Kawahara, 2002), used in conjunction with HMM-based pitch tracking. The initial algorithm returns frame-level estimates, but an additional segmentation step based on HMM modelling of note events was introduced in (Mauch et al., 2015).

Silvet (Benetos and Dixon, 2012) is based on Principal Latent Component Analysis. Although it is designed for polyphonic music transcription, obtaining a single melody track is achievable by simply limiting the number of notes occurring at any time to one. First, a pitch track is generated by factorising the spectrogram according to predefined templates. This is then post-processed with HMM smoothing, in a similar manner to the pYIN segmentation step. This approach has a high computational cost due to the complexity of spectrogram factorisation.

Melodia (Salamon and Gómez, 2012) first extracts a salience function by detecting peaks in the time/frequency representation of the audio signal and then extracts the best continuous pitch track possible. The original algorithm aims at extracting the predominant melody from a polyphonic signal. A variant named PitchMelodia was later developed for monophonic signals.

A benchmark study of the performance of these AMT algorithms on ITM recordings is given in Appendix A.

2.3.1.3 Melodic similarity

Once an audio query has been transcribed into a symbolic representation via an AMT algorithm, it can be compared to items in the search space by using a measure of melodic similarity. How to quantify this musical notion in the form of a computable formula is a complex question, and a number of such metrics and distances have been proposed. Janssen et al. (2017) offers a comparative study of the performance of such measures for retrieving folk songs from short segments. Chen et al. (2018) raises the issue that no single similarity measure can capture all aspects of melodic similarity, and addresses it by using a Similarity Network Fusion (SNF), a method that automatically combines several metrics.

2.3.1.4 Evaluation metrics

In MIREX, submissions to the QbSH task are evaluated with a single metric, the "Top-10 hit rate (1 point is scored for a hit in the top 10 and 0 is scored otherwise)."[9] A Best (or Top-1) hit rate, more interesting from a user perspective, is sometimes also reported (Salamon et al., 2013; Stasiak, 2014).

Another metric that is often reported for QbSH tasks is the Mean Re-

[9] `https://www.music-ir.org/mirex/wiki/2019:Query_by_Singing/Humming`

ciprocal Rank (MRR) (Dannenberg et al., 2007; Salamon et al., 2013). It is defined as:

$$\mathrm{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{2.2}$$

where $rank_i$ is the rank of the correct item in the ordered result set for query i, where the candidates are sorted in order of descending melodic similarity to the query. In some situations, it can occur that several candidates obtain the same similarity score, in which case their ordering in the result set is purely incidental. It is thus important to clarify the definition of rank in case of draws. In order to tackle this problem, Martiniano and Silla (2017) use the Mean draw Reciprocal Rank (MdRR), defined as

$$\mathrm{MdRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{d_i \times r_i} \tag{2.3}$$

where r_i and d_i refer to the rank and the number of draws in the rank of the correct tune.

Stasiak (2014) defines a new metric for evaluating a QbSH system as:

$$\delta = \frac{1}{N}\sum_{i=1}^{N}\frac{E_2^{(i)} - E_1^{(i)}}{E_1^{(i)}} \tag{2.4}$$

where the sum is computed only over the N successful queries of a batch, and $E_1^{(i)}$ and $E_2^{(i)}$ are the distance scores, in that case computed by Dynamic Time Warping (DTW) of, respectively, the first and second closest matches to query i.

As discussed above in Section 2.3.1.1, a way of performing key-invariance in QbSH is to know the key of the query. In order to use this approach in the audio-to-symbolic situation this thesis is concerned with, this requires inferring the key from the audio recording. This task is the subject of the next section.

2.3.2 Key inference

The standard key-finding algorithm, based on key-profiles is first introduced in Section 2.3.2.1. Then, existing key-profiles are given in Section 2.3.2.2.

2.3.2.1 Key-finding algorithm

The standard approach to identifying keys in a musical piece is to use key-profiles (Temperley, 2001). They can be seen as vectors assigning weights to the twelve semitones, denoted $(p[i])_{i=0,\dots,11}$. As there are 12 semitones, and a key can either be major or minor as discussed in Section 2.1, a total of 24 keys exist. Each one of them has its own key-profile, but only one needs to be defined for a given tonic (C by convention) in both major and minor keys, as transposition to another tonic note is performed by rotating the elements in the vector. Key-profiles are then normalised to sum to 1:

$$\overline{p}[i] = \frac{p[i]}{\sum_{i=0}^{11} p[i]} \tag{2.5}$$

A histogram $(h[i])_{i=0,\ldots,11}$ of cumulative durations of each pitch class in the musical excerpt is generated, and the score is the weighted sum of the histogram with the key-profile:

$$s(\overline{p},h) = \sum_{i=0}^{11} \overline{p}[i] * h[i] \tag{2.6}$$

The estimated key is then the one corresponding to the highest scoring profile:

$$key(h) = key(\arg\max_{p \in \mathscr{P}} s(\overline{p},h)) \tag{2.7}$$

where $\mathscr{P}$ is the set of 24 key-profiles representing candidate keys. The normalisation step in Equation 2.5 ensures that there is no bias towards either major or minor keys. Indeed, if the sum $\sum p$ is higher for one of the two key-profiles, major for example, then scores s would be unfairly higher for major keys than for minor ones. Because $\sum \overline{p} = 1$, this bias does not occur in estimating the key in Equation 2.7.

Other methods for key-identification are based on higher-level features. For example, in Madsen and Widmer (2007) the intervals of a melody are analysed, which presupposes that an automatic transcription of the signal has been performed beforehand. In Noland and Sandler (2006), a Hidden Markov Model (HMM) is trained to estimate the key from a sequence of chords.

Baseline key-profiles are introduced in the next section. The computation of pitch class histograms is detailed afterwards in Section 3.1.1.

2.3.2.2 Existing key-profiles

Triads. Certainly the most naive way to define a key-profile is to consider only the triad of the tonic chord. For example, in C it is expected that the pitch classes of the tonic C, the third E and the fifth G will be the most frequent, as reflected in the key-profiles:

$$p_{\text{triad}}(C\text{M}) = [1,0,0,0,1,0,0,1,0,0,0,0]$$

$$p_{\text{triad}}(C\text{m}) = [1,0,0,1,0,0,0,1,0,0,0,0]$$

Krumhansl-Kessler. The key-profiles established in Krumhansl (1990) were obtained by perceptual experiments, in contrast with the triads presented above which were motivated by musical theory. Subjects were asked to rate how well the different pitch classes fit within a tonal context established by a short musical excerpt. The Krumhansl-Kessler key-profiles are a well known method for key detection.

$$\begin{aligned} p_{\text{KK}}(C\text{M}) = [&6.35,2.23,3.48,2.33,4.38,4.09,\\ &2.52,5.19,2.39,3.66,2.29,2.88] \end{aligned}$$

$$\begin{aligned} p_{\text{KK}}(C\text{m}) = [&6.33,2.68,3.52,5.38,2.60,3.53,\\ &2.54,4.75,3.98,2.69,3.34,3.17] \end{aligned}$$

Lerdahl's Basic Spaces. The *basic spaces* defined in Lerdahl (1988) are derived from the diatonic scale of each key. Different weights are given to the degrees of the scale: 5 for the tonic (index 0 for CM and Cm), 4 for the fifth (index 7 for CM and Cm), 3 for the third (index 4 for CM, 3 for Cm), 2 to the rest of the diatonic scale, and 1 to the remaining semitones.

$$p_{\text{Lerdahl}}(C\text{M}) = [5,1,2,1,3,2,1,4,1,2,1,2]$$

$$p_{\text{Lerdahl}}(C\text{m}) = [5,1,2,3,1,2,1,4,2,1,2,1]$$

It is worth noting that the natural minor scale, or Aeolian scale, is considered here: the natural seventh (index 10) is taken as part of the scale, not the augmented seventh (index 11) as would be the case with the harmonic minor scale, more commonly used in tonal music.

Leman's Tone Centre Images. In Leman (1995), the *simple residue image* (or R-image) of a chord is generated as a weighted combination of the undertone series of the tonic. The *tone centre images* are then derived by summing the R-images of the chords present in the common cadences, or sequences of chords used to establish tonal centers. The three typical cadences selected in Leman (1995) are

$$\left\{\begin{array}{cccc} \text{I} & \text{IV} & \text{V} & \text{I} \\ \text{I} & \text{II} & \text{V} & \text{I} \\ \text{I} & \text{VI} & \text{V} & \text{I} \end{array}\right.$$

where the type of the chord depends on the scale considered. For the major key-profiles, the usual major scale (Ionian) is used. However, for the minor ones, the harmonic scale is chosen, where the seventh degree is one semitone higher than in the natural scale.

The Tone Centre Images (TCI) are then obtained by summing the R-images of the chords, weighted by how often they occur in the cadences, and normalising:

$$6 * \mathrm{I} + 3 * \mathrm{V} + \mathrm{II} + \mathrm{IV} + \mathrm{VI}$$

After normalisation, the key-profiles obtained are:

$$\begin{aligned} p_{\text{Leman}}(C\text{M}) = [&0.36, 0.05, 0.21, 0.08, 0.24, 0.21, \\ &0.05, 0.31, 0.07, 0.24, 0.09, 0.10] \\ p_{\text{Leman}}(C\text{m}) = [&0.34, 0.11, 0.15, 0.25, 0.11, 0.25, \\ &0.02, 0.31, 0.24, 0.09, 0.12, 0.14] \end{aligned}$$

2.3.2.3 Evaluation metric

The evaluation metric for the MIREX audio key detection task is defined as follows: let k be the ground truth annotation, and $\hat{k}$ the estimated key, then the accuracy score for this item is:

$$acc = \begin{cases} 1 & \text{if } k = \hat{k} \\ 0.5 & \text{if } \hat{k} \text{ is the perfect fifth of } k \\ 0.3 & \text{if } \hat{k} \text{ is the relative of } k \\ 0.2 & \text{if } \hat{k} \text{ is the parallel of } k \\ 0 & \text{otherwise} \end{cases}$$

These scores are then averaged across the dataset used. The MIREX task focuses on audio only, but the same metric can be used for symbolic representation.

Chapter 3 will present a study of key inference in ITM, using the method, key-profiles, and metric introduced in this section, as well as improvements tailored to ITM.

The next section discusses existing work on rhythm classification. This is of interest for the task at hand since, as discussed above in Section 2.2.1, tunes in ITM are categorised according to dance types, each having specific rhythmic characteristics.

2.3.3 Rhythm classification

Most of the existing literature on rhythm classification, for either symbolic or audio representation of music, are considering time signatures, or metres, as categories.

Brown (1993) is an early example of using an autocorrelation function (ACF) to determine the metre of a piece of music from its score. Decision criteria on the ACF are explicitly defined. Also focusing on symbolic music, Toiviainen and Eerola (2006) use discriminant function analysis to predict the metre of folk tunes. More recent research on metre detection from written music include De Haas and Volk (2016) using statistical analysis of inter onset intervals (IOI), and McLeod and Steedman (2017) using a context free grammar (CFG).

Pikrakis et al. (2004) and Fouloulis et al. (2013) determine the metre of Greek traditional music recordings, including asymmetric metres, by hand-crafted decision criteria or template matching on an auto similarity matrix. In Gainza and Coyle (2007), the time signature is also detected using self-similarity matrix, but the method is based on a prior knowledge of the tempo. The method presented in Gouyon and Herrera (2003) relies on beats extracted in a semi-automatic manner, and uses hand-crafted decision criteria to infer the metre. Gainza (2009) and Varewyck et al. (2013) first extract the beats from the raw audio, then determine the metre by analysing inter-beat similarity.

None of the work cited in this section has focused on ITM. As was discussed above, the fact that Irish tunes are readily categorised into dance types with specific rhythms makes this musical genre suitable for rhythm classification. Chapter 4 will present a novel method for rhythm classification in ITM, which will then be used in Chapter 6 to improve the proposed key-invariant tune recognition system.

2.3.4 MIR and ITM

Duggan et al. (2009) presents the algorithm MATT2, which is at the core of Tunepal, described below in Section 2.3.5. Its performs tune recognition, as defined in the previous section, by using an AMT algorithm on the audio query and substring edit distance as a measure of melodic similarity. Besides its use in Tunepal, MATT2 has also been applied in Duggan et al. (2008) in order to detect changes of tunes in a set. Su and Duggan (2014) describe an attempt to use this tune segmentation algorithm for real time monitoring of sessions. Automatic structural analysis, including tune separation but also the detection of parts within a tune, is tackled in Kelly et al. (2010) by means of cross-correlation.

In order to perform tune recognition, the transcription of audio queries in Duggan et al. (2009) does not aim at retaining all the subtleties of the playing, but rather simplifies the transcribed melody to make it closer to a written score. On the contrary, Jančovič et al. (2015) aims at finely transcribing all the ornaments in Irish flute recordings. The corpus annotation project presented in Köküer et al. (2019) is a necessary endeavour for such studies. In its current state of development, 79 tunes are manually transcribed.

Ali-MacLachlan et al. (2015) relies on similar fine manual annotations of recordings to train models able to recognise musicians from their style of playing. In more recent iterations of this project, models are trained directly on the spectrogram of the recordings, thus removing the need for detailed annotations (Ali-MacLachlan et al., 2018).

Martiniano and Silla (2017) present a study of symbolic-to-symbolic tune recognition, where key-invariance is achieved using key annotations. Queries are created from excerpts of the scores themselves. An interesting finding is that queries taken randomly in the middle of tunes are retrieved better than those taken at beginning or end positions. The results are first sorted by their order of melodic similarity, measured by edit distance, then draws are filtered using a model predicting the genre (or type as described in Section 2.2.1) of the tunes.

2.3.5 Tunepal

In an Irish session, most of the times the names of the tunes are not announced in advance. This leads to a very common situation when a player, hearing a tune he or she does not know, decides to learn it, but can at best record the audio on a phone. Building a repertoire in this manner is very difficult, as unlabelled recordings are hard to organise, and searching for different versions or recordings of a tune without knowing its name is hardly doable. For these reasons, the query-by-playing app Tunepal (Duggan and O'Shea, 2011) was warmly welcomed when it was released in 2009. Tunepal allows a user to record an excerpt of 12 seconds of Irish music, either from a live performance or a commercial recording, and finds the tune in a large collection of transcriptions. The user has then access to the name and score of the tune, as well as pointers to available recordings of it. Today Tunepal handles more than 20,000 queries each month (Duggan et al., 2016). It is the state of the art for the task of ITM tune recognition. In this section, the general architecture of Tunepal, represented on Figure 2.4, is explained.

The first step of the process is to record an audio excerpt and transcribe it into a symbolic sequence. The duration of the recording is set to 12 seconds, which roughly corresponds to the length of one part of a tune (see Table 2.1). The AMT algorithm used is called MATT2, for Machine Annotation of Traditional Tunes. It determines the pitch of each frame by looking at the harmonicity of the power spectrum. Successive frames of the same pitch class are then grouped into notes. The length of a quaver is computed based on the distribution of note durations and is used to quantise the sequence at quaver granularity. Pitch classes are then represented using the ABC convention.

A database of target tunes is created by aggregating ABC transcriptions from several sources, including both historical collections and collaborative online repositories.[10] Normalisation of the ABC strings is done by expanding repeated sections; expanding long notes as quavers; removing metadata such as title, type, metre; removing bar marks and ornaments; and finally representing all notes by their pitch class, by uppercasing all notes. After this normalisation process, a tune is represented by a string of letters `A` to `G`.

A similarity score between the transcription t and each tune c in the corpus is then calculated as

$$s(t,c) = 1 - \frac{d(t,c)}{|t|}$$

where $|t|$ is the length of the transcription in quavers, and d is the substring edit distance (SSED), a simple modification of the Levenshtein distance (Navarro and Raffinot, 2002). Tunepal then returns the highest scoring tunes, which can be several versions of a same tune from different online sources, or tunes that show some melodic similarities with

[10] The complete list is available at `https://tunepal.org/index.html`

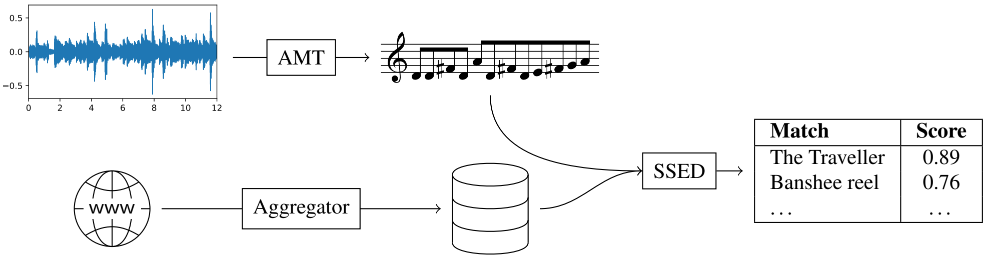


Figure 2.4
Architecture of Tunepal

the one being searched.

Currently Tunepal can only recognise tunes that are played in the same key as the ABC transcription present in the database. This is a problematic limitation, as playing ITM on instruments tuned differently from the norm is not uncommon, as discussed in Section 2.2.5. Allowing key-invariant tune recognition in Tunepal is the main goal of this thesis, and the next chapter will explore a first attempt to address this limitation.

Chapter 3

Key detection

As discussed above in Section 2.3.5, the main motivation for this thesis is to develop a robust and efficient method for key-invariant tune recognition, in order to address an important limitation of Tunepal, the state-of-the-art query-by-playing system for ITM. In a previous study on Irish tune recognition, Martiniano and Silla (2017) propose to solve key-invariance by transposing the query and all the tunes in the search space to a common key. However, the queries they consider are obtained from written tunes, so that the ground truth key annotations are consistent between queries and search space. In a tune recognition system like Tunepal, the key of the (audio) query is not readily available. Thus, in order to solve key-invariance by this common-key transposition method, the key of the query has to be inferred from the audio query itself. Figure 3.1 presents a possible architecture for key-invariant tune recognition based on this method.

This chapter will assess the feasibility of such a system, by focus-

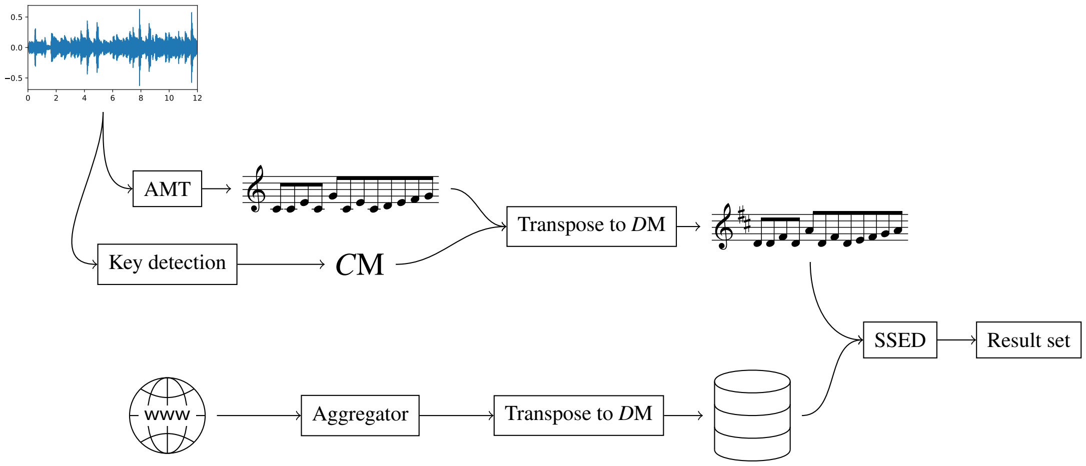


Figure 3.1
Tentative architecture for key-invariant tune recognition

ing on the problem of key detection. The standard method for this task, presented in Section 2.3.2.1, is used here, and the existing key-profiles introduced in Section 2.3.2.2 are used as baselines. In Section 3.1, the details of the pitch class histogram computations chosen for this experiment are given, as well as a refinement of the standard MIREX evaluation metric. The dataset, consisting of both audio and symbolic music, is presented in Section 3.2. A first experiment is presented in Section 3.3, in which new key-profiles are defined, taking into consideration some specificities of ITM. In Section 3.4, a parametric model is defined, and trained on the dataset. The content of this chapter is an extended and corrected version of Beauguitte et al. (2017). The annotations dataset and the implementation for the experiments carried out here are publicly available.[1]

[1] `https://github.com/pierrebeauguitte/keydetection`

3.1 Method

The standard algorithm for key detection has been presented above in Section 2.3.2. It relies on the computation of pitch class histograms, which can be carried out in different ways. Section 3.1.1 gives the details of the method chosen here. Section 3.1.2 addresses an issue with the standard metric as defined above in 2.3.2.3.

3.1.1 Pitch class histogram computation

The key identification method only needs a pitch class histogram (PCH) from the musical excerpt.

In the case of symbolic representation, obtaining this is straightforward. The software `abc2midi`[2] is used to parse the ABC files into a sequence of notes $((\text{pitch class}_t, \text{onset}_t, \text{duration}_t))_{t=1,\dots,T}$, where durations are expressed relative to a quaver. The PCH h is obtained by:

$$h[i] = \sum \{\text{duration}_t,\ \text{where pitch class}_t = i\} \tag{3.1}$$

For each audio recording, a chromagram is first generated, then the chroma vectors are summed over time. Chromagrams are obtained using the `madmom`[3] library (Böck et al., 2016). Several methods of computing the chromas have been tested: standard pitch class profile, harmonic pitch class profile (Gómez, 2006), and Deep Chroma extractor

[2] `http://abc.sourceforge.net/abcMIDI/`

[3] `http://madmom.readthedocs.io`

(Korzeniowski and Widmer, 2016). This last method, using a deep neural network trained to extract chromas from a spectrogram, consistently outperformed the others. Consequently all the results reported below are obtained with the Deep Chroma method.

An important difference between the PCH obtained from the audio recordings and symbolic representations is the presence of harmonics in the audio. In the symbolic case, only the actual pitch of the notes contribute to the symbolic PCH, whereas harmonics in the audio will also contribute to the PCH.

3.1.2 Performance metric

In situations where a draw occurs, *i.e.* when a same score s is obtained for different key candidates, the metric introduced in Section 2.3.2.3 is not properly defined, as the $\arg\max$ operator in Equation 2.7 does not define a unique key candidate. Here, in such cases, the key candidate obtaining the lowest accuracy score is kept as the estimated key. This choice was made to make the evaluation as strict as possible.

3.2 Dataset

3.2.1 Audio datasets

Two sets of recordings are used in this study, representing overall 636 audio items. Each tune was manually annotated with key information by the author, corroborated when possible by scores available on The Session.

Foinn Seisiún

This collection consists of recordings accompanying the Foinn Seisiún books published by Comhaltas Ceoltóirí Éireann, an organisation for the promotion of the music of Ireland, founded in 1951. They offer good quality, homogeneous examples of the heterophony inherent to an ITM session. Instruments in the recordings are representative of the typical Irish session, as presented above in Section 2.2.4. The whole collection consists of 3 CDs, representing 327 tunes. The first 2 CDs (273 tunes) are available under a Creative Commons Licence, while the third is commercially available.[4] In five instances, two recordings of a same tune are present. In four cases, it was decided to keep both as different items in the dataset, since the set of instruments recorded is different. Only in one case is the exact same recording present, in which case one of them was discarded. In the end, this dataset contains 326 distinct recordings, and is denoted FS_{audio}.

Grey Larsen's *300 Gems of Irish Music*

Grey Larsen's recordings, accompanying the book *300 Gems of Irish Music for All Instruments* (Larsen, 2013), is a set of MP3 files commercially available. They consist of studio quality solo recordings of tunes

[4] `https://comhaltas.ie`

played on Irish flute, tin and low whistles, and concertina. This dataset is denoted GL_{audio}, and consists of 300 unique recordings.

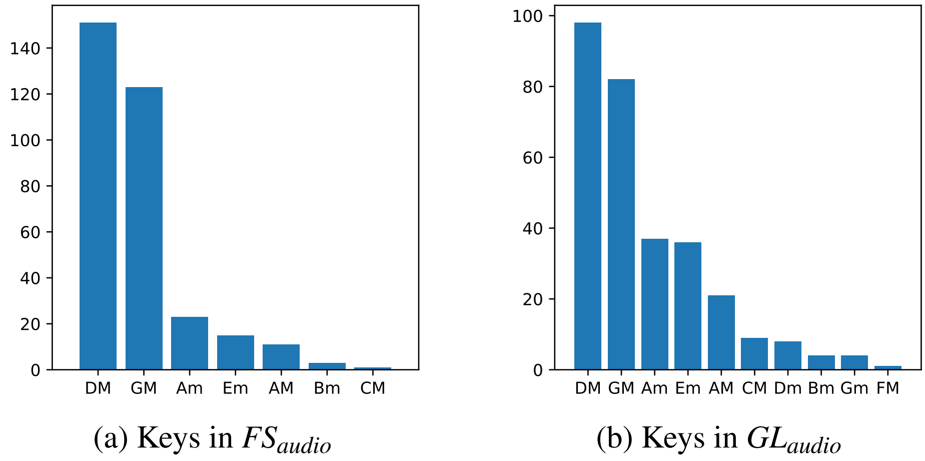
3.2.2 Symbolic datasets

For each tune in both audio corpora, a symbolic transcription was collected in ABC format. The majority of the transcriptions were found online, mostly on The Session (see Section 2.2.2). A small number of tunes were not available, in whose cases the audio recordings were manually transcribed to ABC by the author. As discussed above in Section 2.2.3, the symbolic transcriptions do not correspond exactly to the music played in the audio recording, and are rather outlines of the melodies being played, ignoring ornaments and small variations. The difference between recordings and scores is even clearer for the session recordings: the audio signal is then heterophonic, as the different musicians are not playing exactly the same melody.

This time, all redundant copies of duplicate tunes present in the Foinn Seisiún collection are discarded, as in such cases the score remains the same even though the recording differs. Hence FS_{symb} contains 322 items, and GL_{symb} 300.

3.2.3 Distribution of keys

Both datasets are unbalanced in terms of key distribution, as can be seen on Figure 3.2. These distributions are actually representative of the ITM



(a) Keys in FS_{audio}

(b) Keys in GL_{audio}

Figure 3.2
Distributions of keys in *FS* and *GL*

repertoire (Vallely, 2011). The keys of *D* and *G* are indeed the most common in sessions, in part due to the fact that some instruments are limited to these scales (*e.g.* keyless flute, whistle, uilleann pipes).

3.3 Experiment A: new key-profiles

In this section two new pairs of key-profiles are introduced, and their performances are compared to the baseline ones.

3.3.1 Modal basic spaces

Lerdahl's basic spaces, introduced in 2.3.2.2, are based on the natural scales, or Ionian mode for major and Aeolian mode for minor. In ITM, two other modes are commonly used: the Mixolydian mode (major with a minor seventh) and the Dorian mode (minor with a major sixth). The two basic spaces are modified so that they are suited to both major modes

(Ionian and Mixolydian) and minor modes (Aeolian and Dorian). This is done by setting $p[10] = p[11]$ in the major key-profile, and $p[9] = p[8]$ in the minor one:

$$p_{\text{Lerdahl}^\star}(C\text{M}) = [5,1,2,1,3,2,1,4,1,2,2,2]$$

$$p_{\text{Lerdahl}^\star}(C\text{m}) = [5,1,2,3,1,2,1,4,2,2,2,1]$$

This idea of considering the minor and major seventh as equivalent has already been used for the task of Irish traditional music tune transcription and recognition in Duggan (2009).

3.3.2 Cadences in ITM

These key-profiles are inspired by Leman's tone centre images. As mentioned in 2.3.2.2, cadences play an important role in establishing a tonal centre. In Irish traditional music, the most common cadence is I - IV - V - I (Vallely, 2011). In the case of minor tunes, the chord sequence VII - VII - I - I, often used in accompaniments, is also considered. Consequently the formulae to obtain the key-profiles are

$$\begin{cases} \text{Major:} & 2*\text{I}+\text{IV}+\text{V} \\ \text{Minor:} & 4*\text{I}+2*\text{VII}+\text{IV}+\text{V} \end{cases}$$

Instead of considering R-images, chords are simply represented by

	FS_{audio}	FS_{symb}	GL_{audio}	GL_{symb}
Triad	0.873	0.792	0.671	0.707
KK	0.854	0.869	0.665	0.696
Lerdahl	0.887	**0.890**	0.689	0.774
Leman	0.848	0.829	0.646	0.666
Lerdahl*	0.893	**0.890**	0.711	**0.798**
Cadences	0.883	0.874	0.677	0.766
Cadences*	**0.902**	0.815	**0.713**	0.747

Table 3.1
MIREX scores for the 4 corpora using different key-profiles

their triads, as introduced in 2.3.2.2. The resulting key-profiles are:

$$p_{\text{Cadences}}(C\text{M}) = [3,0,1,0,2,1,0,3,0,1,0,1]$$
$$p_{\text{Cadences}}(C\text{m}) = [5,0,3,4,0,3,0,5,1,0,3,0]$$

Finally, these profiles are also modified to account for the Mixolydian and Dorian modes:

$$p_{\text{Cadences}^\star}(C\text{M}) = [3,0,1,0,2,1,0,3,0,1,1,1]$$
$$p_{\text{Cadences}^\star}(C\text{m}) = [5,0,3,4,0,3,0,5,1,1,3,0]$$

3.3.3 Results

Results are given for the seven pairs of key-profiles considered. All the MIREX accuracy scores are reported in Table 3.1.

Two observations can be made from this table. First, comparing the MIREX scores on the two symbolic datasets shows that inferring the key of the tunes in *GL* is harder than in *FS*. Second, on the *FS* collection, most key-profiles yield better results on the audio data than on the symbolic data. The opposite is true for *GL*. Hence it appears that inferring keys from heterophonic or polyphonic audio is easier than on monophonic recordings. An explanation for this is that the harmonic content is richer in heterophonic and polyphonic signals.

The new modal key-profiles introduced in Section 3.3 (Lerdahl* and Cadences*) outperform the existing key-profiles on all four datasets. Inspired by Lerdahl's original key-profiles which assign different weights to degrees of the scale (see Section 2.3.2.2) the next section will present an attempt at improving the performance of the Cadences key-profiles, which are the best performing ones on audio recordings, by assigning different weights to the tonic, third and fifth degrees in the triads uses to build the profiles, as presented above in Section 3.3.2.

3.4 Experiment B: parametric profiles

3.4.1 Methodology

The model proposed here is a parameterised version of the previously introduced Cadences profiles. The parameters considered are the three weights given to the three notes of the triads, denoted $W = (w_1, w_3, w_5)$ for the tonic, third and fifth respectively. Then, the following profiles can be derived from the cadences chosen in Section 3.3.2

$$\begin{aligned}
p_{\text{Cadences}(W)}(C\text{M}) &= [2w_1 + w_5, 0, w_5, 0, 2w_3, w_1, 0,\\
&\qquad w_1 + 2w_5, 0, w_3, 0, w_3]\\
p_{\text{Cadences}(W)}(C\text{m}) &= [4w_1 + w_5, 0, 2w_3 + w_5, 4w_3,\\
&\qquad 0, w_1 + 2w_5, 0, w_1 + 4w_5,\\
&\qquad w_3, 0, 2w_1 + w_3, 0]
\end{aligned}$$

The modal versions of these profiles, Cadences*(W) are obtained in the same manner as in Section 3.3.2.

In order to evaluate the performance of this parametric model, and to find an optimal set of parameters W, a grid search is performed. The process iterates across a three dimensional discrete space, called the grid, where each point defines a different set of weights W, and hence a different model. The MIREX key detection metric introduced in 3.1.2 is used to evaluate the models. Each parameter w_i is allowed to take integer values in $[1, g]$, where g is the size of the grid. Here g is a hyper-parameter of the grid search algorithm, as opposed to the parameters w_i of the models being evaluated.

Performing a grid search on a complete dataset is straightforward, but it merely returns a model fitted to the data, without any indication about how well it could generalise to unseen data. In order to assess the performance of the models on new data, it was decided to conduct the grid search inside a 10-fold cross validation, following Kelleher et al.

(2015). The dataset is first split into 10 equally sized subsets, called *folds*. Each fold is then, in turn, kept aside as a *test set*, while the grid search is performed on the rest of the dataset, then called *training set*. In each of these 10 experiments (one per fold), the resulting model is tested on the test set, and a confusion matrix is kept. The aggregate matrix is finally obtained by summing all 10 confusion matrices.

It is important to note that although a MIREX score can be computed from the aggregate matrix, it does not correspond to any single model. Indeed, the models obtained from each of the 10 iterations described above can all be different. The score is rather an indication of average generalisation power of parametric models created using a grid search to fit the parameters applied to different samples of data. Once this is ensured, one final grid search is run, this time on the whole dataset, to obtain one single model.

3.4.2 Results

The only hyper-parameter in this experiment is g, the width of the grid. A wide range allows a better fit on the training data, but poses a risk of overfitting it, resulting in poor performance on the test sets. The experiment was performed for g ranging from 2 to 10. The grid size $g = 3$, allowing the weights w_i to take values in $[1,2,3]$, gave the best performance, and is used for the following results.

Scores calculated from the aggregate matrices after the cross valida-

	FS_{audio}	FS_{symb}	GL_{audio}	GL_{symb}
Cadences(W)	0.891	0.873	0.706	0.779
Cadences$^\star(W)$	**0.908**	0.842	**0.723**	0.741

Table 3.2
MIREX scores computed from the aggregate matrices after cross-validation on the 4 corpora

tion on each of the four datasets are presented in Table 3.2. The models Cadences$^\star(W)$ outperform all other methods on the two audio corpora. However, the Lerdahl* key-profiles evaluated in Experiment A remain the best performing ones on the symbolic data (see Table 3.1). Consequently the rest of this section focuses on Cadences$^\star(W)$ on the audio datasets.

The result of the cross-validation method suggests that the models Cadences$^\star(W)$ generalise well to unseen audio data. In order to obtain one single model, a final grid search was then performed on the combined dataset $(FS + GL)_{audio}$. Grouping the two collections of audio recordings means that the profiles should perform well on both heterophonic and monophonic recordings. The weights obtained are $(3,1,2)$, corresponding to the intuition that the tonic and fifth are more important than the third, as in Lerdahl's basic spaces (Section 2.3.2.2). The resulting key-profiles are:

$$p_{\text{Cadences}^\star(3,1,2)}(C\text{M}) = [8,0,2,0,2,3,0,7,0,1,1,1]$$

$$p_{\text{Cadences}^\star(3,1,2)}(C\text{m}) = [14,0,4,4,0,7,0,11,1,1,7,0]$$

With these profiles, the MIREX scores are 0.901 on FS_{audio} and 0.730 on GL_{audio}, to be compared to the scores in Table 3.1. The lower score on *FS* is not unexpected: the grid search maximises the overall score across the combined audio collection, regardless of the scores on the individual collections. The overall MIREX score on the combined collection is 0.819, compared to 0.811 with the non-parametric Cadences* profiles.

The confusion matrices for these new key-profiles on the audio collections, and for the Lerdahl* ones on the symbolic datasets (on which they are still the highest scoring ones), are given in Tables B.1 to B.4, pages 159 to 162. Rows indicate the actual keys in the ground truth annotations, while columns indicate estimated keys. Keys that never occur in either the ground truth or the estimations are omitted.

The three types of errors taken into account in the MIREX evaluation metrics are highlighted in different shades of blue. Another error that occurs frequently in this experiment is between a major key IM and its supertonic IIm, or a minor key Im and its subtonic ♭VIIM. The term "neighbour" was chosen to designate this relationship between such closely related keys. In terms of modes, the scales of a Ionian (resp. Mixolydian) mode and its neighbour Dorian (resp. Aeolian) mode contain the exact same pitch classes. It is not rare in ITM that a tune labelled as one key to change its tonic centre for a few bars to the neighbour key, *e.g.* Ballydesmond Polka N. 3 shown on Figure 3.3, considered to be a tune in *G*M, but with a second part in *A*m. On both audio datasets, this

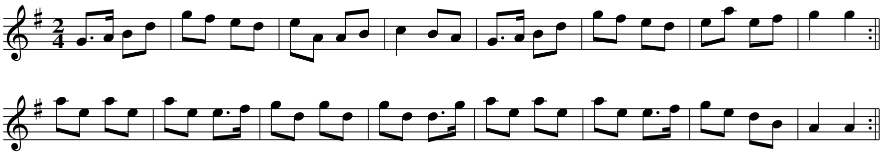


Figure 3.3

Ballydesmond Polka N. 3, a tune with first part in *G*M and second in *A*m

type of error is the most common. As such, and although the MIREX evaluation metric does not take these errors into account, reporting them seems relevant.

Relative keys are the next most common errors, on both audio and symbolic datasets. The scales of two relative keys contain, as is the case with neighbour keys, the same pitch classes, if one considers the Aeolian mode. Changes of tonic centre in a tune between its key and the relative key are also quite common in ITM. The high frequencies of these two types of errors can be explained by the specific characteristics of this musical idiom.

Table 3.3 gives the percentages of correctly inferred keys per mode for the best scoring methods, *i.e.* Cadences* on audio datasets and Lerdahl* on symbolic ones. A clear difference in performance appears between the major and minor keys, suggesting that minor keys are harder to detect than major keys. A possible reason for this is that many minor tunes have complete bars or sections borrowing to a neighbour or relative major key.

	FS_{audio}	FS_{symb}	GL_{audio}	GL_{symb}
Major	97.5%	91.1%	80.6%	82.5%
Minor	26.8%	58.5%	39.3%	64.0%

Table 3.3
Proportions of correct inference per mode

3.5 Discussion

Although the key-profiles introduced in the chapter managed to improve on the state of the art for key detection in ITM, the algorithm still fails in about 20% of cases overall, and even more on audio recordings. Therefore, the performance of the key-invariant tune recognition method presented in Martiniano and Silla (2017), or in Figure 3.1, relying on common-key transposition, would be undermined by the added difficulty on extracting the key from the audio query.

Before proposing a different way to tackle key-invariance in Chapter 5, the next chapter will focus on extracting rhythmic properties from audio recordings of tunes. A new dataset, designed to be more representative of ITM in terms of rhythm and tune types (see Section 2.2.1), will be introduced.

Chapter 4

Rhythm classification

The focus of the previous chapter was the harmonic content of music, both in symbolic notation and audio recording. This chapter now focuses on the temporal aspect, *i.e.* the rhythm of the music. Categorisation of tunes in terms of their rhythm, or dance type, has been done in all historical and modern collections, and is relatively unambiguous. Metres used in ITM, and the corresponding dance types, are:

- simple duple: $\frac{4}{4}$ (reel, hornpipe, fling, barndance) and $\frac{2}{4}$ (polka)
- simple triple: $\frac{3}{4}$ (waltz, mazurka)
- compound duple: $\frac{6}{8}$ (double and single jigs) and $\frac{12}{8}$ (slides)
- compound triple: $\frac{9}{8}$ (slip and hop jigs)

Simple and *compound* refer to the beat subdivision, while *duple* and *triple* refer to the grouping of beats. No asymmetric metres such as $\frac{5}{8}$ or $\frac{7}{8}$ are found in ITM. Rather than focusing on the metre, the aim here is to detect the tune type. Indeed, inferring a $\frac{4}{4}$ metre would not allow to differentiate between a reel and a hornpipe, although their rhythm is noticeably different, the latter typically using dotted notes (Vallely, 2011).

A new method, combining low-level spectral features and machine learning, is presented in Section 4.1. This approach is then used for two different categorisations: the first one attempts at distinguishing between duple and triple metres, while in the second one, the actual tune types are considered. A new dataset is introduced in Section 4.2. Results of the experiments are given in Section 4.3, and Section 4.4 gives some concluding remarks. Most of the content in this chapter has been published in Beauguitte et al. (2018), although the experiment is carried here on a different dataset. The annotations and the implementation of the novel method realised in the context of this chapter are publicly available.[1]

4.1 Method

The method introduced next relies on features extracted from the signal, called *quantised lag vectors*, presented in Section 4.1.1. Similarly to some of the work presented in Section 2.3.3, it is based on autocorrelation. A logistic regression model is then used to predict the rhythm category from these features, as explained in Section 4.1.2.

[1] `https://github.com/pierrebeauguitte/ITM_rhythm`

4.1.1 Feature extraction

The audio files are sampled at 44100Hz. A magnitude spectrogram is generated, with window size of 2048 and step size of 10ms, or 441 samples. Following Jehan (2005), this spectrogram is reduced, via a bank of triangular filters, to a 24-band Bark spectrogram $X_k(t)$ where $1 \leq k \leq 24$ is the Bark index. Then, following Bello et al. (2005), an onset detection function is obtained by a method of spectral difference:

$$SD(t) = \sum_{k=1}^{24} (H(X_k(t) - X_k(t-1)))^2 \quad \text{for } t > 0$$

where the rectifier $H(x) = (x + |x|)/2$ has the effect of ignoring decreases of energy, as it is equal to zero for negative values. As a consequence, it emphasises onsets more than offsets. As the energy difference is computed in each spectral band before being summed, changes in the harmonic content will appear in the *SD* function, and the presence of percussive instruments is not required to detect onsets.

The autocorrelation function is then computed on a 5-second window of the *SD* function $(w_t) = (SD(t_0 + t))_{0 \leq t < N = 500}$ (where t_0 is the start of the window) using Pearson's correlation coefficient. The autocorrelation for a lag l is:

$$ACF(l) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \text{ where } \begin{cases} X = (w_t)_{0 \leq t < N-l} \\ Y = (w_t)_{l \leq t < N} \end{cases}$$

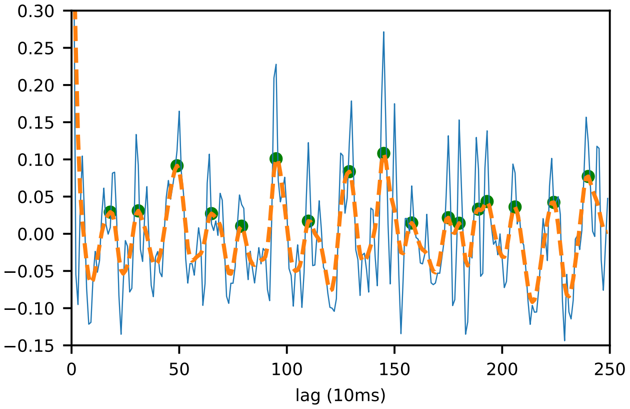


Figure 4.1
Peak picking of the ACF function. Solid line: ACF function. Dashed line: smoothed function

where cov is the covariance and σ designates standard deviation. This function is then smoothed by Gaussian filtering with a standard deviation of 20ms, and the local maxima of this smoothed curve are picked, ignoring the always present peak at $l = 0$. Figure 4.1 shows an example of the peak picking procedure on a window of a jig. Each peak p has a lag p_l and an amplitude p_a. For the goal of this study, what matters is not the actual locations of the peaks p_l, but their relative positions from each other. By abstracting the representation from the actual lag values, tempo invariance will be obtained. The quaver duration will be extracted from the peaks locations and then used to compute a *quantised* representation.

The quaver duration q is found by the *fuzzy histogram* algorithm, introduced in Duggan (2009), and given in Algorithm 1. The intervals, or lag differences, between the peaks are grouped into bins, allowing for a deviation of a fraction of the bin centre, set to $1/3$. The centres

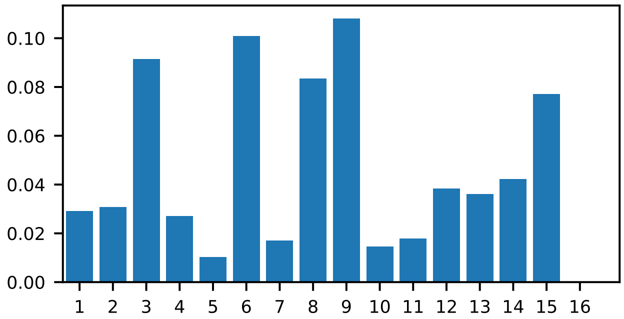


Figure 4.2
Quantised lag vector

of the bins are adjusted for each new interval added. Once all peaks are processed, the quaver length q is taken as the centre of the largest bin. This value will then be used to quantise the set of peaks P into a tempo-invariant representation.

This novel representation is dubbed *quantised lag vector*. Its elements $(ql_i)_{1 \leq i \leq 16}$ are obtained by first grouping the peaks as follows:

$$P_i = \{p \in P \text{ where } round(p_l/q) = i\}$$

and averaging across these sets:

$$ql_i = \begin{cases} \left(\sum_{p \in P_i} p_a\right) / |P_i| & \text{if } P_i \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

The number of 16 quavers was chosen empirically. Experiments with alternative values did not lead to significantly different results.

An example of such a vector is plotted in Figure 4.2, computed from the ACF peaks shown on Figure 4.1. The ratio of the first nine peaks is

Algorithm 1: Fuzzy histogram algorithm, adapted from (Duggan, 2009)

```
Data: P, list of peaks of the ACF (size l)
Result: quaver length
bins ← {};
max ← 0;
for $i \leftarrow 1$ to $l$ do
    if $i$ = $1$ then
        dur = $P[i]_l$;
    else
        dur = $P[i]_l - P[i-1]_l$;
    end
    found ← false;
    for b in bins do
        bin_start ← b.centre$*(1-1/3)$;
        bin_end ← b.centre$*(1+1/3)$;
        if $dur \geq bin\_start$ and $dur \leq bin\_end$ then
            found ← true;
            b.centre ← (b.centre ∗ b.count + dur)/(b.count + 1);
            b.count += 1;
            break;
        end
    end
    if found = false then
        newBin.centre ← dur;
        newBin.count ← 1;
        bins.add(newBin);
    end
end
for b in bins do
    if b.count > max then
        maxBin ← b;
        max ← b.count;
    end
end
return maxBin.centre;
```

preserved, but the absolute durations of the lags have been discarded, making this representation tempo-invariant. Some of the subsequent peaks are grouped together by the rounding operations. More prominent peaks appear at multiples of 3, as is to be expected from the compound metre of that tune type (jig).

Each 5-second window produces a 16-valued vector, and the window is slid with a step size of 0.5 second. Choosing such a small step size results in a large amount of examples, which is an advantage for the machine learning methodology presented in the next section.

4.1.2 Model training

Regression analysis in general attempts at modelling the relationship between independent variables x (here the *ql* vectors) and a dependent variable y (here the rhythm category). Logistic regression models, or classifiers, are used, because the dependent variables are categorical, *i.e.* they can only take one of a given set of values. A similar methodology will be used to predict, in a first experiment, the metre type, or beat subdivision and, in a second one, the tune type.

4.1.2.1 Experiment A: beat subdivision

The dataset consists of pairs (x,y), where x is a *ql* vector and y a label in $\{simple, compound\}$. A 10-fold cross validation methodology is used as a way of evaluating how well the models generalise (Kelleher et al., 2015). Each fold is, in turn, kept as a test set, and a binary classifier is trained on the remaining 9. When preparing the folds, all *ql* vectors from one tune are kept in only one of the folds. This way, the models will be tested on recordings that have not been used during training, thus avoiding a form of cheating.

To account for the fact that the classes *simple* and *compound* are not balanced in the dataset, during training the error on an instance is weighted by the inverse of the relative frequency of the output class of the instance in the training set; *i.e.*, errors on *compound* instances are given a higher weighting than errors on *simple* instances in the calculation of the loss function to account for the fact that *compound* instances are rarer.

4.1.2.2 Experiment B: tune type

In this second experiment, a model is trained to predict the tune type from the *ql* vector. As will be presented in the next Section, because some tune types are rare, using 10-fold cross validation would only result in too few examples of them in each fold. To avoid this problem, 4-fold validation is used instead. For each fold, a multinomial logistic regression classifier is trained in a *one-versus-all* manner, meaning that the model actually consists of a set of binary classifiers. As in experiment A, during the training phase, errors are weighted by the inverse of the relative frequency of the output class.

4.2 Dataset

As discussed in Section 2.2, the repertoire of Irish Traditional Dance Music is categorised in a number of distinct rhythms, or tune types, originating in the dances these tunes were accompanying. In order to build a dataset that is representative, in the relative proportion of the different tune types, of the body of ITM, 4 reference collections, mentioned in Section 2.2, are considered:

- Francis O'Neill's *The Dance Music of Ireland* (1907) (FON), is a collection of 1001 dance tunes, and considered to be "an important landmark for the traditional musician" (Ó Canainn, 1993, p. 20).
- Breandán Breathnach's *Ceoil Rince na hÉireann* (1963 - 1999) (CRÉ), is a series of 5 books comprising more than 1200 dance tunes.
- Henrik Norbeck's ABC collection[2] (HN) is a popular online tune book, published for the first time in 1997 and growing since, for a current total of over 2500 Irish tunes.
- The Session[3] is a collaborative website that offers among other things a large database of over 15,000 tunes of mostly, but not exclusively, Irish traditional music.

The first two collections in this list are historically significant, but

[2] https://norbeck.nu/abc/

[3] https://thesession.org

give a rather conservative view of the repertoire. On the contrary, the two online collections are more representative of the current repertoire in ITM. For example, neither CRÉ nor FON contain any barndances or waltzes, whereas these have gained popularity in Irish repertoire (Vallely, 2011, p. 739).

In addition to some categories of tunes being entirely absent from a collection, the grouping of types also differs between them. CRÉ and HN have only one category for slides and single jigs, The Session only has slides, and FON only single jigs. The question of the difference between these 2 types has been the subject of a number of online discussions.[4] Following CRÉ and HN, these 2 types will be considered here as a single group.

Because of their relative rarity, barndances, highlands, strathspeys and flings are also grouped in a single type labelled *Other 4/4*, and waltzes, mazurkas and "three-twos" in a single group labelled *waltzes*. Despite a slight heterogeneity, these categories are consistently duple simple and triple simple metres, respectively.

Set dances (present in all collections except The Session) were discarded, as this type does not constitute a homogeneous rhythmic category. Vallely writes:

> "Such tunes were often in 2/4 time ('The Blackbird', 'King of the Fairies'), sometimes 9/8 ('Is the Big Man Within?' [the

[4] for example `https://thesession.org/discussions/3854/`, last visited August 27, 2019

> second part of which is in 6/8], ‘The Barony Jig’), but generally resembled 6/8 double jigs (‘Orange Rogue’, ‘St Patrick’s Day’) or 4/4 hornpipes (‘Garden of Daisies’, ‘Job of Journeywork’).” (Vallely, 2011, p. 612)

Other discarded categories are airs, O’Carolan tunes (composed by Turlough O’Carolan, an Irish harper and singer from the late 17th and early 18th century), country dances, marches, slow airs and songs (in HN) and “Misc.” (in FON).

The distribution of tunes per type in all 4 collections is given in Table 4.1, in raw count in the top half and in percentages in the lower half. Apart from the absence of some categories in the two historical collections, rather large differences in distributions appear: FON contains similar numbers of reels and jigs, whereas CRÉ has about three times as many reels as it has jigs. Numbers for The Session are for the collection as it was on February 24th 2019[5].

A size of $N = 500$ recordings was chosen for the dataset. The distribution of tunes per type was chosen by averaging the percentages of Table 4.1 across the 4 collections, and taking $n_{type} = round(Np_{type})$. The numbers are given in Table 4.2, along with the resulting distribution of tunes in simple and compound metres.

The primary source of recordings is the Foinn Seisiún CD series. In

[5] weekly snapshots of all data from The Session are available at `https://github.com/adactio/TheSession-data`

	CRÉ	FON	HN	The Session
reels	646	350	1047	5834
jigs	229	365	506	4044
slides	110	45	119	324
slip jigs	55	45	98	523
hornpipes	82	150	149	1176
polkas	86	0	139	1120
other 4/4	0	0	85	1156
waltzes	0	0	27	1610
Total	1208	955	2170	15787
reels (%)	53.48	36.65	48.25	36.95
jigs (%)	18.96	38.22	23.32	25.62
slides (%)	9.11	4.71	5.48	2.05
slip jigs (%)	4.55	4.71	4.52	3.31
hornpipes (%)	6.79	15.71	6.87	7.45
polkas (%)	7.12	0	6.41	7.09
other 4/4 (%)	0	0	3.92	7.32
waltzes (%)	0	0	1.24	10.20

Table 4.1
Number of tunes per type in collections

Tune type	p_{type} (%)	n_{type}
reels	43.83	219
jigs	26.53	133
slides	5.34	27
slip jigs	4.27	21
hornpipes	9.20	46
polkas	5.15	26
other 4/4	2.86	14
waltzes	2.81	14
simple	63.86	319
compound	36.14	181

Table 4.2
Distribution of tunes in the dataset

addition, recordings from the Comhaltas Live program were gathered, mostly from the *session* category, but also from solo or concerts settings. The complete list with the URLs of the audio files, along with the relevant annotations, is publicly available.[6]

4.3 Results

This section presents the results of the 2 experiments. Accuracy scores are given for aggregate matrices resulting from the *k*-fold cross validation methodology described above.

The models of both experiments predict a label for a 5-second window. In addition to the window-level scores, predictions across a span of several consecutive windows are also considered. The reason for this is that rhythm is not as easily identifiable on all 5-second sections of a tune. Thus it is possible to reach better accuracy by gathering predictions on a longer segment. The prediction over a span of s windows is obtained by averaging the prediction probabilities for each class, and choosing the class that has the highest average probability. Performances are reported at window-level, over spans of s windows, and finally over whole tunes.

4.3.1 Experiment A

The aggregate confusion matrix resulting from the 10-fold cross validation is given on Table 4.3. The overall accuracy score at the 5-second

[6] `https://github.com/pierrebeauguitte/ITM_rhythm`

window level is 87.14%. The prediction accuracy is slightly lower on the *simple* class than on the *compound* class. A possible explanation for this is that there are more distinct tune types included in the *simple* class (reel, hornpipe, polka, waltz,...) than in the *compound* class (only jig, slip jig and slide). Looking at the score per tune type on Table 4.4, it appears that performance is lowest for waltzes. Reasons for this may be that the rhythms of such melodies are less repetitive, with more long notes and slower tempi, thus making periodicity harder to detect on the ACF function. Vallely writes: "as is the case with marches, popular Irish songs [...], Moore's Melodies [...] and some traditional song-airs [...] have typically been recruited for service as waltzes" (Vallely, 2011, p.739), suggesting the non-homogeneity of waltz melodies.

When considering spans of successive overlapping windows, the accuracy increases up to 95.11%, as is shown on Figure 4.3. It is possible to compute this only up to a span size of 55 windows, corresponding to the duration of the shortest tune in the collection.

Lastly, the tune-level accuracy is obtained by averaging the predictions over the span of all windows of its recording. The overall prediction accuracy is of 96.80%.

Although the task tackled in this first experiment is arguably easy, these high accuracy scores are very encouraging and suggest that the *ql* vector representation does capture some useful rhythmic information.

	simple	compound
simple	86.65	11.91
compound	13.35	88.09
simple	312	9
compound	7	172
accuracy (%)	97.81	95.03

Table 4.3
Aggregate confusion matrix at window-level (top) and tune-level (bottom) for experiment A (column: reference, line: prediction)

Type	Accuracy (%)
reels	93.93
jigs	93.62
slides	73.40
slip jigs	69.45
hornpipes	72.85
polkas	76.41
other 4/4	92.01
waltzes	63.18

Table 4.4
Window-level accuracy score per tune type for experiment A

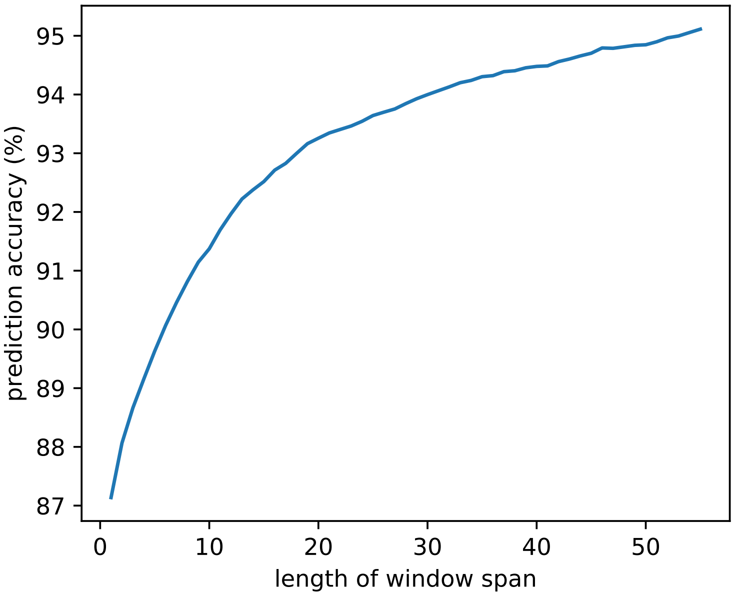


Figure 4.3
Prediction accuracy by span length for experiment A

4.3.2 Experiment B

The aggregate confusion matrix resulting from the 4-fold cross validation is given on Table 4.5. The overall accuracy score at the 5-second window level is 66.84%. The accuracy per window span length is shown on Figure 4.4, and reaches a maximum of 79.80% at $s = 55$.

Finally, the confusion matrix for tune-level prediction is given in Table 4.5. Overall accuracy reaches 83.2%. The scores are lowest on slip jigs, of which none are correctly classified, and slides. In both cases, the tunes are most often misclassified as jigs, which are also of compound metre: the prediction is still correct at the level of beat subdivision. However, slip jigs are in triple metre, but are mostly classified as jigs, that are in simple metre, suggesting that beat grouping is harder to capture than beat subdivision. A similar trend appears with waltzes (triple duple): 4 out of 14 of them are classified as hornpipes (simple duple).

4.3.3 Results on Greek music dataset

The method described in this chapter has been applied to a dataset of recordings of traditional Greek music during the 8th International Workshop on Folk Music Analysis (Thessaloniki, Greece, 2018).[7] The task consisted of predicting the metre, or time signature, of the pieces. A multinomial model was trained to predict the 7 different categories in-

[7] http://fma2018.mus.auth.gr/challenge.html

	reels	jigs	slides	slip jigs	hornpipes	polkas	other 4/4	waltzes
reels	79.98	3.77	6.85	6.32	4.56	15.59	32.37	7.28
jigs	2.20	81.52	42.90	45.19	8.37	1.62	2.90	7.61
slides	1.86	5.46	15.46	7.73	6.21	1.47	3.32	9.26
slip jigs	0.40	1.85	5.71	4.88	2.54	0.53	1.07	2.82
hornpipes	4.92	1.45	11.53	7.91	51.37	4.81	21.91	23.24
polkas	1.65	1.82	6.70	5.33	6.80	69.67	2.52	10.87
other 4/4	3.47	0.33	0.60	5.25	7.11	1.37	30.78	7.71
waltzes	5.53	3.80	10.24	17.38	13.04	4.94	5.14	31.22
reels	210	2	1	0	1	3	4	0
jigs	2	129	13	11	1	0	0	2
slides	0	1	6	2	1	0	0	0
slip jigs	0	0	1	0	1	0	0	0
hornpipes	3	0	1	0	33	0	2	4
polkas	0	0	2	0	1	23	0	0
other 4/4	4	0	0	2	2	0	8	1
waltzes	0	1	3	6	6	0	0	7
accuracy (%)	95.89	96.99	22.22	0.00	71.74	88.46	57.14	50.00

Table 4.5

Aggregate confusion matrix at window-level (top) and tune-level (bottom) for experiment B (column: reference, line: prediction)

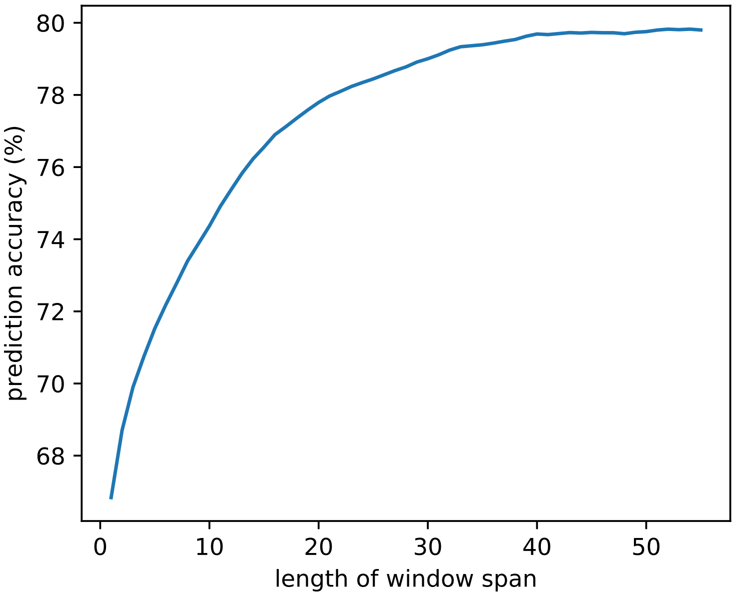


Figure 4.4

Prediction accuracy by span length for experiment B cluded in the challenge, in the same manner as in Experiment B. Some parameters were fine-tuned to obtain the best possible predictions.

- The sliding window for computing the ACF function is 10 seconds long, instead of 5.
- The "fuzz" parameter of the fuzzy histogram, allowing to match durations up to a certain ratio, is 0.25 instead of $1/3$.
- The *ql*-vectors are of size 64 instead of 16, and the P_i sets are now defined as $P_i = \{p \in P \text{ where } round(2 * p_l/q) = i\}$. Multiplying the numerator by 2 allows rounding at semiquaver positions.

The first modification is a consequence of the sampling rate used in the challenge (22.05kHz), twice lower than the one used in the present. The other two were obtained by performing a grid search, using the public dataset as a validation set.

The multinomial model succeeded at tune-level in 47 out of 80 recordings in the test set, or 58.8%. The accuracy is much lower than for ITM, but well above chance, here at 1/7 or 14.3%. The main difficulties were the presence of asymmetric rhythms in Greek music, and the small size of the public training set.

4.4 Discussion

A novel method for inferring rhythm information from an audio recording has now been introduced, relying on low-level spectral features and logistic regression classifiers. The dataset for this experiment was designed to follow the distribution of tune types in 4 reference collections. The performance on the method on this dataset reached 96.8% and 83.2% accuracy for metre type and tune type prediction, respectively.

One possible source of errors is the estimation of quaver duration. Indeed, the fuzzy histogram algorithm returns the most frequent note duration in a transcription, allowing for variations of ±33% (see Algorithm 1 page 68). According to Vallely, a reel "consists largely of quaver movement", a double jig is "characterised throughout by the rhythmic pattern of groups of three quavers" (Vallely, 2011, p. 570 and p. 368). These 2 tune types alone account for more than 70% of the repertoire, according to the average percentages presented in Table 4.2. Hence, in the majority of tunes, the value returned by the fuzzy histogram should indeed correspond to the duration of a quaver.

However, again according to Vallely, in a single jig "the pre-dominant rhythmic pattern is crotchet followed by quaver" and for a slide "the predominant rhythm involves the alternation of crotchets and quavers creating the feeling of long and short" (Vallely, 2011, p.368-369). This would mean that crotchets and quavers are in similar numbers in tunes of these types, and the fuzzy histogram could return either of the two.

Vallely also writes that "hop jigs are to slip jigs as slides are to double jigs ... in each case the three-quaver group is typically changed to a crochet + quaver group" (Moylan and O'Leary, 2014, cited in Vallely (2011)). In the choice of categories for the dataset, it was decided to group hop and slip jigs together, but this might have to be reconsidered in order to study more finely the way quaver duration is found.

A larger dataset would allow for more training, hopefully leading to higher accuracy. This would be especially important in the tune types that are less represented, as generally in machine learning more examples in the training set can help models generalise better.

Testing the method on solo recordings would be useful to further assess the robustness of the proposed approach. Indeed, although the onset detection function relies on spectral content and not on hard onsets from percussive instruments, drums or plucked string instruments (guitar, banjo) are present in most of the recordings in the dataset. Investigating the performance of the proposed method on flute or fiddle solo recordings could establish to what extent hard onsets help the rhythm inference.

Chapter 5

Key-invariant tune recognition

This chapter focuses on the task, presented in Section 2.3.1, of recognising tunes from audio recordings in a key-invariant manner. A new architecture is proposed, that does not rely on key recognition, and thus does not suffer from the difficulties of that task, discussed in Section 3.5.

The basic structure for tune recognition is based on the Tunepal architecture, presented in 2.3.5. Section 5.1 introduces the general architecture of the new proposed solution. The data for this experiment is presented in Section 5.2. Both patterns and searched texts have to be represented in a new way compared to Tunepal, presented in Section 5.3. At the core of this new method is the computation of pitch class histograms, which is detailed in Section 5.4. Section 5.5 details the implementation of the substring edit distance used as a measure of melodic similarity. The performance of the new algorithm is measured via 4 different metrics introduced in Section 5.6. It will be compared with the performance of the existing, non key-invariant, Tunepal architecture, and with that of a third approach, using the brute force method of considering all possible transpositions. Section 5.7 discusses these methods, and results are given in Section 5.8. The annotations dataset and the implementation of the algorithms presented in this chapter are publicly available.[1]

5.1 Architecture

Like in Tunepal, the aim is to recognise a tune played in an audio recording. A corpus, or search space, is prepared from an online collection of tune notations. The audio query is transcribed using an AMT algorithm. Melodic similarity is measured by substring edit distance, and the tunes in the search space that are most similar to the transcribed query are returned to the user.

In its current state, one of the main limitations of the system is that it is not key-invariant: a tune can only be recognised when played in the same key as the score present in the search space. However, as discussed in Section 2.2, the actual key of a tune can be said to be of little importance in practice. In Chapter 3, an architecture was proposed, following the method proposed in Martiniano and Silla (2017) of transposing queries and candidates to a common key. However, the lack of robustness of the key detection algorithm, an essential element in that architecture, as well as the ambiguity of the notion of key in ITM, motivated the new

[1] `https://github.com/pierrebeauguitte/tunerecognition`

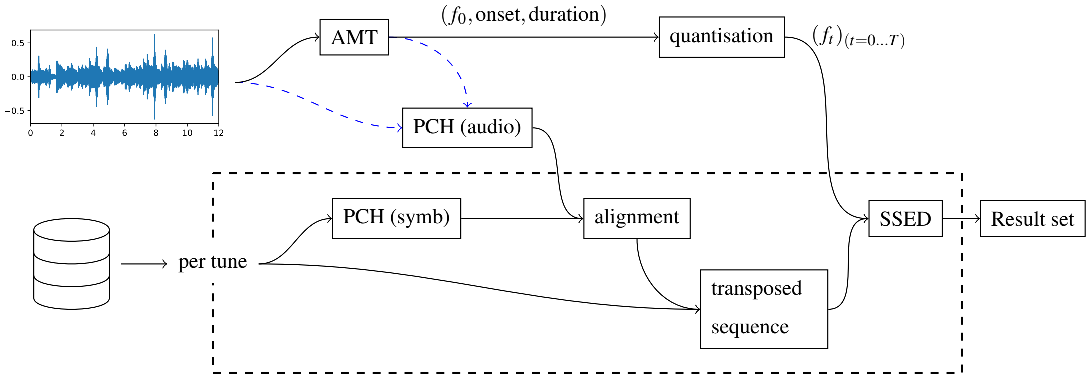


Figure 5.1
Architecture for key-invariant tune recognition

architecture presented in this chapter.

The new architecture is illustrated on Figure 5.1. It does not require any key annotation, but rather relies on the computation and alignment of pitch class histograms (PCH). Two different techniques for the computation of the audio PCH will be considered, and are represented by the two dashed blue arrows on the diagram.

5.2 Dataset

This section presents the dataset used in the tune recognition study. It consists of 2 parts: a collection of scores, or tunes in symbolic format, referred to as the search space and presented in 5.2.1; and a collection of audio recordings of tunes, that the system has to identify. The audio collection is actually an extension of the dataset from the previous chapter, as is explained in 5.2.2.

5.2.1 Search space

The collection of tunes from The Session, already presented in Chapter 4, is chosen as the search space for this experiment. It is the largest available collection, and other collections mentioned above are actually included in it. It contains scores for all of the tunes present in the audio dataset. Because it is a collaborative site, the set of tunes is always evolving. An online repository[2] contains weekly snapshots of the database. As in Chapter 4, and for the rest of this thesis, the snapshot from February 24th 2019 is used.

Each score is identified with a unique *setting* id, and a *tune* id, which can be shared by different settings. The term *cardinality* will denote the number of settings a particular tune has. This allows the collection to account for variations: two settings of a same tune are meant to be variations of a tune. In practice however this is not always respected, and some duplicates exist (different tune ids when they should be variants of a same tune), and settings can sometimes be different melodies entirely, but sharing a same name.

In total, the database comprises 32,747 settings grouped under 15,787 tunes, of cardinality ranging from 1 to 28.

[2] `https://github.com/adactio/TheSession-data`

5.2.2 New data fields on audio dataset

The same audio dataset introduced in 4.2 is used here. Three new fields are introduced in the annotations:

- tune id in the search space. This will serve as the ground truth for the tune recognition experiments. Because the recording versions typically differ to some extent from any of the scores on The Session, annotating an exact setting is unpractical, and would require to somewhat arbitrarily decide when a setting is the "same" tune, a variation of the tune, or a different tune altogether. Because duplicate tunes exist on The Session, some tunes are annotated with several ids (up to 4). Cardinality in such cases is defined as the sum of the cardinalities of the individual tune ids. With this new definition, cardinality of audio excerpts reaches 29.
- key deviation, indicating when the recording is in a different key than all the scores from The Session. This information is not necessary for the key-invariant architecture, but will be passed as a ground truth annotation to the non key-invariant recognition algorithm $Original_{key}$ used as a ceiling, as detailed below in 5.7. Only 20 of the 500 recordings are in transposed keys.
- as the aim is to identify excerpts of tunes, randomly placed in the tune, 4 random offsets are chosen on each tune. The excerpts duration is chosen at 12 seconds, so that it typically corresponds to at

least 8 bars of melody, as per Table 2.1. Offsets are spaced by a minimum of 5 seconds such that overlap is allowed. As the dataset comprises 500 recordings, this results in 2000 12-second excerpts.

5.3 Representation

This section introduces the way melodies are represented in the tune recognition system, as sequences of quavers. Pitch is represented in a different manner than in Tunepal (Duggan, 2010). The motivation for this, and the new representation, is explained in Section 5.3.1. Rhythm is represented in the same way as in Tunepal, and explained in Section 5.3.2. The next two sections detail how the quaver sequences are obtained from the audio query and the ABC tune notation.

5.3.1 Pitch representation

In Tunepal, pitch is represented by letters `A` to `G`. The symbols are obtained by normalising the ABC notation, as detailed in (Duggan, 2009). Because mode and accidentals are discarded in the process, a symbol represents indiscriminately the flat, natural and sharp note. This design choice addressed the fact that in ITM, thirds and sevenths can be considered "mobile", or be tuned in quarter tone between the tempered pitches. It is not unusual, for example, that tunes in D use both natural seventh $C\natural$ and flat seventh C. The pitch representation in Tunepal can be understood as considering only the degree of the note, discarding possible alterations.

An implicit assumption in what is described above is that tunes are played in *D*, or more precisely on *D*-based instruments. For the task at hand, this hypothesis has to be removed, and the equivalence between alterations of a note cannot be used. Indeed, when the audio query will be transcribed, there will be no way to *a priori* know which semitone intervals are equivalent or mobile (degrees $\flat 3/3$ and $\flat 7/7$) and which are not: *e.g.* degrees 3 and 4 are a semitone apart in a major mode, but are not equivalent.

For this reason, pitch will be represented by integers, with the convention $C = 0, C\sharp/D\flat = 1, \ldots, B = 11$. Pitch class is represented, and not absolute pitch value (as MIDI notes), because as discussed in Section 2.2.6 octave equivalence is suited to the heterophony of ITM. In Tunepal, octave equivalence was achieved by uppercasing all note symbols and discarding the octave change symbols (, and ' for lower and higher, respectively).

Finally, the value 12 is added to the alphabet to represent silence.

5.3.2 Rhythm representation

Melodies are represented as sequences of quavers, as in Tunepal. There is no other rhythm structure added such as beats or bars. Both the audio query and ABC tune candidates will be represented in this manner. As a consequence, parts of tunes can be simplified or omitted: semiquavers and short ornamental notes are discarded, and triplets are represented as normal quavers. However, this simplification is justified by the musical idiom considered, since, as discussed in Section 2.2.1, quavers are the fundamental rhythmic unit in ITM.

The next 2 sections explain how these representations are obtained from the ABC tunes and the audio queries.

5.3.3 ABC to quaver sequence

In (Duggan, 2009, Section 6.7), the ABC notation is normalised through a 4-step process, which is summarised here and illustrated in Figure 5.2. First, text annotations and ornaments are discarded; then long notes are expanded in quavers; repeated sections are expanded, and bar symbols are removed; finally all notes are brought to the same register by uppercasing all symbols, and removing `,` and `'`.

Here, the same process is applied, with two differences: first, alterations (_, =, ^ for flat, natural and sharp, respectively) are kept in the representation, whereas they are considered are ornaments and thus discarded in Tunepal; second, normalisation is followed by a last step translating the note symbols into pitch class. In order to do so, the text field indicating the mode (`K:`) is used to define a mapping between letters and pitch classes, represented by integers. Alterations are processed to affect the pitch class value by ±1. In the example shown on Figure

Original:

```
T: Come West Along The Road
K: G
[...]
d2BG dGBG|~G2Bd efge|d2BG dGBG|1 ABcd edBc:|2 ABcd edBd||
```

After ornamentation filtering:

mapping (G=7, A=9, B=11, C=0, D=2, E=4, F=6)

```
d2BGdGBG|G2Bdefge|d2BGdGBG|1ABcdedBc:|2ABcdedBd||
```

After note expansion:

```
ddBGdGBG|GGBdefge|ddBGdGBG|1ABcdedBc:|2ABcdedBd||
```

After section expansion:

```
ddBGdGBGGGBdefgeddBGdGBGABcdedBc
ddBGdGBGGGBdefgeddBGdGBGABcdedBd
```

After register normalisation:

```
DDBGDGBGGGBDEFGEDDBGDGBGABCDEDBC
DDBGDGBGGGBDEFGEDDBGDGBGABCDEDBD
```

Figure 5.2

Normalisation stages for ABC notation, adapted from Duggan (2009)

ABC	Tunepal	Proposed representation

`T: Kitty Lie Over`
`[...]`
`K: Dmaj`
`|:B|AFD DFA|BdB BAF|`
`ABA F2D|FEE E2B|`
	BAFDDFABDBBAF
	ABAFFDFEEEEB
		11, 9, 6, 2, 2, 6, 9, 11, 2, 11, 11, 9, 6, 9, 11, 9, 6, 6, 2, 6, 4, 4, 4, 4, 11

`T: The Musical Priest`
`[...]`
`K: Gmin`
`|:GF|DGG^F GABG|`
`AG=Fd cAGF|DGG^F GABG`
	GFDGGFGABG
	AGFDCAGFDGGFGABG
		7, 5, 2, 7, 7, 6, 7, 9, 10, 7, 9, 7, 5, 2, 0, 9, 7, 5, 2, 7, 7, 6, 7, 9, 10, 7

Table 5.1

Comparison of representation of ABC tunes in Tunepal and in the proposed method

5.2, the annotation `K: G` would result in the following mapping for *G* major:

$$(\mathtt{G}=7,\mathtt{A}=9,\mathtt{B}=11,\mathtt{C}=0,\mathtt{D}=2,\mathtt{E}=4,\mathtt{F}=6)$$

Two examples of ABC notations to quaver sequences are given in Table 5.1, along with the Tunepal representation for comparison.

Rests, represented by the letter z, are kept like all other notes. As will be explained below in Section 5.5, silences in the query will act as a wildcard, that is, will be allowed to match with any other note. This is useful for instruments such as flute and whistle, where the player often needs to skip a note in order to take a breath. However, some ABC tunes in the search space contain long sequences of rests, and allowing these to act as wildcards would result in these tunes matching any query.

As was explained above in Section 5.2.2, audio queries are taken as 12-second audio excerpts placed randomly in a recording. In most cases, the tunes are played several times, thus it can happen that such an excerpt spans the end of the tune and the beginning of the next repetition. In order for the symbolic representations in the search space to contain the quaver sequences corresponding to these situations, and thus to make matching possible (see below Section 5.5), the first half of the sequence is appended to the representation. Appending only half of the sequence is sufficient, as it corresponds to at least one full part, or 16 bars, typically lasting longer than 12 seconds (see Section 2.2.1).

5.3.4 Audio to quaver sequence

An audio excerpt is first transcribed using an AMT algorithm, resulting in a sequence of notes in the form $(f_0, \text{onset}, \text{duration})$. The quaver duration q is obtained by constructing the fuzzy histogram (Algorithm 1 page 68) from this sequence of notes. The notes are then quantised: a note of pitch p and duration d results in $round(\frac{d}{q})$ quavers added to the sequence. This means that notes shorter than $0.5q$ are discarded. The gap between notes is also quantised, and represented by the value 12. Pitch is first aligned to the closest MIDI note, then reduced to a pitch class in [0, 11].

The state-of-the-art AMT algorithms presented in Section 2.3.1.2, namely pYIN, Silvet, Melodia, and its variant PitchMelodia, will be considered. In addition, the algorithm originally designed for Tunepal, MATT2 (Duggan, 2009), standing for Machine Annotation of Traditional Tunes, will also be considered. In MATT2, pitch is estimated from the harmonicity of the signal, by processing the peaks of the power spectrum. Although it is not a state-of-the-art AMT algorithm, it was designed and fine-tuned specifically for ITM, and will serve as a baseline here.

Open source implementations of these algorithms are available.[3] Silvet, pYIN, and MATT2 all return notes in the desired format. Melodia (and its variant PitchMelodia) do not return a sequence of discrete notes, but a continuous pitch contour. To obtain discrete sequences of notes, the PitchContourSegmentation algorithm provided by the Essentia library[4] is applied (Bogdanov et al., 2013).

[3] respectively at https://code.soundsoftware.ac.uk/projects/pyin, https://code.soundsoftware.ac.uk/projects/silvet, http://mtg.upf.edu/technologies/melodia, and https://github.com/skooter500/matt2

[4] https://essentia.upf.edu

5.4 Pitch class histogram computation and alignment

This section introduces the computation of both symbolic and audio PCH, as well as the alignment method.

5.4.1 Symbolic PCH

The symbolic PCH is obtained from the original ABC by summing the durations of the notes of each pitch class, on a 120-valued vector where the unit is 10 cents. At this point, only multiple of 100 cents, corresponding to the tempered semitones, are non-zero, as is shown on Figure 5.3a. To make this representation closer to the audio one, and allow for slight tuning deviations in the match, this vector is then filtered by performing a convolution with a Gaussian kernel of size 150 cents. The vector is normalised to sum to 1. An example is plotted on Figure 5.3b.

5.4.2 Audio PCH

Two different ways of computing the audio PCH are presented below. In Section 5.4.2.1, the PCH is derived from the power spectrum of the recording, obtained by Fourier transform. In Section 5.4.2.2, it is computed from the notes resulting from the AMT algorithm, as in Chordia and Şentürk (2013). An advantage of this second method is that only pitch information contributes to the PCH. In contrast with this, the power spectrum contains all the harmonics of the signal, and the PCH will be affected by non-pitch properties of the audio recording, such as loudness and timbre. However, the second method is affected by errors made by the AMT algorithms, and the power spectrum offers a less error-prone representation.

5.4.2.1 From the power spectrum

In this first method, the audio PCH is derived from the power spectrum of the signal, and notated PCH$_{PS}$. A short-time Fourier transform (STFT) is performed, with a window size of 4096 samples, or approximately 93ms. This choice of a large window size is motivated by the fact that temporal information is irrelevant for the task at hand, thus frequency resolution is more important than time resolution. The power spectrum of the whole signal is obtained by summing the spectrum at each window. The next steps are to convert the frequencies of the Fourier bins to a cent scale, and to wrap this on one octave. Choosing a 10-cent resolution, this results in a 120-valued vector. Here too the vector is normalised to sum to 1. An example is plotted on Figure 5.3c.

Following this logic of discarding temporal information further, another method is to only perform a single Fourier transform on the whole signal. The previous method, based of STFT, will be referred to as *local*; this new one as *global*. A post processing step of discarding all frequencies higher than 5kHz from the power spectrum is also considered.

5.4.2.2 From the transcribed notes

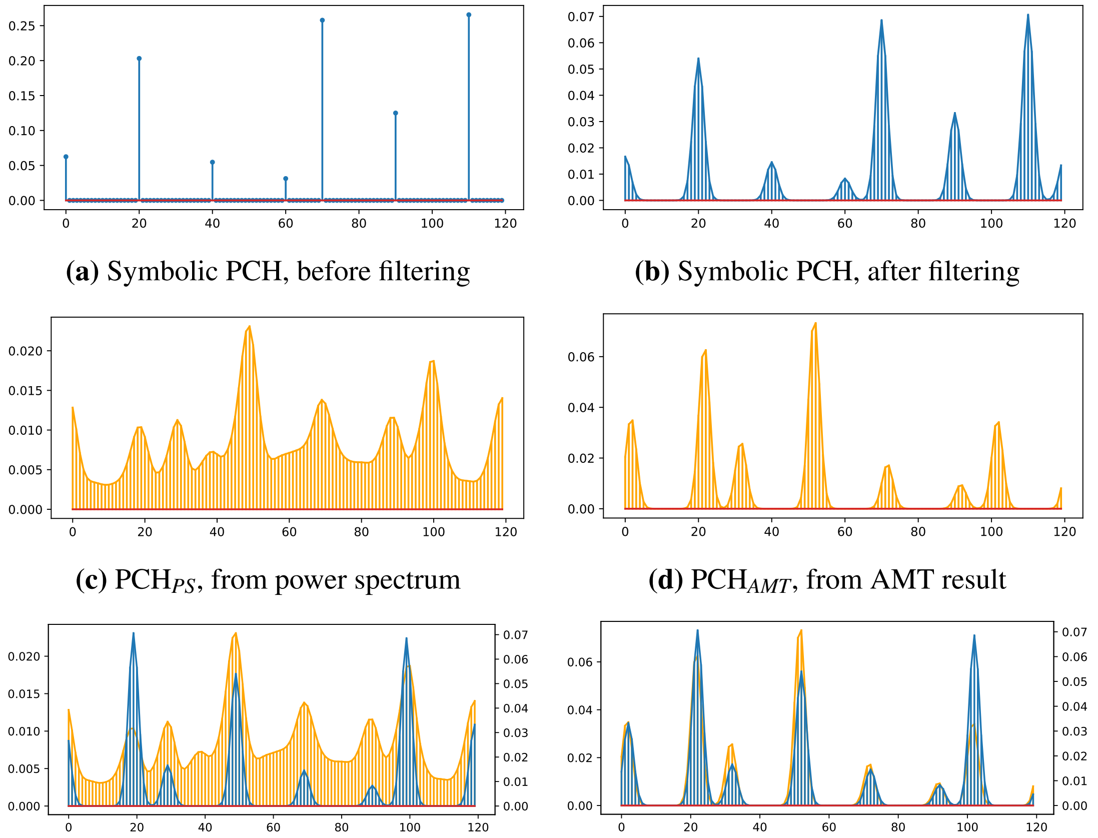
A second way of computing the audio PCH is to consider the notes extracted by the AMT algorithm, as explained in Section 5.3.4, and to then apply the same method as for the symbolic PCH, described in Section 5.4.1. Audio PCHs computed in this manner are denoted by PCH$_{AMT}$. As discussed above, this representation will not be affected by changes in timbre or loudness; however, it will be affected by errors in the transcription. An example is plotted on Figure 5.3d.

5.4.3 Alignment

Once the 2 PCHs are generated, the next step is to find the best alignment. In more precise terms, if the audio and symbolic PCHs are denoted by, respectively, $h^A = (h^A[i])_{i \in [1,120]}$ and $h^S = (h^S[i])_{i \in [1,120]}$, and the symbolic PCH *rotated* by k steps $h^S_k = (h^S[i-k])_{i \in [1,120]}$, the optimal rotation value k is the one maximising the similarity between h^A and h^S_k. Following Chordia and Şentürk (2013), the Bhattacharyya coefficient is chosen as a similarity measure:

$$BC(p,q) = \sum_{i=1}^{n} \sqrt{p_i q_i} \tag{5.1}$$

Going back to the architecture, only one audio PCH is needed for a query, and this PCH will be aligned to all of the symbolic PCH in the search space. For efficiency, these symbolic PCH can be pre-computed.



(a) Symbolic PCH, before filtering

(b) Symbolic PCH, after filtering

(c) PCH$_{PS}$, from power spectrum

(d) PCH$_{AMT}$, from AMT result

(e) Alignment of PCH$_{symb}$ and PCH$_{PS}$

(f) Alignment of PCH$_{symb}$ and PCH$_{AMT}$

Figure 5.3
Pitch class histograms

Once this optimal rotation k is found, the symbolic quaver sequence is transposed by $round(k/10)$ semitones, as the unit for k is 10 cents. For the examples plotted on Figure 5.3, BC is maximised with $k = 91$ for PCH$_{PS}$, and $k = 88$ for PCH$_{AMT}$. Figures 5.3e and 5.3f shows how the peaks are aligned with this optimal k. In this case, the symbolic sequence is transposed by $round(k/10) = 9$ semitones.

For each tune in the dataset, and for each of the 4 audio PCH computation methods discussed above, optimal rotation k is computed, and compared to the ground truth annotation introduced in Section 5.2.2, in-

Method	Number of success (out of 2000)
local	1835
local (<5kHz)	1850
global	1986
global (<5kHz)	1986
Silvet	1994
PitchMelodia	1998
Melodia	1991
pYIN	1939
MATT2	1900

Table 5.2
Alignment performance with PCH_{PS} (top) and PCH_{AMT} (bottom)

dicating the key deviation between recording and score. The numbers of successful transpositions per computation method are given in Table 5.2. With PCH_{PS}, it appears that the *global* method, based on a single Fourier transform, performs best, and that discarding high frequencies does not affect the number of success. Consequently, in the rest of this thesis all PCH_{PS} are computed using the *global* method.

The success of alignments using PCH_{AMT} varies depending on the algorithm used. For Silvet, PitchMelodia, and Melodia, alignment is better than with PCH_{PS}, but not for pYIN and MATT2.

5.5 Computing similarity between sequences

A number of metrics and distances aiming at quantifying the notion of melodic similarity have been proposed (Janssen et al., 2017). Here, as in Tunepal, melodic similarity is measured via the substring edit distance between pitch sequences.

A straightforward way to compute the edit distance, or Levenshtein distance, is presented in (Wagner and Fischer, 1974). It is a dynamic programming algorithm, using a matrix of size $m \times n$, where m and n are the lengths of the two strings to compare. However what is needed here is *substring* edit distance (SSED), looking for occurrences of a pattern (the transcribed audio query) within a text (the candidate tune), allowing for edit operations. Sellers (1980) gives such an algorithm, where the only difference with (Wagner and Fischer, 1974) is the initialisation of the matrix. Its complexity is $\mathcal{O}(mn)$, where m is the length of the pattern p and n the length of the text T.

Wu and Manber (1992) present a bit-parallel algorithm for finding occurrences of p in T allowing for k edit operations, given here in Algorithm 2. In the present case, the alphabet Σ is the set of integers [0, 12] used in the representation of pitch sequences. Its complexity is $\mathcal{O}(k\lceil \frac{m}{w} \rceil n)$ where w is the bit word length and $\lceil . \rceil$ denotes the ceiling operation. As will be seen below, the pattern can reasonably be constrained to fit within only 1 computer word, hence $\lceil \frac{m}{w} \rceil = 1$, and the algorithm complexity is $\mathcal{O}(kn)$. The only restriction is that an upper bound on the distance k has to be set at the beginning. In the present context, the duration of audio queries is set to 12 seconds. As shown in Table 2.1 in Section 2.2.1 the duration of a quaver is typically above 100ms. Thus, an upper bound of about 120 symbols exists on the length of search patterns p, and having an upper bound k on the distance value is not problematic.

In this algorithm, the pattern p is encoded in a binary table B of size mk. An example, adapted from (Navarro and Raffinot, 2002) is shown in Figure 5.4. An occurrence of the pattern with l errors is recorded when the leftmost bit of R_l is set to 1. The SSED is then the minimum l whose leftmost bit becomes 1 during the reading of the text. The state of the leftmost bits is checked at line 19, and the variable d is updated accordingly. In order to allow for a silence (i.e. the value 12) to behave as a wildcard, the corresponding column in table B is filled with 1s whenever it is encountered (line 5). This will result in the & operator in line 16 to always return 1. Allowing a silence in the text to match any character in the pattern can be achieved simply by setting $B[12] = 1^k$. However, as explained above in Section 5.3.3, it was decided to only let rests in the query (*i.e.* pattern) act as wildcards.

An example of the computation of Algorithm 2 is given Figure 5.4a. The substring edit distance between these two sequences is 2, as is visible from the aligned sequences on Figure 5.4b.

As mentioned above, considering bit words of length 120 is enough for the task at hand. Some modern C compilers , like the GNU C Compiler[5], define the type `__uint128_t` to represent unsigned 128 bit integers. This makes it possible to represent the matrix B and $(R_i)_{i=0,\ldots,k}$ by numerical variables of this type, without needing the use of non-

[5] `https://gcc.gnu.org/`

$$B = \left\{ \begin{array}{|c|c|} \hline 2 & 0\,0\,1\,0\,1\,0\,1 \\ \hline 4 & 0\,0\,1\,0\,0\,1\,0 \\ \hline 7 & 0\,0\,1\,1\,0\,0\,0 \\ \hline 9 & 1\,1\,1\,0\,0\,0\,0 \\ \hline 12 & 0\,0\,1\,0\,0\,0\,0 \\ \hline \end{array} \right.$$

1. Initialise R_i

$R_0 = 0\ 0\ 0\ 0\ 0\ 0\ 0$
$R_1 = 0\ 0\ 0\ 0\ 0\ 0\ 1$
$R_2 = 0\ 0\ 0\ 0\ 0\ 1\ 1$
$R_3 = 0\ 0\ 0\ 0\ 1\ 1\ 1$

2. Read 9: 1 1 1 0 0 0 0

$R_0 = 0\ 0\ 0\ 0\ 0\ 0\ 0$
$R_1 = 0\ 0\ 0\ 0\ 0\ 0\ 0$
$R_2 = 0\ 0\ 0\ 0\ 0\ 1\ 1$
$R_3 = 0\ 0\ 0\ 0\ 1\ 1\ 1$

3. Read 2: 0 0 1 0 1 0 1

$R_0 = 0\ 0\ 0\ 0\ 0\ 0\ 1$
$R_1 = 0\ 0\ 0\ 0\ 0\ 1\ 0$
$R_2 = 0\ 0\ 0\ 0\ 1\ 0\ 0$
$R_3 = 0\ 0\ 0\ 1\ 1\ 1\ 1$

4. Read 2: 0 0 1 0 1 0 1

$R_0 = 0\ 0\ 0\ 0\ 0\ 0\ 1$
$R_1 = 0\ 0\ 0\ 0\ 1\ 1\ 1$
$R_2 = 0\ 0\ 0\ 1\ 1\ 1\ 0$
$R_3 = 0\ 0\ 1\ 1\ 1\ 0\ 0$

5. Read 2: 0 0 1 0 1 0 1

$R_0 = 0\ 0\ 0\ 0\ 0\ 0\ 1$
$R_1 = 0\ 0\ 0\ 0\ 1\ 1\ 1$
$R_2 = 0\ 0\ 1\ 1\ 1\ 1\ 1$
$R_3 = 0\ 1\ 1\ 1\ 1\ 1\ 0$

6. Read 7: 0 0 1 1 0 0 0

$R_0 = 0\ 0\ 0\ 0\ 0\ 0\ 0$
$R_1 = 0\ 0\ 0\ 1\ 0\ 1\ 1$
$R_2 = 0\ 0\ 1\ 1\ 1\ 1\ 1$
$R_3 = 0\ 1\ 1\ 1\ 1\ 1\ 1$

7. Read 7: 0 0 1 1 0 0 0

$R_0 = 0\ 0\ 0\ 0\ 0\ 0\ 0$
$R_1 = 0\ 0\ 1\ 0\ 0\ 0\ 0$
$R_2 = 0\ 1\ 1\ 1\ 1\ 1\ 1$
$R_3 = \mathbf{1}\ 1\ 1\ 1\ 1\ 1\ 1$
$d = 3$

8. Read 7: 0 0 1 1 0 0 0

$R_0 = 0\ 0\ 0\ 0\ 0\ 0\ 0$
$R_1 = 0\ 0\ 0\ 0\ 0\ 0\ 0$
$R_2 = 0\ 1\ 1\ 1\ 0\ 0\ 0$
$R_3 = \mathbf{1}\ 1\ 1\ 1\ 1\ 1\ 1$
$d = 3$

9. Read 9: 1 1 1 0 0 0 0

$R_0 = 0\ 0\ 0\ 0\ 0\ 0\ 0$
$R_1 = 0\ 0\ 0\ 0\ 0\ 0\ 0$
$R_2 = \mathbf{1}\ 1\ 1\ 0\ 0\ 0\ 0$
$R_3 = \mathbf{1}\ 1\ 1\ 1\ 0\ 0\ 0$
$d = 2$

(a) Steps of Algorithm 2

text	9	2	2	2	7	7	7	9
pattern		2	4	2	7	12	9	9
operation			S			*	S	

(b) Edit operations on aligned sequences(S = substitution, * = wildcard)

Figure 5.4
SSED computation for pattern (2,4,2,7,12,9,9) in text (9,2,2,2,7,7,7,9)

Algorithm 2: Bit-parallel SSED computation

```
Data: $p = p_1 p_2 \ldots p_m$ the pattern, $T = t_1 t_2 \ldots t_n$ the text, $k$ the maximum number of edits
Result: $d$ the substring edit-distance
/* Preprocessing                                                    */
for $c \in \Sigma$ do $B[c] \leftarrow 0^m$
for $j \in 1 \ldots m$ do
    $B[p_j] \leftarrow B[p_j] \mid 0^{m-j}10^{j-1}$
    if $p_j = 12$ then
        for $l \in 0 \ldots 11$ do $B[l] \mid= 0^{m-j}10^{j-1}$
    end
end
$d = k$
/* Searching                                                        */
for $i \in 0 \ldots k$ do $R_i \leftarrow 0^{m-i}1^i$
for $pos \in 1 \ldots n$ do
    $oldR \leftarrow R_0$
    $newR \leftarrow ((oldR << 1) \mid 0^{m-1}1) \ \& \ B[t_{pos}]$
    $R_0 \leftarrow newR$
    for $i \in 1 \ldots k$ do
        $newR \leftarrow ((R_i << 1) \ \& \ B[t_{pos}]) \mid oldR \mid ((oldR \mid newR) << 1)$
        $oldR \leftarrow R_i$
        $R_i \leftarrow newR$
        if $i < d$ and $R_i \ \& \ 10^{m-1} \neq 0^m$ then
            $d \leftarrow i$
        end
    end
end
return $d$
```

standard libraries for big integers. The maximum number of edits k is set to 64, thus allowing half the pattern to be edited. This is enough for the task at hand, as changing half the pattern already results in a widely different melody.

To be consistent with the annotations (see Section 5.2), the distance between a transcribed pattern and a tune is defined as the lowest distance between the pattern and all the settings of that tune.

5.6 Performance metrics

In this section, several ways of evaluating the tune recognition system are discussed. Metrics introduced earlier in Section 2.3.1.4 are discussed again and adapted to the present experiment. Section 5.6.1 presents the standard MIREX metrics. Rank metrics are discussed in Section 5.6.2, and the definition of rank is clarified to account for draws. A new metric is introduced in Section 5.6.3 to provide better insight into the robustness of the retrieval algorithm.

5.6.1 Hit rates

The tune recognition task discussed in this chapter corresponds to the audio-to-symbolic situation of the Query by Singing/Humming (QbSH) task presented in Section 2.3.1.

Submissions to the QbSH task are evaluated with a single metric, the "Top-10 hit rate (1 point is scored for a hit in the top 10 and 0 is scored otherwise)."[6] This metric will be reported, along with a Best (or Top-1) hit rate, which from a user perspective is a more meaningful metric.

[6] `https://www.music-ir.org/mirex/wiki/2019:Query_by_Singing/Humming`

5.6.2 Rank

Another metric that is often reported for QbSH tasks is the Mean Reciprocal Rank (MRR) (Dannenberg et al., 2007; Salamon et al., 2013).

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{5.2}$$

where $rank_i$ is the rank of the correct item (here, the candidate tune) in the ordered result set for query i. For clarity, this will also be referred to as the rank of the query from now on. This value is in $]0,1]$, higher values indicate better performance. MRR is the average of the reciprocal rank (*i.e.* inverse of the rank) over N queries.

The candidate tunes are sorted in order of ascending edit distance to the query. As this distance takes only integer values, it can occur frequently that several tunes are at the same distance, in which case their ordering in the result set is purely incidental. It is thus important to clarify the definition of rank in case of draws.

In order to tackle this problem, Martiniano and Silla (2017) use the Mean draw Reciprocal Rank (MdRR), defined as

$$\text{MdRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{d_i \times r_i} \tag{5.3}$$

where r_i and d_i refer to the rank and the number of draws in the rank of the correct tune. Two examples of draw reciprocal ranks $(1/(d_i \times r_i))$ over possible result sets are given in Figure 5.5. An issue appears with

id	d	rank
723	2	1st
924	2	1st
157	6	3rd
522	7	4th
65	**7**	**4th**
147	11	6th
35	12	7th
...		

$\text{dRR} = \frac{1}{4\times 2} = 0.125$

id	d	rank
865	3	1st
143	3	1st
295	**3**	**1st**
754	3	1st
174	3	1st
907	3	1st
616	3	1st
772	3	1st
621	7	9th
...		

$\text{dRR} = \frac{1}{1\times 8} = 0.125$

Figure 5.5
Examples of MdRR (the correct tune appears in bold)

this metric: in the first situation, the correct tune can only be in position 4 or 5; in the second, it could appear anywhere between positions 1 and 8. However the dRR value is the same in both cases. The position of the correct answer in an information retrieval task is important, and from that point of view the first example performs better than the second one, which the MdRR metric fails to capture.

To address this problem, a new way of ranking is now introduced. For a query i, let $(d_k^{(i)})_k$ be the distinct edit distances obtained in the result set, in decreasing order, and let $(n_k^{(i)})_k$ be the number of candidate tunes at each d_k value, such that there are n_k tunes obtaining distance d_k. Then the *worst possible rank* (WPR) of that query is defined as $\sum_{l=1}^{k} n_l$ where d_k is the distance of the correct tune. The two examples from Figure 5.5 obtain WPR values of 5 and 8, respectively. Figure 5.6 gives 3 further examples of such result sets.

id	d
64	**4**
3562	6
260	6
1483	6
2935	10
13	10
5298	12
237	12
758	12
19	15
...	...

$(d_i) = (4, 6, 10, 12, 15, \dots)$
$(n_i) = (1, 3, 2, 3, \dots)$

$\text{WPR}_i = 1$

id	d
572	**3**
134	3
2389	3
180	5
6002	7
3874	9
54	11
248	11
549	11
439	11
...	...

$(d_i) = (3, 5, 7, 9, 11, \dots)$
$(n_i) = (3, 1, 1, 1, \dots)$

$\text{WPR}_i = 3$

id	d
62	3
28	3
3460	3
383	**5**
902	5
504	9
340	11
349	11
2459	11
1903	11
...	...

$(d_i) = (3, 5, 9, 11, \dots)$
$(n_i) = (3, 2, 1, \dots)$

$\text{WPR}_i = 5$

Figure 5.6
Examples of result sets and ranks

From now on, any use of *rank* will refer to this definition, and MRR is defined as per Equation 5.2. A query is counted as a Top-10 hit if its rank is less or equal to 10, and as a Best hit if its rank is 1.

So far, three metrics are defined, which all summarise the performances of a batch of queries in a single number (an integer in $[0, N]$ for Top-10 hit and Best hit, and a real number in $]0, 1]$ for MRR). This type of single number metric is good for comparing different algorithms, but does not allow for more powerful statistical analysis of the results. Keeping one value per query, or N values in total, will allow to run statistical tests on samples, instead of comparing single values.

Rank is a useful indicator of performance for information retrieval, but it does not allow to assess the robustness of a successful query. For example, a query of rank 1 with $d_1 = 2, d_2 = 15$ can be said to be more robust than one with $d_1 = 20, d_2 = 21$. Both are of rank 1, but in the first case the correct tune is better segregated from the rest than in the second case. One more error (in the computation of the edit distance) would make the second query fail, but not the first one. In the next subsection, a new metric that will account for this is introduced.

5.6.3 Relative difference

In (Stasiak, 2014), a new metric for evaluating a QbSH system is defined as:

$$\delta = \frac{1}{N} \sum_{i=1}^{N} \frac{E_2^{(i)} - E_1^{(i)}}{E_1^{(i)}} \tag{5.4}$$

where the sum is computed only over the N successful queries of a batch, and $E_1^{(i)}$ and $E_2^{(i)}$ are the edit distances of, respectively, the first and second results given by the algorithm for query i. Because this is only computed for successful queries, $E_1 < E_2$, and all summed values are positive.

Equation 5.4 is modified to define a new metric on a single query i as:

$$a_i = \frac{d_F^{(i)} - d_T^{(i)}}{max(d_F^{(i)}, d_T^{(i)})} \tag{5.5}$$

where d_T is the distance of the correct tune, and d_F is the lowest distance obtained by a wrong tune, or in other words the unwanted tune that is closest to the query. Dividing by the max ensures that $a_i \in [-1, 1]$. The metric is defined on all queries regardless of whether they are successful or not. $a_i > 0$ if and only if the search was successful (0 indicates a draw), or equivalently if the query has rank 1. Finally, a_i can be interpreted as a measure of confidence: higher a_i means higher distance between T and F, in other words higher discriminative power.

The divisor can only be 0 when both d_F and d_T are 0, in which case adopting $a_i = 0$ by convention is consistent with what happens in a draw. Instead of averaging all a_i over a set of queries, all individual values are kept for statistical testing, and the median a_i value over all queries (successful and failing) of a batch will be reported for summarising performances.

Four metrics are now defined to evaluate the tune recognition algorithm. Best hit rate, Top 10 hit rate and MdRR summarise the performance over N queries in a single value, while the whole sets of N values of the relative difference a can advantageously be kept for more powerful comparisons between different retrieval methods.

5.7 Methods for comparison

The proposed approach will be referred to as *Align* from now on, with subscript *PS* or *AMT* according to the audio PCH used. In order to assess its performance, it will be compared with three other methods.

The first one is the method currently implemented in Tunepal, which is not key-invariant. On the 80 audio excerpts taken from the 20 tunes that are recorded in a key different than that of the written transcription, the algorithm will compute an edit distance between sequences of pitch that are not aligned, and the odds of finding a match are low. This method is denoted $Original_0$ and will serve as a baseline.

The second method, $Original_k$, is an informed approach, where the key deviation introduced in 5.2.2 will simply be given to the program so that the symbolic pitch sequences can be transposed accordingly. This can be seen as the ceiling, as no errors can be made regarding the PCH alignment and transposition of symbolic pitch sequence.

The last one, *All*, is a brute force approach, where the similarity of the recording with all possible 12 transpositions of each tune is computed. It is a computationally expensive method, as 12 times more SSED have to be computed. Furthermore, the search space effectively becomes 12 times larger, making retrieval harder from a probabilistic point of view. This last method, unlike the other two, is key-invariant.

The proposed *Align* methods will be compared with these other three, in terms of retrieval accuracy, and in terms of computational cost, measured as run times.

5.8 Results

The results of the experiment are presented in this section. First, general observations and overall statistics are given in 5.8.1. Then Section 5.8.2 focuses on computational costs, and Section 5.8.3 offers more precise error analysis.

5.8.1 Overall statistics

The Best hit rate, Top-10 hit rate and median a value are given for all 5 AMT algorithms and all 5 search methods in Table 5.3. A first observation is that for all AMT algorithms except pYIN, the Best hit rate, Top-10 hit rate, and MRR of all 3 key-invariant methods are consistently lower than the ceiling $Original_k$, but higher than the baseline $Original_0$. For pYIN, this only holds for the Best hit rate of $Align_{PS}$.

For Silvet, PitchMelodia, and Melodia, $Align_{AMT}$ yields the highest Best hit rate among the 3 key-invariant methods. In the case of pYIN, this is achieved by $Align_{PS}$.

The median a value is, for all algorithms except MATT2, lower for all 3 key-invariant methods than for both the ceiling and baseline *Original* methods. To assess the significance of these differences, the Wilcoxon signed-rank test is used. Indeed, the samples are paired because the results are obtained from identical audio excerpts, and none of the distributions of the a samples are following a normal distribution.

AMT	Method	Best hit	Top-10	Med(a)	MRR
Silvet	*Original*$_0$	1693 (84.65%)	1838 (91.90%)	0.441	0.876
	Original$_k$	1750 (87.50%)	1902 (95.10%)	0.455	0.906
	Align$_{PS}$	1724 (86.20%)	1886 (94.30%)	0.438	0.894
	Align$_{AMT}$	1740 (87.00%)	1898 (94.90%)	0.438	0.902
	All	1736 (86.80%)	1899 (94.95%)	0.435	0.901
PitchMelodia	*Original*$_0$	1638 (81.90%)	1802 (90.10%)	0.324	0.852
	Original$_k$	1699 (84.95%)	1870 (93.50%)	0.333	0.884
	Align$_{PS}$	1666 (83.30%)	1838 (91.90%)	0.314	0.867
	Align$_{AMT}$	1676 (83.80%)	1854 (92.70%)	0.318	0.873
	All	1670 (83.50%)	1851 (92.55%)	0.316	0.872
Melodia	*Original*$_0$	1486 (74.30%)	1691 (84.55%)	0.286	0.783
	Original$_k$	1538 (76.90%)	1752 (87.60%)	0.294	0.810
	Align$_{PS}$	1492 (74.60%)	1711 (85.55%)	0.273	0.789
	Align$_{AMT}$	1494 (74.70%)	1715 (85.75%)	0.273	0.791
	All	1492 (74.60%)	1723 (86.15%)	0.273	0.791
pYIN	*Original*$_0$	1284 (64.20%)	1509 (75.45%)	0.176	0.985
	Original$_k$	1324 (66.20%)	1553 (77.65%)	0.190	0.706
	Align$_{PS}$	1288 (64.40%)	1508 (75.40%)	0.167	0.686
	Align$_{AMT}$	1278 (63.90%)	1500 (75.00%)	0.167	0.681
	All	1283 (64.15%)	1508 (75.40%)	0.167	0.684
MATT2	*Original*$_0$	1212 (60.60%)	1374 (68.70%)	0.085	0.639
	Original$_k$	1268 (63.40%)	1438 (71.90%)	0.098	0.669
	Align$_{PS}$	1236 (61.80%)	1406 (70.30%)	0.083	0.652
	Align$_{AMT}$	1241 (62.05%)	1415 (70.75%)	0.086	0.655
	All	1248 (62.40%)	1423 (71.15%)	0.087	0.662

Table 5.3
Results

These tests show statistically significant differences (p-values $< .001$) between the a values samples of any of the 3 key-invariant methods and both the ceiling and baseline *Original* methods. This reflects the fact that key-invariant tune recognition is a more complex task than the one tackled by both *Original* methods. However, a is merely an indication of the confidence of the matching, and all other metrics suggest that the

AMT	$Align_{PS}$ / $Align_{AMT}$	$Align_{PS}$ / All	$Align_{AMT}$ / All
Silvet	0.5157	0.1360	0.0019
PitchMelodia	0.2583	0.2104	0.0004
Melodia	0.6665	0.0472	0.2946
pYIN	4.19e-6	2.22e-6	0.1153
MATT2	0.2808	0.1868	0.5068

Table 5.4
p-values for Wilcoxon signed-rank tests between key-invariant methods

key-invariant approaches do outperform the baseline $Original_0$.

Table 5.4 show the p-values returned by the Wilcoxon signed-rank tests between the 3 key-invariant methods for all 5 AMT algorithms. For all algorithms except pYIN, no statistically significant difference appears between the 2 *Align* variants. At least one of the 2 *Align* variants result in a p-value > 0.1 when compared to the brute force method *All* across all AMT algorithms. In the case of Silvet and PitchMelodia, the median a value is higher for $Align_{AMT}$ than for *All*, suggesting that the significant difference observed (p-values < 0.005) is actually in favour of $Align_{AMT}$. The proposed *Align* method performs on a par with, and sometimes even better than, the brute force *All*.

5.8.2 Computational costs

The execution times of each batch of 2000 queries were measured using the Linux command `time`.[7] The `real` elapsed time is of little use, as it is affected by other local computations. Only `user` and `system` time, measuring CPU time spent on the actual process being monitored

[7] `https://linux.die.net/man/1/time`

Method	Step	CPU min.	Factor
Original	batch search	644.35	1
Align	PCH computation	22.98	
	batch search	800.28	
	total	823.25	1.278
All	batch search	7706.83	11.961

Table 5.5
Measured execution times

are worth reporting. Table 5.5 reports the sum of these two measures, in minutes, averaged across all 5 algorithms, and across the *Original* and *Align* variants, as the computational complexity of the variants is the same. The execution times for the automatic transcription of queries are not included, as these were computed beforehand. Because this step is identical for all 3 variants of the tune recognition system, it can be discarded in the comparison. For the *Align* search method, the computation time for the PCH generation for all 2000 excerpts is added. The overhead factor compared to *Original* is also reported.

It is worth noting that although these numbers seem to indicate a very time consuming process, the actual run times are much shorter than this, as the similarity computation across the search space can be done in parallel. The implementation written for this study[8] uses 8 threads, and runs in about 80 minutes (*Original*) on an i7 quad core CPU, or about 2.5 seconds per query.

These timings show that while the proposed approach *Align*, as was

[8] `https://github.com/pierrebeauguitte/tunerecognition`

discussed in the previous subsection, does as well as the brute force search *All*, it imposes a much lighter overhead in terms of computational runtime.

5.8.3 Error analysis

In this section, results are analysed following different categorisations. All results discussed here are obtained with *Silvet* and $Align_{AMT}$. The notation S_X will be used to refer to the sample of scores on the subset of category X.

5.8.3.1 Per instruments

Figure 5.7 shows the results obtained on solo recordings, grouped by instrument. A drop in performance for banjo and harp recordings is noticeable. This can be explained by the lack of harmonicity in the signal in the case of the banjo: as it is a plucked string instrument, notes are not sustained for a long time, making the AMT step challenging. In the case of the harp, the polyphonic style of playing is also making transcription harder.

The rest of the recordings are grouped according to how many musicians are playing (duo, trio, or group for 4 or more) and, for duos and trios, whether or not one of the instrumentalists is playing accompaniment, *i.e.* guitar, piano, bouzouki, harp or bodhrán. These instruments (except the last) can also be used melodically, but not in this set of re-

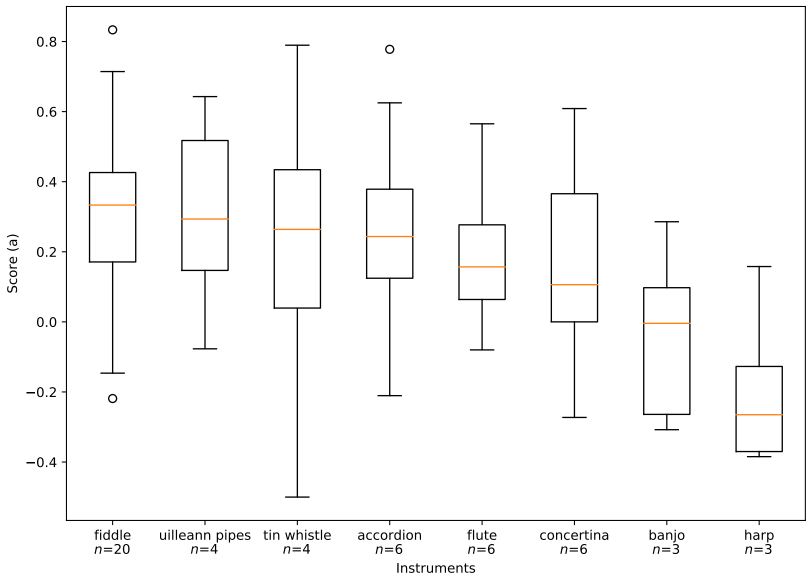


Figure 5.7
Scores per instruments for solo recordings

cordings. Conversely, accordions can be used harmonically, and accompany themselves on the left hand, but are mostly limited to the melodies in these recordings. The results are plotted in Figure 5.8. On both duos and trios, it appears that the presence of accompaniment makes tune recognition harder. However, the scores for recordings of larger ensembles, many of which contain some sort of accompaniment, tend to be higher. A possible explanation for this is the fact that harmonic accompaniment is more prominent is a small ensemble (*e.g.* a guitar and a flute) than in a large session, where as a rule harmonic instruments are a minority (Breathnach, 1996). Thus, duos and trios with accompaniment are closer to the definition of polyphony, which can be harder to transcribe than the heterophony of larger ensembles.

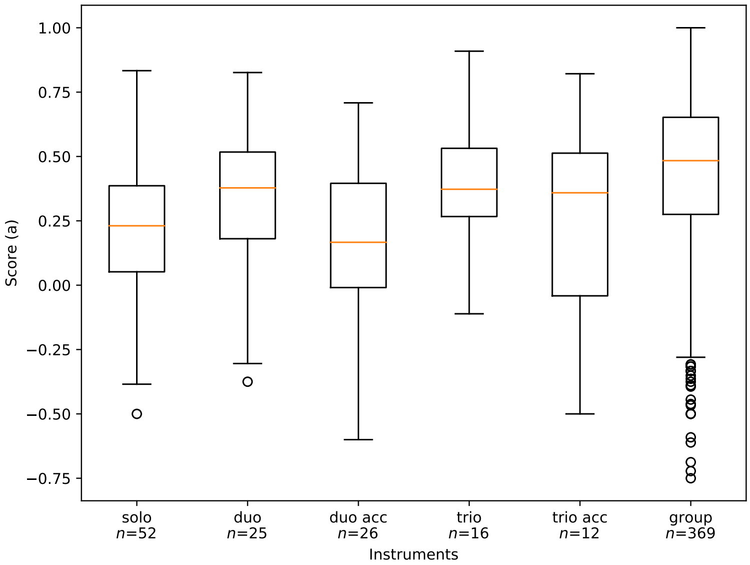


Figure 5.8
Scores per instruments

5.8.3.2 Per tune types

The scores grouped by tune types are plotted on Figure 5.9. Waltzes appear to be the most challenging tunes to recognise in the dataset. However, using Mann Whitney U test to compare the samples, it appears that the differences between S_{waltz} and $S_{slip\,jig}$, S_{polka}, and $S_{hornpipe}$ are not significant, with p-values > 0.1.

5.8.3.3 Per cardinality

As mentioned above, the structure of the search space is such that a tune can have one or more settings, a number referred to as the *cardinality* of a tune (see Section 5.2). Figure 5.10 shows how the 500 tunes of the dataset are distributed according to their cardinality. Most of the tunes have between 2 and 9 settings, with a mode at 6.

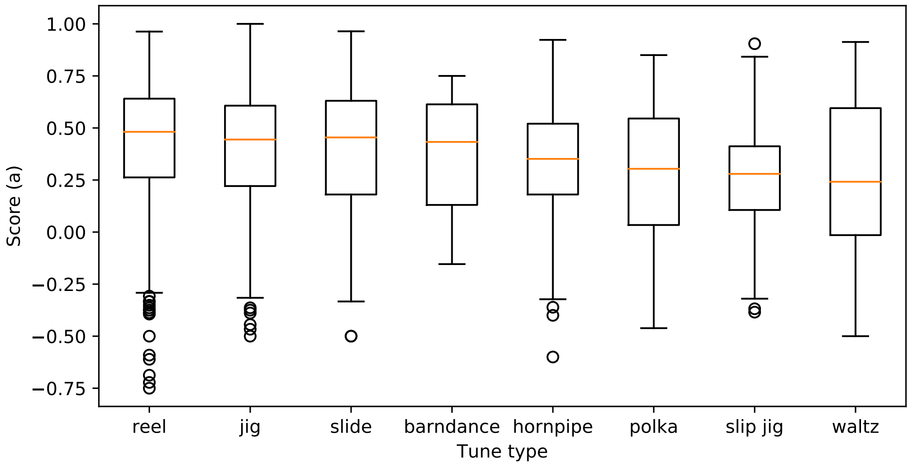


Figure 5.9
Scores per tune type

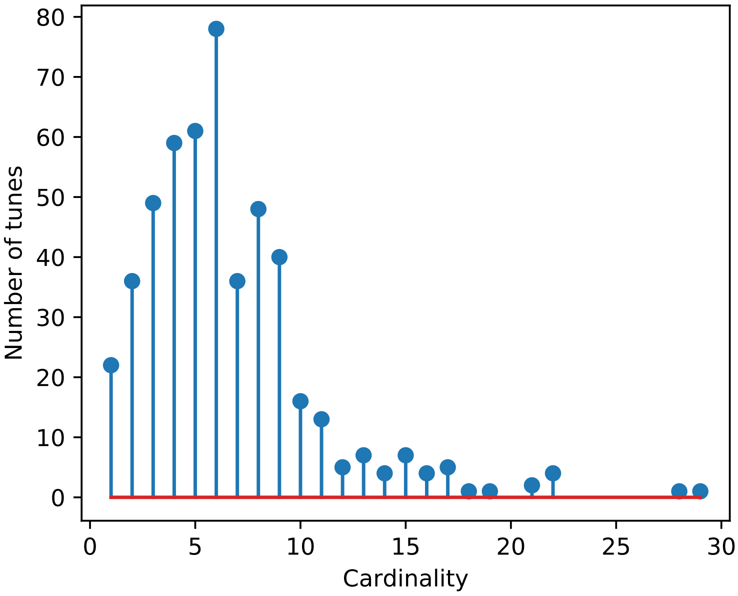


Figure 5.10
Number of tunes in the dataset per cardinality

On Figure 5.11, the scores grouped by cardinality are shown. S_k denotes the set of scores for excerpts of tunes with cardinality k, and $S_{>k}$ the set of scores for those with cardinality strictly above k. It appears that S_1 are lower than S_2, which in turn are lower than S_3. After this, the scores seem to flatten, and no clear trend is visible. Using a Mann-

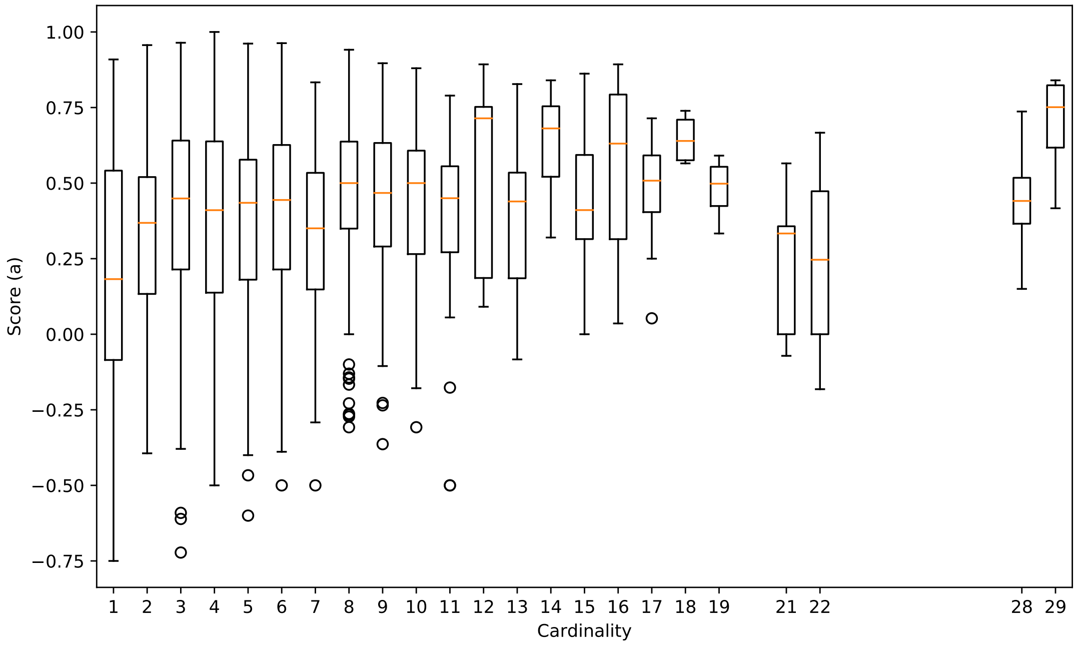


Figure 5.11
Scores per cardinality

Whitney U test, significant differences (p-values < 0.01) appear for pairs of samples (S_1, S_2), (S_2, S_3), $(S_1, S_{>1})$, and $(S_2, S_>2)$. The flattening observed is confirmed by the statistical testing, as the p-value for $(S_3, S_{>3})$ is large (0.319).

Two reasons can be given for these observations. The first one is a statistical explanation: an excerpt for a tune with cardinality 1 is statistically harder to find, as there is only 1 correct score in the search space. The second reason is musical: by the design of The Session, and consequently the search space, popular tunes tend to have more variants, and to be more normalised in playing. Thus, tunes of cardinality 1 can be more subject to interpretation.

5.9 Discussion

This chapter presented an architecture for key-invariant tune recognition, based on a novel transposition method based on PCH alignment introduced in Section 5.4. Two different approaches were explored for obtaining the audio PCH, from the power spectrum directly or from the notes returned by the AMT algorithm. Existing literature has often focused on audio-to-audio alignment in the context of cover song identification (Ellis and Poliner, 2007; Serrà et al., 2008), or has relied on key annotation when dealing with symbolic music (Chordia and Şentürk, 2013; Martiniano and Silla, 2017). The method proposed here performs audio-to-symbolic alignment without the need for key annotation or inference, thus alleviating the difficulty of defining key in ITM, discussed in Chapter 3.

Section 5.6 introduced two novel evaluation metrics: *worst possible rank* (WPR), is a way of ranking results that appropriately deals with draws. The relative difference measure *a* gives an indication of the robustness of retrieval, while its sign indicates the success or failure of a query. These metrics were designed to suit the experiment discussed in this chapter, but their definitions are general enough for them to be used in other similar experiments.

The key-invariant method *Align* has be shown to perform on a par with the brute force method *All*, while being only about 1.28 times slower than *Original* (versus about 12 times for the brute force). Although the non key-invariant ceiling method *Original*$_k$ did perform significantly better according to statistical testing, the Best hit success rate of *Align*$_{AMT}$ is only 0.5% lower (see Table 5.3). These findings are encouraging, and the next chapter will explore a method to improve further on the performance of key-invariant tune recognition by taking into consideration the rhythm classification models presented in Chapter 4.

Chapter 6

Using rhythm predictions to improve tune recognition

In terms of computational performance, the key-invariant *Align* method presented in the previous chapter has a large advantage compared to the brute force method *All*. However, its recognition performance is still not as good as the informed, non key-invariant, $Original_k$ method, and adds an overhead of more than 25% in terms of execution time.

This chapter proposes two different ways of integrating the rhythm predictions obtained from the models introduced in Chapter 4 to the tune recognition system, in order to improve on both computational complexity and recognition accuracy.

In a first method, presented in Section 6.1, the probability estimates returned by the models are interpreted as confidence scores, and are used to weight the distance between the query and a candidate tune. This method may improve accuracy, but will actually add even more overhead to the system. To address this concern, a second method is introduced in Section 6.2, which consists of pruning the search space according to the prediction. Three ways of performing this filtering are discussed. The possible benefits of this are twofold: computational cost is likely to be lower, as extracting low-level features and running them through a simple logistic regression model is cheaper than computing edit distance on the thousands of discarded candidate tunes; and precision might improve as pruning the search space can discard some spurious matches. Results will only be reported for the best performing AMT algorithm and search method, *i.e. Silvet* and $Align_{AMT}$, with the *AMT* subscript discarded for clarity. The same dataset as in the previous chapter is used. The implementation for the experiments carried out in this chapter is publicly available.[1]

6.1 Method A: weighted distances

In this first experiment, the probability estimates given by the rhythm prediction models from Chapter 4 are interpreted as confidence scores. Because logistic regression models are used, either binomial or multinomial depending on whether metre types (*duple* vs. *simple*) or tune types are predicted, the probability estimates are already calibrated.

Considering an audio query q, $d(q,t)$ is the edit distance between q and a candidate tune t (see Section 5.5), and $(p_k(q))_k$ the probability

[1] `https://github.com/pierrebeauguitte/tunerecognition`

estimates of the model, for k ranging in the different classes considered (2 or 8 depending on the models). Instead of sorting the candidates by the distances themselves as in Chapter 5, they are sorted on the weighted distance $d'(q,t) = f(p_k(q)) \times d(q,t)$, where f is a decreasing function of p, and k is the rhythm class of the candidate tune t. The function f has to be decreasing because the weighted value is a distance, *i.e.* a dissimilarity score: the higher the probability of q being of type k is, the smaller the weighting factor should be in order to make the weighted distance d' smaller. Four different functions are considered, and reported below in the results.

The same methodology as in Chapter 4 is used: a k-fold cross validation is performed, and all k models resulting from the successive trainings are kept. Then, for each audio excerpt, its tune or metre type is predicted using the model that has not seen it in training. This makes it possible to have a prediction for all the excerpts without any peeking.

Results are reported in Table 6.1, along with the performance of *Original*$_k$ and *Align* for comparison. It appears that using the tune type prediction (*i.e.* multinomial model) with $f(p) = \sqrt{1-p^2}$ gives the highest scores of the 8 combinations. This can seem surprising, as this model has poorer accuracy than the simpler binomial model (see Section 4.3).

Method	f	Best hit	Med(a)	MRR
Align		1740 (87.00%)	0.438	0.902
Original$_k$		1750 (87.50%)	0.455	0.906
Align with binomial model	$1-p$	1718 (85.90%)	0.438	0.889
	$\sqrt{1-p}$	1731 (86.55%)	0.440	0.896
	$1-p^2$	1728 (86.40%)	0.438	0.894
	$\sqrt{1-p^2}$	1737 (86.85%)	0.440	0.899
Align with multinomial model	$1-p$	1711 (85.55%)	0.435	0.887
	$\sqrt{1-p}$	**1742 (87.10%)**	**0.440**	**0.902**
	$1-p^2$	1739 (86.95%)	0.440	0.902
	$\sqrt{1-p^2}$	**1747 (87.35%)**	**0.444**	**0.906**

Table 6.1
Results of rhythm weighting

The median a value is reported, but now defined using the weighted distance d':

$$a_i = \frac{d_F'^{(i)} - d_T'^{(i)}}{max(d_F'^{(i)}, d_T'^{(i)})} \tag{6.1}$$

Although it is derived from a different distance measure, this new definition is consistent with the one given previously in Equation 5.5 and used in the two top lines of the table. Indeed, not using rhythm prediction is equivalent to weighting all distances by a same $f(p)$ value. The ordering of the results, and the value for a, are left unchanged: (i superscripts discarded for clarity)

$$\begin{aligned} \frac{d_F' - d_T'}{max(d_F', d_T')} &= \frac{f(p)d_F - f(p)d_T}{max(f(p)d_F, f(p)d_T)} \\ &= \frac{f(p)(d_F - d_T)}{f(p)max(d_F, d_T)} \\ &= \frac{d_F - d_T}{max(d_F, d_T)} \end{aligned} \tag{6.2}$$

As the probability estimates p are obtained from a logistic regression, their values are strictly bound by $0 < p < 1$. Thus, with the 4 functions f defined above, $f(p)$ is strictly positive, and the simplification in the last line of Equation 6.2 is valid. The consistency of the definition of a makes it possible to compare a values with and without weighting, and to use the same statistical tools as in Chapter 5.

Wilcoxon signed rank tests between the 16 weighted variants and *Original*$_k$ all return p-values close to zero, indicating significantly worse performance. On the other hand, the same test performed between the highlighted results and *Align* also return a p-value close to zero, which, together with the higher median a, indicate that the weighting method significantly improves the recognition accuracy. The last line in Table 6.1, using *Align*, multinomial model, and the function $f(p) = \sqrt{1-p^2}$, also shows the same MRR as *Original*$_k$. Results are better than those of *Align* alone, but they still do not reach the performance of *Original*$_k$, and obtaining the rhythm predictions adds a computational cost. In the next section, a different method of using the prediction to prune the search space is explored.

6.2 Method B: search space pruning

In their tune recognition system, Martiniano and Silla (2017) train a machine learning (ML) model to predict the tune type of a query, from features extracted from the MIDI representation, as in that case queries are not audio recordings but snippets of tunes in the search space. This filter is then used to discard matches of a different type than the one predicted by the model, but only in cases where there are draws is the result set, that is when $d_i > 1$ in the notation of Section 5.6.2. In their study, using the ML-based filter improved the retrieval accuracy of the system. However, as was the case in the previous section, no improvement on the computational cost can be gained, as the filtering happens *a posteriori*, once all distances between the tunes and the query have been computed.

Here, the rhythm predictions from the models introduced in Chapter 4 are used *a priori* to restrict the search space to contain only the tunes of the predicted rhythm type. In addition to a computational benefit, due to a smaller search space, an improvement in retrieval accuracy can be expected, as possible spurious matches of an incorrect rhythm type will be discarded. Three ways of performing this pruning of the search space are explored in this section.

The same metrics as in Chapter 5 are reported. However, the definitions of a and MRR have to be modified to account for situations when the correct tune is pruned out of the search space. For a query i, a_i in its current definition (Equation 5.5) is undefined if the correct tune is not in the search space, as d_T has no value. In such a case, the convention that $a = -1$ is adopted, corresponding to the worst case scenario when $d_T \to \infty$. Similarly, as the correct tune does not appear in the result list,

Model used	Best hit	Med(a)	MRR	Search space (%)
None	1740 (87.00%)	0.438	0.902	100
binomial	1654 (82.70%)	0.429	0.855	54.94
multinomial	1393 (69.65%)	0.373	0.720	25.87

Table 6.2
Results of pruning search space using all predictions

the reciprocal rank of the query $\mathrm{RR}_i = 1/rank_i$ is not defined. This definition is extended as $\mathrm{RR}_i = 0$, corresponding to the limit when $rank_i \to \infty$. The percentage of the total search space being used on average on the batch of queries is reported as well.

6.2.1 Using all predictions

A straightforward way to use the predictions to prune the search space is to only look within the tunes of the predicted class. All predictions are kept, even though the models do not have perfect accuracy; this means that whenever the model makes a false prediction, the correct tunes are discarded from the search space and the search cannot succeed.

Results are reported in Table 6.2. The lower retrieval accuracy was expected, as a wrong rhythm prediction implies a failed search. However, the difference of performance in terms of best hit rate is less than 5%, from 87.00% without pruning to 82.70% using binomial model predictions, when the size of the search space is almost halved. While these preliminary results are encouraging, the next 2 sections will attempt as improving on this method in order to reduce the difference in accuracy.

6.2.2 Maximising F1 score

As a binomial model is a binary classifier, it is possible to compute a F1-score (Kelleher et al., 2015). In order to do so, one of the 2 classes has to be considered positive. By convention, the rarer class *compound* is taken as the positive class, and *simple* as negative. The F1 score is then defined in terms of the precision P and the recall R:

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad F_1 = 2\frac{PR}{P+R}$$

where TP (true positives) is the number of instances of the positive class correctly classified; FP (false positives) and FN (false negatives) are the number of instances of the negative and positive class, respectively, incorrectly classified. This computation is only doable with the binomial model, *i.e.* the metre type predictions.

For an item x, a binomial logistic regression model will evaluate $p(x)$, the probability that x belong to the positive class; symmetrically, $1 - p(x)$ is the probability that x belong to the negative class. The standard approach, which was used in Chapter 4 and in the previous section, is to consider the class with the highest probability as the model prediction. Another way of using the probability estimates from the model is to define a threshold $\tau \in [0,1]$, and to classify an item x as positive if $p(x) > \tau$. The standard approach corresponds to $\tau = 0.5$. Here, the value of τ resulting in the highest F1 score is chosen.

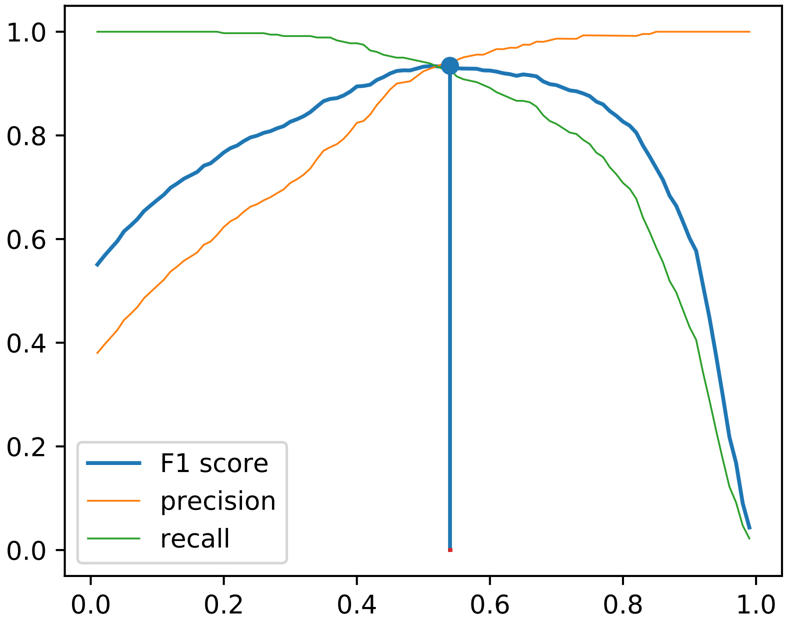


Figure 6.1
F1-score as a function of τ on training set

Following a general ML methodology, the dataset is split in training and test set to assess how well the method generalises. The 500 tunes in the dataset are grouped in two sets of 250 tunes each, balanced in terms of tune types and, consequently, also in metre types. As in Chapter 4, predictions are gathered on the training set using 10-fold cross validation, and the threshold maximising the F1-score on these 1000 predictions is determined as $\tau = 0.54$. Figure 6.1 shows how the F1-score varies as a function of τ.

Subsequently, one single rhythm prediction model is trained on the whole training set (as opposed to the 10 models needed in the 10-fold cross validation method), and its probability estimates for the *compound* and *simple* classes (p_c, p_s) on the held out test set are evaluated. Here again, the predicted class is *compound* if and only if p_c is greater than the

Method	Best hit	Med(a)	MRR	Search space (%)
Original$_k$	874	0.444	0.900	100
Align	864	0.429	0.892	100
$\tau = 0.5$ (standard)	809	0.414	0.835	55.25
$\tau = 0.54$ (max F1)	808	0.410	0.834	55.70

Table 6.3
Results on test set

chosen threshold $\tau = 0.54$. For example, an excerpt obtaining the probability estimates $(p_c, p_s) = (0.52, 0.48)$ would be predicted as *simple* (the negative class), even though $p_c > p_s$. Once the metre type for the excerpt is obtained, as in the previous section the search space is restricted to tunes of the same type.

Results on the test set for this new method are reported in Table 6.3, along with the results without filtering and with the standard classification ($\tau = 0.5$). The approach proposed here performs slightly worse than the standard binomial classification by all reported metrics. The difference between the two filtered results is not significant, as a Wilcoxon signed rank test gives a p-value of 0.433. This was to be expected, as the threshold found to maximise the F1-score is very close to the standard method. Only tunes obtaining $p_c \in [0.5, 0.54]$ have been classified differently.

A better way to integrate rhythm prediction in the system would be to allow some predictions to be ignored. Indeed, pruning the search space following a false prediction always leads to a unsuccessful search; however, the main downside of ignoring a true prediction is a loss of computational optimisation, as distances have to be computed for the whole search space.

The choice of a threshold in this section was made intrinsically: it was determined only from the performance of the rhythm prediction, and the impact of this choice on retrieval accuracy was only observed afterwards. In the next section, this choice will be made extrinsically: the training set will be used to find the best thresholds in terms of retrieval accuracy.

6.2.3 Per-class thresholds

As mentioned above in Section 6.1, the rhythm predictions p_k can be interpreted as confidence scores. The results of the 2 previous methods of filtering the search space suggested that ignoring some of the predictions could be beneficial. Here, only predictions above a certain threshold will be used to prune the search space; otherwise they will be ignored, their low value being interpreted as a lack of confidence. Results presented in 4.3 showed that prediction accuracy was different depending on the rhythm type, suggesting that confidence levels in the predictions can vary across classes. Consequently, a set of thresholds (τ_k) will be determined, and only if $p_k > \tau_k$ will the search space be restricted to tunes of type k.

6.2.3.1 Grid search

These thresholds are determined via an extrinsic evaluation, using a grid search, where each point of the grid corresponds to a set of thresholds (τ_k). The dimension of the grid is the number of classes N_c predicted by the model: 2 for the binomial model (metre type), and 8 for the multinomial model (tune type). For a query q, the predicted type k is taken as the class getting the highest probability estimate $p_k(q)$. In a logistic regression model, $\sum_k p_k(q) = 1$, thus the probability $p_k(q)$ of the predicted class is greater than $1/N_c$. Consequently, on each dimension k, the threshold τ_k varies in the range $[1/N_c, 1]$, or $[0.5, 1]$ for the binomial model and $[0.125, 1]$ for the multinomial. The grid step is chosen as 0.05.

With the binomial model, each axis $[0.5, 1]$ has 11 points, and the size of the 2-dimensional grid is $11^2 = 121$. With the multinomial model, each axis $[0.125, 1]$ has 18 points, and the size of the 8-dimensional grid is $18^8 = 1.1\text{e}10$. Two observations will help run the grid search in an efficient manner, which is particularly important for the multinomial case. First, by noticing that for each single query, only one of two things can happen: either the probability is below the threshold, the prediction is ignored, and the whole search space is used; or the prediction is kept and only the corresponding section of the search space is used. As the probability $p_k(q)$ remains constant across the grid, the unfiltered and filtered result sets can be pre-computed. Then, running the grid search only consists of combining these result sets according to the thresholds.

Second, the portion of the grid to consider can be reduced by looking at the range of the probabilities for each rhythm type. For example, all queries in the training set that are classified as *hornpipes* have $p_{hornpipe} \in [0.153, 0.610]$, so the threshold $\tau_{hornpipe}$ only needs to take values in $[0.15, 0.65]$. By reducing all 8 axes in this way, the grid size is 1.8e7, smaller by 3 orders of magnitude. With the binomial model, probability estimates for both classes range in $[0.5, 1[$, and the full grid has to be searched.

The same folds as in the previous section are used. On each point of the portion of the grid under consideration, the *Align* method of the tune recognition algorithm is run on the training set, using the thresholds-based filtering explained above. The point of the grid with the lowest thresholds, $\tau_k = 1/N_c$, corresponds to using all predictions as in Section 6.2.1; the one with the highest thresholds, $\tau_k = 1$, corresponds to the standard *Align* method with no filtering, as $p_k < 1$.

The criterion to select the best set of thresholds on the grid is the following: select the point that results in the highest number of best hits, while keeping the portion of the search space used as small as possible. The latter part of the criterion is useful to choose between several values resulting in the same number of best hits.

6.2.3.2 Results

On the training set, the *Align* method without filtering succeeds on 876 excerpts. Using the binomial model, there are 60 grid points, or almost half of the grid, that result in an equal or greater number of successes. The criterion defined above leads to the selection of the thresholds

$$(\tau_{compound} = 0.75, \tau_{simple} = 0.65)$$

resulting in 879 successes, while only looking in 63.73% of the search space on average. Rhythm predictions are used on 82.2% of the queries.

To assess how well this method, designated by $Align(\tau_{metre})$ below, generalises to unseen data, the algorithm is then run on the test set. The standard *Align* search succeeds on 864 excerpts with a median *a* value of 0.429. With the selected thresholds, there are 856 best hits, and the median *a* has the same value, 0.429. Only 64.60% of the search space is used on average, with 81.6% of the queries using filtering. These results are given on Table 6.4. The proposed method $Align(\tau_{metre})$ fails in 8 more cases than the *Align* method, and the MRR is slightly lower. However, a Wilcoxon test between the sets of *a* values of these 2 approaches gives a *p*-value of 0.899, indicating no significant difference. Moreover, the same test between the results of $Align(\tau_{metre})$ and *All* returns a *p*-value of 0.212, showing that the proposed method has a similar performance to the brute force *All*. However, a Wilcoxon test between res-

Split	Method	Best hit	Med(a)	MRR	Search space (%)	Filtered (%)
Train	$Original_k$	876	0.462	0.912	100	0
	$Align$	876	0.455	0.912	100	0
	All	876	0.455	0.912	100	0
	$Align(\tau_{metre})$	879	0.456	0.913	63.73	82.2
	$Align(\tau_{tune\ type})$	878	0.455	0.914	94.65	6.1
Test	$Original_k$	874	0.444	0.900	100	0
	$Align$	864	0.429	0.892	100	0
	All	860	0.424	0.890	100	0
	$Align(\tau_{metre})$	856	0.429	0.882	64.60	81.6
	$Align(\tau_{tune\ type})$	860	0.427	0.888	94.23	6.2

Table 6.4
Results of per-class thresholds filtering

ults of $Align(\tau_{metre})$ and $Original_k$ gives a p-value close to 0, indicating that performance is still significantly worse than the non key-invariant method.

Using the multinomial model, the best set of thresholds is:

$$(\tau_{hornpipe} = 0.45, \tau_{jig} = 0.90, \tau_{other44} = 0.45, \tau_{polka} = 0.35,$$
$$\tau_{reel} = 0.90, \tau_{slide} = 0.35, \tau_{slipjig} = 0.25, \tau_{waltz} = 0.35)$$

On the training set, this method designated by $Align(\tau_{tune\ type})$ results in 878 successful searches, 6.1% of the predictions are used and 94.65% of the search space is used on average. On the test set, it succeeds on 860 queries, 4 more than with the metre predictions. These results are also reported in Table 6.4. Wilcoxon tests between this new method and the two key-invariant methods $Align$ and All return p-values of, respectively, 0.044 and 0.0011, showing significant differences. However, neither the number of best hits nor the MRR improve, and the median a value is lower than that of *Align*. The performance also remains worse than $Original_k$, with a p-value close to 0.

Filtering on tune types instead of metre types could have been interesting in terms of pruning, as the corresponding sections of the search space are much smaller than the ones for metre types. However, for both the training and test sets, more than 94% of the search space had to be used on average, as only about 6% of the predictions on queries have been used. It appears that the thresholds found require such a high confidence that only a little computational efficiency is gained. The relatively poor accuracy of the multinomial model compared to the binomial one is a likely explanation for this.

As in previous chapter, the execution times for $Original_k$ and for the proposed method $Align(\tau_{metre})$ with the best thresholds found above are given in Table 6.5. Durations are given in CPU minutes, considering only `user` and `sys`. These only correspond to the 1000 queries of the test set, and as in 5.8.2 the execution time for the transcription of the queries is ignored. For the filtered results, execution times for the extraction of harmonic and rhythmic features, respectively audio PCH and quantised lag (*ql*) vectors, are added. Instead of adding an overhead to the $Original_k$ method, $Align(\tau_{metre})$ actually results in a shorter execution time.

Method	Step	CPU min.	Factor
$Original_k$	batch search	323.28	1
$Align(\tau_{metre})$	PCH computation	0.03	
	ql vectors computation	21.19	
	batch search	259.39	
	total	280.61	0.868

Table 6.5
Measured execution times

6.3 Discussion

Four different ways of integrating the rhythm predictions presented in Chapter 4 to the key-invariant tune recognition system have been discussed in this chapter.

The first strategy, presented in Section 6.1, consisted of ordering the result set on the edit distance weighted by a function of the probability estimates from the rhythm prediction models. Although it was possible to improve on the *Align* method from the previous chapter by using either $f(p) = \sqrt{1-p}$ or $f(p) = \sqrt{1-p^2}$ and the multinomial model, performances remained worse than the non key-invariant $Original_k$, and an additional computational overhead was incurred.

The other 3 methods consisted of using the rhythm predictions to restrict the search space, and consequently to reduce the computational cost. Only with the third one, presented in Section 6.2.3, was it possible to obtain results statistically similar to those of *Align* and *All*, by allowing the system to ignore the predictions below a certain confidence threshold. By using the metre type predictions in this way, $Align(\tau_{metre})$

runs faster than *Original$_k$* by more than 13%. Indeed, performing the actual search is the most time consuming step of the process, as it requires the computation of substring edit distance between the transcribed query and each item in the search space, of which there are more than 32,000 (see Section 5.2.1). Extracting the *ql* vectors on each excerpt is relatively fast, and the thresholds found for the binomial model predictions in the previous section allow to discard more than a third of the search space on average, as shown on Table 6.4.

To summarise, out of the several techniques attempted in this chapter, 2 stand out as improvements on *Align*:

- weighting edit distances by $f(p) = \sqrt{1-p^2}$ with the probability estimates from the multinomial model (*i.e.* tune type classification). This results in a more accurate key-invariant recognition system, but has a greater computational cost;
- pruning the search space when probability estimates from the binomial model exceed the thresholds $(\tau_{compound} = 0.75, \tau_{simple} = 0.65)$. This technique $Align(\tau_{metre})$ results in retrieval accuracy similar to *Align* and *All* and is more efficient than all the other methods, with an execution time shorter than *Original$_k$* by more than 13%.

A clear trade-off appears between efficiency and accuracy. Depending on the use-cases for a key-invariant tune recognition system, a choice can be made in favour of either solutions. For example, a tool on hand-held devices like Tunepal might put more priority on efficiency, while automatic analysis of archive recordings would likely favour accuracy.

Finally, as was discussed at the end of Chapter 4, a larger dataset of recordings categorised in rhythm types would help to train better models, and consequently affect the results presented in this chapter.

Chapter 7

Conclusion

This thesis has presented advancements in MIR methods for ITM, culminating in the development of a new key-invariant tune recognition system. Using harmonic and rhythmic features extracted from the audio queries, it performs as well as a brute force search considering all possible transpositions, and is faster than the existing, non key-invariant, system. This proposed method addresses one of the main limitations of the state-of-the-art query-by-playing software Tunepal. This contribution has the potential to not only extend the capabilities of Tunepal, but also to offer a better tool for annotating and exploring digital music archives.

Section 7.1 will summarise the main conclusions of the previous chapters. Then Section 7.2 will point out some limitations of the proposed methods and contributions. Finally, propositions for future work are discussed in Section 7.3.

7.1 Summary

After the necessary background was introduced in Chapter 2, Chapter 3 proposed an architecture for key-invariant tune recognition, relying on audio key inference. Subsequently, the rest of that chapter was a study on key detection for ITM, where the goal was to detect the musical key, *i.e.* tonic note and mode (major or minor), from a tune. A standard method, based on predefined key-profiles, was used. New key-profiles, that take into account the specificities of the ITM idiom, in particular its modal nature, were proposed. The results of the evaluations conducted indicate that these new key-profiles outperform the existing ones on ITM. However, in some cases defining the key of a tune can be ambiguous, and even the best performing key-profiles failed on over 20% of the audio recordings.

The study reported in Chapter 3 focused on the harmonic content of musical signal. In contrast, Chapter 4 presented a study about the temporal content, or rhythm, of audio recordings of tunes. Using *quantised lag vectors*, a tempo-invariant representation derived from the autocorrelation of an onset detection function, logistic regression models were trained to predict either metre types, *i.e.* beat subdivision, or tune types. Their classification accuracy reached, respectively, 96.8% and 83.2%. An ancillary research contribution in this chapter was the creation and presentation of a new set of annotations of 500 tune recordings, chosen to follow the distribution of tune types in reference collections.

As key inference proved to lack robustness, and to require key annotations that are sometimes hard to define, a new method for key-invariant tune recognition was introduced in Chapter 5. In this new *Align* technique, the symbolic representation is transposed to optimise the alignment of its pitch class histogram with that of the audio query. Substring edit distance is used for computing melodic similarity. Five AMT algorithms, and two ways of computing the audio PCH, were considered, and the best results were obtained with *Silvet* and PCH_{AMT}, computing the audio PCH from the transcribed notes. New metrics were defined to measure the performance of the system, and to compare its performance to 3 other methods: two non key-invariant ones, $Original_0$ and $Original_k$, the second one being informed by possible transpositions, and a brute force, key-invariant one, *All*, that computes melodic similarity with all possible transpositions. $Original_0$ and $Original_k$ can be considered as, respectively, the baseline and ceiling in this experiment. Both *Align* and *All* performed slightly worse (resp. 87.0% and 86.80%) than $Original_k$ (87.50%), but better than $Original_0$ (84.65%), and the differences are statistically significant. The difference between *Align* and *All* is also significant, in favour of *Align*: the proposed method performs slightly better than the brute force search, with a much lighter overhead in terms of execution time.

To improve further on this, Chapter 6 explored different ways of integrating the rhythm predictions from the models presented in Chapter 4 to the tune recognition system. By weighting the melodic similarity measure by a function of the probability estimates from the models, it was possible to improve on the performance of *Align*, despite remaining worse than the ceiling. A second method consisted in restricting the search space to the predicted rhythm categories. In this situation, errors in the rhythm prediction always cause the search to fail. Hence it was necessary to find thresholds for the probability estimates, interpreted as confidence scores, below which predictions were ignored. This method $Align(\tau_{metre})$ resulted in performances statistically similar to *Align* and *All*, with an execution time shorter than *Original*.

To conclude, this thesis has introduced improvements on key detection for Irish music, a novel method for classifying audio recordings in rhythm categories, and has proposed an efficient key-invariant tune recognition system. Contributions also include the sets of annotations prepared for these studies.

7.2 Limitations

First, limitations concerning the datasets and annotations prepared for the different studies in this thesis will be discussed. Manual annotations were carried out by a single annotator, the author of this thesis. A better practice would be to collect annotations from multiple annotators, as a way to reduce bias as well as errors in the annotations. This would be particularly interesting for key annotation: as was discussed in Chapter 3, the tonality of tunes can be ambiguous, and studying agreement between annotator (*e.g.* with Cohen's κ coefficient) could help quantify and understand this more closely.

The rhythm class of a tune can be subject to some ambiguity as well, and the methodology just discussed would also be of use to Chapter 4. Due to variations in the interpretation of tunes, tune ids introduced in Chapter 5 may also vary between annotators.

In Chapter 3, it was mentioned that the tonal centre of a melody can change between or even within parts. Annotating these changes would allow for a more precise study of tonality in ITM. However, ambiguity would still arise in some melodies, and, as practitioners typically do not refer to local tonal centres, but only of the key of a tune, obtaining such annotations from multiple annotators would be difficult.

Another important limitation is the representation of instruments in the main dataset, used in Chapter 4 onwards. Recordings were collected from the Comhaltas archives, while respecting the tune type distribution as explained in Section 4.2. Consequently the distribution of recordings per instruments was simply a result of their availability in the archives, and the dataset is unbalanced in this regard. For example, there are only 3 recordings on solo banjo and solo harp, but 20 on solo fiddle, and the error analysis conducted in Section 5.8.3.1 showed that tune recognition accuracy varied across these categories. A dataset more balanced in terms of instruments would allow for a more systematic study of the performance of the different components of the system (music transcription, pitch class histogram alignment, rhythm classification). A way to obtain such a dataset would be to carefully decide on the distribution of tune types and instruments before organising recording sessions specifically for this purpose. Data augmentation techniques could also be used to reach a better balance in the dataset.

Then, limitations of the methods developed in this thesis are discussed. In Chapters 4 and 5, the *fuzzy histogram* is used to find the quaver duration q, which is then used to quantise the lag vectors or the transcribed notes, respectively. Subsequently, only the quantised representations are used, and q is effectively discarded. As it gives an indication of tempo, it could have been beneficial to use q for rhythm classification, and potentially improve results in Chapters 4 and 6.

Microtonality in ITM was discussed in Section 2.2, but the PCH alignment method presented in Section 5.4 aims at aligning peaks of the audio PCH to those of the symbolic PCH, which are always well-tempered. This important difference in pitch content between audio recordings and symbolic representations suggests that the PCH alignment method could be improved.

Finally, some limitations about the results analysis are now discussed. All the evaluations, statistical analysis, and error analysis in the thesis have been conducted at the level of the dataset, or subsets of it. If considering sets of examples rather than individual ones is necessary for discussing statistical significance and general observations, conducting finer error analysis at the level of individual instances could have provided valuable insights into the strengths and weaknesses of the proposed methods.

Standard metrics, as defined by MIREX, have been used, and their limitations have been discussed already, in particular in Chapter 3. They are symptomatic of the difficulty of reconciling the need for standard metrics, necessary for comparative studies, and the specificities of distinct musical idioms.

7.3 Future work

In addition to the suggestions regarding datasets, annotations, methods, and results analysis discussed above, this section outlines possible future work building on this thesis.

First, making the proposed MIR methods available for end users and musical archives would be an important step. Integrating them in Tunepal would both offer an improved tune recognition tool for musicians, and help disseminate the product of this research.

Finally, some of the features extracted from audio recordings could be used to develop a better automatic music transcription (AMT) algorithm for ITM, which is an important step for tune recognition. Indeed, the study presented in Appendix A showed that the accuracy of note-level automatic transcription on session recordings only reaches a F-measure of 0.75. By using automatically extracted harmonic and rhythmic features to guide an AMT algorithm, more accurate transcription may be obtained.

Appendix A

A corpus of annotated Irish traditional music recordings

In this Appendix, a dataset of manual pitch-track annotations, published in Beauguitte et al. (2016a), is presented. The aim of this preliminary study was first to evaluate the performance of existing automatic music transcription (AMT) algorithms on recordings of ITM, but also to offer a set of data useable for training ML models. Köküer et al. (2019) have since undertaken a similar annotation effort, aiming at stylistic analysis.

A.1 Presentation of the dataset

In this section, the design of the corpus is presented. Section A.1.1 introduces the set of audio recordings that was chosen to make the corpus representative of Irish traditional dance music. The annotation format used to deal with the characteristics of these recordings is then detailed in A.1.2.

A.1.1 Source of audio recordings

Three sources of recordings are included in the corpus:

- session recordings accompanying the Foinn Seisiún books published by the Comhaltas Ceoltóirí Éireann organisation, available with Creative Commons licence. Instruments in the recordings include flute, tin whistle, uillean pipes (Irish bagpipes), accordion, concertina, banjo, piano, guitar, bodhran (drum). These offer good quality, homogeneous examples of the heterophony inherent to an ITM session.
- Grey Larsen's *MP3s for 300 Gems of Irish Music for All Instruments*, commercially available. These consist of studio quality recordings of tunes played on Irish flute, tin whistle and concertina.
- Personal recordings of Bryan Duggan on the Irish flute. These are available together with the annotations.

The corpus comprises 30 tunes in total, which add up to more than 30 minutes of audio. It was decided to include both solo and session recordings as a way of comparing the performance of transcription algorithms on respectively monophonic and heterophonic music. Table A.1 categorises the tunes in the corpus by tune type and performance type. The complete list of tunes with the relevant metadata is included with the dataset.

	Reel	Jig	Hornpipe	Polka	Slide	Air
Session	5	5	1	1	1	0
Solo	4	4	3	3	2	1

Table A.1
Classification of tunes in the corpus by tune type and performance type

A.1.2 Format of the annotations

Each audio recording is annotated with note events, consisting of three values: (pitch, onset, duration). For the goal of obtaining a symbolic transcription, this format of annotation is more useful as well as easier to manually annotate than a continuous pitch track labelling every audio frame. The session recordings are heterophonic, but only interested in the underlying melodic line shared by all the instrumentalists is considered. For this reason there is no overlap between the notes, and the resulting transcriptions are monophonic.

Due to the heterophonic nature of Irish traditional music as played during a session, and to the slight tuning differences between the instruments, a single fundamental frequency cannot appropriately describe a note. Therefore it was decided to report only MIDI note references instead of frequencies.

In session performances, the playing of ornamentation such as *rolls* and *cuts* (Vallely, 2011) often results in several successive onsets for a single long note. Figure A.1 shows an example of such a situation, where three instruments interpret differently the same note (the short notes played by the flute are part of a *roll* and are not melodically signi-

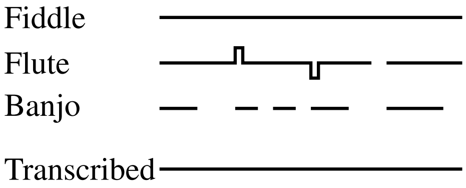


Figure A.1
Example of ornamented notes on different instruments

ficant, therefore they are not transcribed). This makes it difficult, even for experienced musicians and listeners, to know when repeated notes are to be considered as a single ornamented note or as distinct notes. Because of this inherent ambiguity, it is not appropriate to associate one onset with one note in the transcription. For this reason, consecutive notes of identical pitch are merged into one single note. A change of note in the annotations thus corresponds to a change of pitch in the melody.

In solo performances, there are some clear silences between notes, typically when the flute or whistle player takes a breath. Whenever such a silence occurs, two distinct notes were annotated even if they are of the same pitch. In the solo recordings present in the corpus, the typical duration of a breath was measured at around 200ms. Notes repeated without pauses or cut by an ornament were still reported as a single note, in order to be consistent with the annotations of session recordings.

Manual annotations were made with the aid of the Tony software Mauch et al. (2015). After importing an audio file, Tony offers estimates using the pYIN algorithm, already presented in Section 5.3.4.

These were then manually corrected, by adding new notes, merging repeated notes and adjusting frequencies. The annotations were finally post-processed to convert these frequencies to the closest MIDI note references. With this annotation format, the dataset comprises in total more than 8600 notes.

A.1.3 Open publication of the dataset

The dataset is publicly available as a set of csv files.[1] Each file contains the annotation for an entire audio file. Each line represents a note (as onset time, duration, MIDI note). The annotations can be easily used in any evaluation framework, *e.g.* `mir_eval` (Raffel et al., 2014), as well as with the software Sonic Visualiser (Cannam et al., 2010).

A.2 Results of AMT algorithms

In this Section, the transcriptions obtained by the AMT algorithms introduced in Section 5.3.4 are evaluated on the dataset. In order to be consistent with the annotation format presented above, it is necessary to post-process the estimates of these algorithms in the following manner:

- align all frequencies to the closest MIDI note;
- for note-level estimates, merge consecutive notes of same pitch separated by less than 200ms.

[1] `https://github.com/pierrebeauguitte/tuneset`

The second step is particularly critical for the note-level metrics of the MIREX Note Tracking task (see below Section A.2.2), but will also affect the frame-level metrics of the Melody Extraction tasks for the frames in the filled gaps.

Evaluations are performed with the `mir_eval` framework (Raffel et al., 2014).[2]

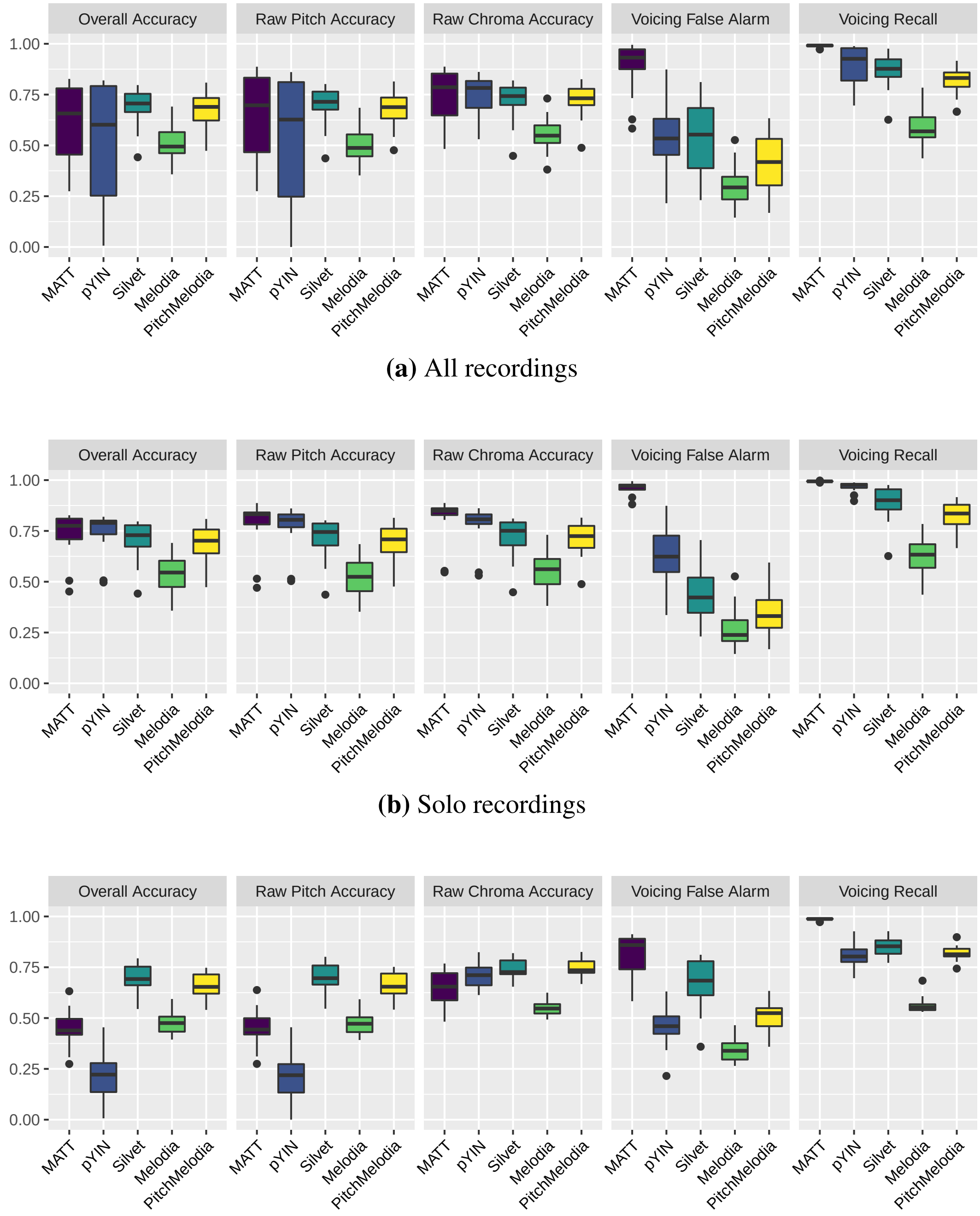
A.2.1 Frame-level evaluation: Melody Extraction

The MIREX Melody Extraction task evaluates transcriptions at frame level. Pre-processing of both the ground truth and the estimates is necessary, and simply consists of aligning both on the same 10ms time grid. The pitch estimate for a frame is considered correct if it is distant from the ground truth by less than a quarter of a tone (50 cents). The metrics also look for voicing errors: a voiced frame is one where a melody pitch is present. Five different metrics are computed for each tune. Results are shown in Figure A.2.

A.2.2 Note-level evaluation: Note Tracking

This section presents the results of the MIREX Note Tracking task. Although this task is primarily aiming at polyphonic music transcription systems, it also applies directly to monophonic music as long as both ground truth annotations and returned estimates are in a note-event

[2] `https://github.com/craffel/mir_eval`



(a) All recordings

(b) Solo recordings

(c) Session recordings

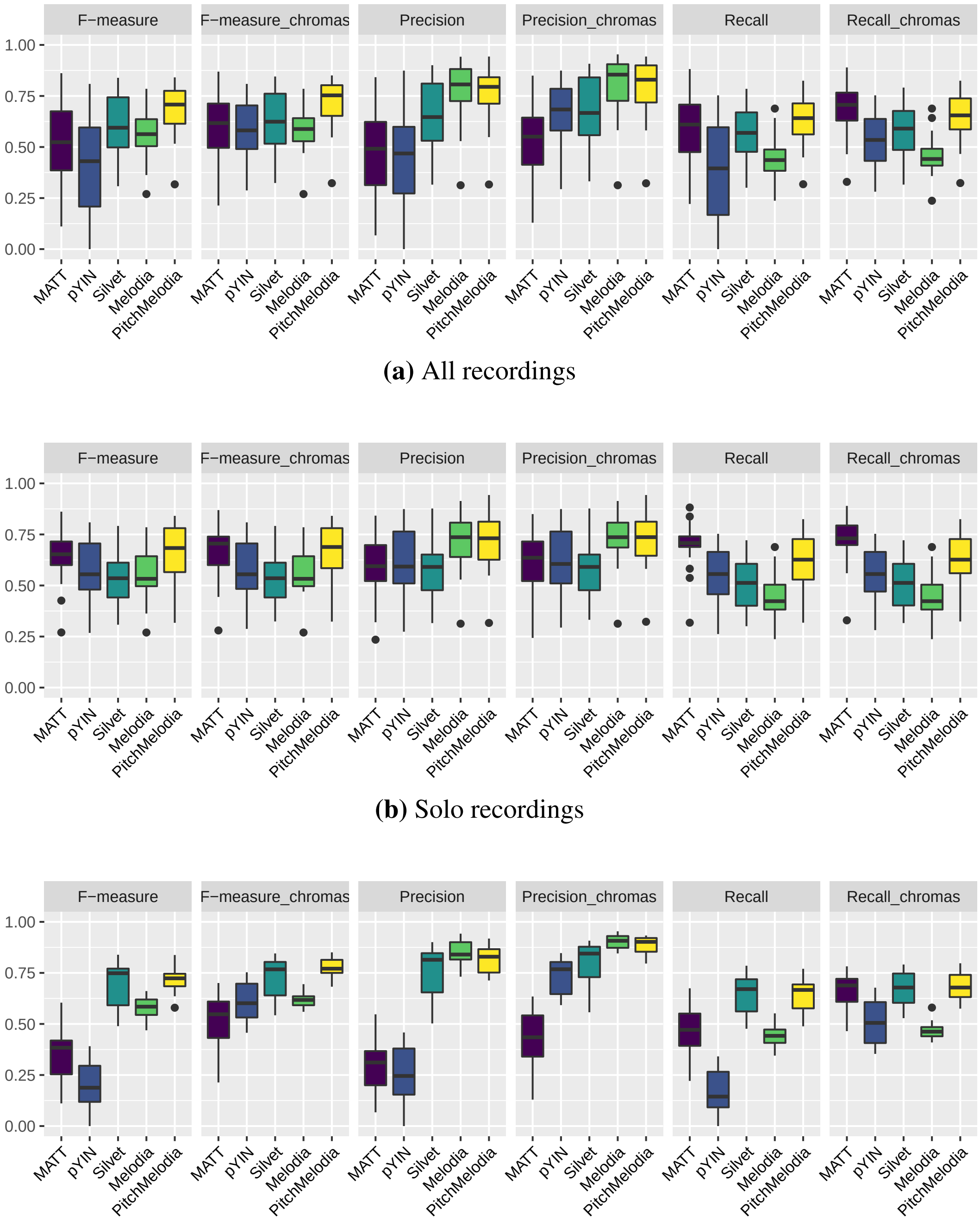
Figure A.2
Scores of the MIREX Melody Extraction task evaluation

format. MATT2, pYIN and Silvet already return transcriptions in this format. Melodia and PitchMelodia have to be post-processed by the PitchContourSegmentation algorithm of the Essentia library (Bogdanov et al., 2013).

Estimated notes are associated with their closest match from the reference annotation, and a note is considered correctly transcribed if its onset is distant from the reference note by less than 50ms and its pitch by less than a quarter of a tone. Another way of evaluating the transcription is to also take the duration of notes into account. Commonly found instruments in Irish music have a wide range of acoustical characteristics: some (like the flute, the fiddle, the uilleann pipes) can be played *legato* or *staccato*, depending on personal or regional styles, or on the type of tunes performed; others (typically the banjo) can only be played *staccato*, with hard onsets and very little sustain. Consequently, the offset of the notes is of little interest for the present evaluations, particularly in session recordings where all these instruments can play together. This is why only results for the first type of evaluation are reported.

Precision, recall and F-measures are computed with the `mir_eval` framework (Raffel et al., 2014), and plotted in Figure A.3. Results for the Chroma correctness, where a note is correctly transcribed if its onset (±50ms) and its pitch class are correct, are also shown.

A statistical analysis comparing the performances of the different AMT algorithms on these 2 tasks was given in Beauguitte et al. (2016a).



(a) All recordings

(b) Solo recordings

(c) Session recordings

Figure A.3
Scores of the MIREX Note Tracking task evaluation

Here, it will simply be observed that for the Note Tracking task, which corresponds to the way audio queries are transcribed in Chapters 5 and 6 (see Section 5.3.4), Silvet obtains the highest F-measure on session recordings, followed by PitchMelodia and Melodia. This is consistent with the performances observed for these algorithms in the tune recognition task (see Section 5.8.1). Although MATT2 obtains a higher F-measure, its precision on chromas is much lower than that of pYIN, which explains why it performed worse in the tune recognition task.

Appendix B

Confusion matrices for key detection

This Appendix gives the the confusion matrices obtained in the second experiment for key detection in Chapter 3, Section 3.4.2.

	DM	*GM*	*Am*	*Em*	*AM*	*Bm*	*CM*	*Dm*
DM	149	1	0	0	0	0	0	0
GM	4	119	0	0	0	0	0	0
Am	5	9	7	0	0	0	1	1
Em	8	3	1	3	0	0	0	0
AM	2	0	0	0	9	0	0	0
Bm	2	0	0	0	0	1	0	0
CM	0	0	0	0	0	0	1	0

Correct	289
Fifth	6
Relative	6
Parallel	0
Neighbour	17
Other	8

Table B.1
Confusion matrix for FS_{audio} Cadences*(3, 1, 2)

	DM	GM	Am	Em	AM	Bm	CM	F#m	BM
DM	131	3	1	0	0	14	0	1	0
GM	0	113	1	4	0	1	0	0	0
Am	0	7	11	1	4	0	0	0	0
Em	1	1	0	11	0	2	0	0	0
AM	0	0	0	0	11	0	0	0	0
Bm	0	0	0	0	0	2	0	0	1
CM	0	0	0	0	0	0	1	0	0

Correct	280
Fifth	0
Relative	19
Parallel	5
Neighbour	9
Other	9

Table B.2
Confusion matrix for FS_{symb} with Lerdahl*

	*D*M	*G*M	*A*m	*E*m	*A*M	*C*M	*D*m	*B*m	*G*m	*F*M
*D*M	82	4	6	1	1	0	4	0	0	0
*G*M	6	71	4	1	0	0	0	0	0	0
*A*m	5	9	17	0	1	3	2	0	0	0
*E*m	10	6	5	13	2	0	0	0	0	0
*A*M	7	0	2	0	11	0	1	0	0	0
*C*M	0	1	2	0	0	5	1	0	0	0
*D*m	1	0	2	0	0	2	3	0	0	0
*B*m	3	0	0	1	0	0	0	0	0	0
*G*m	0	0	0	0	0	2	0	0	2	0
*F*M	0	0	0	0	0	0	0	0	0	1

Correct	205
Fifth	16
Relative	15
Parallel	8
Neighbour	27
Other	29

Table B.3
Confusion matrix for GL_{audio} with Cadences*(3, 1, 2)

	*D*M	*G*M	*A*m	*E*m	*A*M	*C*M	*D*m	*B*m	*G*m	*F*M	*F*#m
*D*M	85	1	5	1	0	0	0	6	0	0	0
*G*M	2	71	1	7	0	0	0	1	0	0	0
*A*m	1	13	18	3	1	1	0	0	0	0	0
*E*m	3	1	0	26	1	0	0	5	0	0	0
*A*M	4	0	1	0	14	0	0	0	0	0	2
*C*M	0	1	4	0	0	4	0	0	0	0	0
*D*m	0	0	1	0	0	1	6	0	0	0	0
*B*m	0	0	0	0	0	0	0	4	0	0	0
*G*m	0	0	0	0	0	0	0	0	3	1	0
*F*M	0	0	0	0	0	0	1	0	0	0	0

Correct	231
Fifth	3
Relative	22
Parallel	2
Neighbour	20
Other	22

Table B.4
Confusion matrix for GL_{symb} with Lerdahl*

Appendix C

Links to implementations and datasets

This Appendix lists the URLs for the datasets and implementations for the contributions presented of this thesis.

Chapter 3. Key detection

`https://github.com/pierrebeauguitte/keydetection`

Chapter 4. Rhythm classification

`https://github.com/pierrebeauguitte/ITM_rhythm`

Chapter 5 & 6. Tune recognition

`https://github.com/pierrebeauguitte/tunerecognition`

Appendix A

`https://github.com/pierrebeauguitte/tuneset`

Bibliography

Ali-MacLachlan, I., Köküer, M., Athwal, C., and Jančovič, P. (2015). Towards the Identification of Irish Traditional Flute Players from Commercial Recordings. In *Proceedings of the 5th International Workshop on Folk Music Analysis*, pages 13–17, Paris, France.

Ali-MacLachlan, I., Southall, C., Tomczak, M., and Hockman, J. (2018). Player Recognition for Traditional Irish Flute Recordings. In *Proceedings of the 8th International Workshop on Folk Music Analysis*, pages 3–8, Thessaloniki, Greece.

Beauguitte, P. (2017). The AEPEM Collection: A Set of Annotated Traditional French Music Scores. In *Proceedings of the 7th International Workshop on Folk Music Analysis*, pages 123–124, Málaga, Spain.

Beauguitte, P., Duggan, B., and Kelleher, J. (2016a). A Corpus of Annotated Irish Traditional Dance Music Recordings: Design and Benchmark Evaluations. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, New York, USA.

Beauguitte, P., Duggan, B., and Kelleher, J. (2017). Key Inference from

Irish Traditional Music Scores and Recordings. In *Proceedings of the 14th Sound and Music Computing Conference*, pages 85–91, Espoo, Finland.

Beauguitte, P., Duggan, B., and Kelleher, J. (2018). Rhythm Inference From Audio Recordings of Irish Traditional Music. In *Proceedings of the 8th International Workshop on Folk Music Analysis*, pages 9–13, Málaga, Spain.

Beauguitte, P., Duggan, B., and Kelleher, J. D., editors (2016b). *Proceedings of the 6th International Workshop on Folk Music Analysis*. Dublin, Ireland.

Beauguitte, P. and Huang, H.-C. (2019). Content-based Music Retrieval of Irish Traditional Music Via a Virtual Tin Whistle. In *Proceedings of the 9th International Workshop on Folk Music Analysis*, Birmingham, UK.

Bello, J., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. B. (2005). A Tutorial on Onset Detection in Music Signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047.

Benetos, E. and Dixon, S. (2012). A Shift-Invariant Latent Variable Model for Automatic Music Transcription. *Computer Music Journal*, 36(4):81–94.

Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., and Klapuri, A. (2013). Automatic Music Transcription: Challenges and Future Directions. *Journal of Intelligent Information Systems*, 41(3):407–434.

Benetos, E. and Holzapfel, A. (2015). Automatic Transcription of Turkish Microtonal Music. *The Journal of the Acoustical Society of America*, 138(4):2118–2130.

Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F., and Widmer, G. (2016). Madmom: A New Python Audio and Music Signal Processing Library. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 1174–1178, Amsterdam, The Netherlands.

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J. R., and Serra, X. (2013). ESSENTIA: An Audio Analysis Library for Music Information Retrieval. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 493–498, Curitiba, Brazil.

Breathnach, B. (1996). *Folk Music and Dances of Ireland.* Ossian, Cork, Ireland. (original work published 1971).

Brown, J. C. (1993). Determination of the Meter of Musical Scores by Autocorrelation. *The Journal of the Acoustical Society of America*, 94(4):1953–1957.

Cannam, C., Landone, C., and Sandler, M. (2010). Sonic Visualiser:

An Open Source Application for Viewing, Analysing, and Annotating Music Audio Files. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 1467–1468, Firenze, Italy.

Chen, N., Li, W., and Xiao, H. (2018). Fusing Similarity Functions for Cover Song Identification. *Multimedia Tools and Applications*, 77(2):2629–2652.

Chordia, P. and Şentürk, S. (2013). Joint Recognition of Raag and Tonic in North Indian Music. *Computer Music Journal*, 37(3):82–98.

Dannenberg, R. B., Birmingham, W. P., Pardo, B., Hu, N., Meek, C., and Tzanetakis, G. (2007). A Comparative Evaluation of Search Techniques for Query-by-Humming Using the MUSART Testbed. *Journal of the American Society for Information Science and Technology*, 58(5):687–701.

de Cheveigné, A. and Kawahara, H. (2002). YIN, a Fundamental Frequency Estimator for Speech and Music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930.

De Haas, W. B. and Volk, A. (2016). Meter Detection in Symbolic Music Using Inner Metric Analysis. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 441–447, New York, USA.

de Valk, R., Volk, A., Holzapfel, A., Pikrakis, A., Kroher, N., and Six,

J. (2017). MIRchiving: Challenges and Opportunities of Connecting MIR Research and Digital Music Archives. In *Proceedings of the 4th International Workshop on Digital Libraries for Musicology*, pages 25–28, Shanghai, China.

Doherty, E. (1996). *The Paradox of the Periphery: Evolution of the Cape Breton Fiddle Tradition C1928-1995*. PhD thesis, Univeristy of Limerick, Ireland.

Downie, J. S. (2003). Music Information Retrieval. *Annual Review of Information Science and Technology*, 37(1):295–340.

Duggan, B. (2009). *Machine Annotation of Traditional Irish Dance Music*. PhD thesis, Dublin Institute of Technology, Ireland.

Duggan, B. (2010). Tunepal: The Traditional Musician's Toolbox. In *Proceedings of the Second Workshop on eHeritage and Digital Art Preservation*, pages 25–30, Firenze, Italy.

Duggan, B. and O'Shea, B. (2011). Tunepal: Searching a Digital Library of Traditional Music Scores. *OCLC Systems & Services: International Digital Library Perspectives*, 27(4):284–297.

Duggan, B., O'Shea, B., Gainza, M., and Cunningham, P. (2008). Machine Annotation of Sets of Traditional Irish Dance Tunes. In *Proceedings of the 9th International Society for Music Information Retrieval Conference*, pages 401–406, Philadelphia, USA.

Duggan, B., O'Shea, B., Gainza, M., and Cunningham, P. (2009). Compensating for Expressiveness in Queries to a Content Based Music Information Retrieval System. In *Proceedings of the International Computer Music Conference*, pages 33–36, Montreal, Canada.

Duggan, B., Xu, J., Denbrok, L., and Knowlton, B. (2016). A Search Through Time: Connecting Live Playing to Archive Recordings of Traditional Music. In *Proceedings of the 6th International Workshop on Folk Music Analysis*, pages 64–66, Dublin, Ireland.

Ellis, D. P. and Poliner, G. E. (2007). Identifying Cover Songs With Chroma Features and Dynamic Programming Beat Tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1429–1432, Honolulu, Hawaii.

Fouloulis, T., Pikrakis, A., and Cambouropoulos, E. (2013). Traditional Asymmetric Rhythms: A Refined Model of Meter Induction Based on Asymmetric Meter Templates. In *Proceedings of the 3rd International Workshop on Folk Music Analysis*, pages 28–32, Amsterdam, Netherlands.

Futrelle, J. and Downie, J. S. (2002). Interdisciplinary Communities and Research Issues in Music Information Retrieval. In *Proceedings of the 3rd International Society for Music Information Retrieval Conference*, pages 215–221, Paris, France.

Gainza, M. (2009). Automatic Musical Meter Detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 329–332, Taipei, Taiwan.

Gainza, M. and Coyle, E. (2007). Time Signature Detection by Using a Multi-Resolution Audio Similarity Matrix. In *Proceedings of the 122nd Convention of the Audio Engineering Society*, Vienna, Austria.

Gómez, E. (2006). Tonal Description of Polyphonic Audio for Music Content Processing. *INFORMS Journal on Computing*, 18(3):294–304.

Gouyon, F. and Herrera, P. (2003). Determination of the Meter of Musical Audio Signals: Seeking Recurrences in Beat Segment Descriptors. In *Proceedings of the 114th Convention of the Audio Engineering Society*, Amsterdam, The Netherlands.

Jančovič, P., Köküer, M., and Wrena, B. (2015). Automatic Transcription of Ornamented Irish Traditional Flute Music Using Hidden Markov Models. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, pages 756–762, Málaga, Spain.

Janssen, B., van Kranenburg, P., and Volk, A. (2017). Finding Occurrences of Melodic Segments in Folk Songs Employing Symbolic Similarity Measures. *Journal of New Music Research*, 46(2):118–134.

Jehan, T. (2005). *Creating Music by Listening*. PhD thesis, Massachusetts Institute of Technology, USA.

Kelleher, J. D., Mac Namee, B., and D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. The MIT Press, Cambridge, USA.

Kelly, C., Gainza, M., Dorran, D., and Coyle, E. (2010). Locating Tune Changes and Providing a Semantic Labelling of Sets of Irish Traditional Tunes. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 129–134, Utrecht, Netherlands.

Kim, J. W., Salamon, J., Li, P., and Bello, J. P. (2018). Crepe: A Convolutional Representation for Pitch Estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165, Calgary, Canada. IEEE.

Köküer, M., Ali-MacLachlan, I., Kearney, D., and Jančovič, P. (2019). Curating and Annotating a Collection of Traditional Irish Flute Recordings to Facilitate Stylistic Analysis. *International Journal on Digital Libraries*, 20(1):107–121.

Korzeniowski, F. and Widmer, G. (2016). Feature Learning for Chord Recognition: The Deep Chroma Extractor. In *Proceedings of the*

17th International Society for Music Information Retrieval Conference, pages 37–43, New York, USA.

Krumhansl, C. L. (1990). *Cognitive Foundations of Musical Pitch*. Oxford Psychology Series. Oxford University Press, Oxford, UK.

Larsen, G. (2013). *300 Gems of Irish Music for All Instruments*. Mel Bay Publications, Fenton, USA.

Lee, J., Chang, S., Choe, S. K., and Lee, K. (2018). Cover Song Identification Using Song-to-Song Cross-Similarity Matrix with Convolutional Neural Network. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 396–400, Calgary, Canada.

Leman, M. (1995). *Music and Schema Theory*, volume 31 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg.

Lemström, K. and Perttu, S. (2000). SEMEX - An efficient Music Retrieval Prototype. In *Proceedings of the 1st International Symposium on Music Information Retrieval*, Plymouth, USA.

Lerdahl, F. (1988). Tonal Pitch Space. *Music Perception: An Interdisciplinary Journal*, 5(3):315–349.

Madsen, S. T. and Widmer, G. (2007). Key-Finding with Interval Profiles. In *Proceedings of the 33th International Computer Music Conference*, pages 212–215, Copenhagen, Denmark.

Martiniano, L. and Silla, C. (2017). BIRITS: A Music Information Retrieval System Using Query-by-Playing Techniques. In *Proceedings of the 29th IEEE International Conference on Tools with Artificial Intelligence*, pages 535–542, Boston, USA.

Mauch, M., Cannam, C., Bittner, R., Fazekas, G., Salamon, J., Dai, J., Bello, J., and Dixon, S. (2015). Computer-Aided Melody Note Transcription Using the Tony Software: Accuracy and Efficiency. In *Proceedings of the First International Conference on Technologies for Music Notation and Representation*, pages 23–31, Paris, France.

Mauch, M. and Dixon, S. (2014). pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 659–663, Firenze, Italy.

McLeod, A. and Steedman, M. (2017). Meter Detection in Symbolic Music Using a Lexicalized PCFG. In *Proceedings of the 14th Sound and Music Computing Conference*, pages 373–379, Espoo, Finland.

McNab, R. J., Smith, L. A., Witten, I. H., Henderson, C. L., and Cunningham, S. J. (1996). Towards the Digital Music Library: Tune Retrieval From Acoustic Input. In *Proceedings of the First ACM International Conference on Digital Libraries*, pages 11–18, Bethesda, USA. ACM.

Molloy, R. (2017). Microinterval Modality in Traditional Irish Music: An Empirical Approach. In *Proceedings of the 7th International Workshop on Folk Music Analysis*, Málaga, Spain.

Moylan, T. and O'Leary, J., editors (2014). *Johnny O'Leary of Sliabh Luachra: Dance Music from the Cork-Kerry Border*. The Lilliput Press, Dublin, Ireland.

Navarro, G. and Raffinot, M. (2002). *Flexible Pattern Matching in Strings: Practical on-Line Search Algorithms for Texts and Biological Sequences*. Cambridge University Press, Cambridge, UK.

Noland, K. and Sandler, M. B. (2006). Key Estimation Using a Hidden Markov Model. In *Proceedings of the 7th International Society for Music Information Retrieval Conference*, pages 121–126, Victoria, Canada.

Ó Canainn, T. (1993). *Traditional Music in Ireland.* Ossian, Cork, Ireland. (original work published 1978).

Ó hAllmhuráin, G. (2004). *O'Brien Pocket History of Irish Traditional Music*. O'Brien, Dublin, Ireland. (original work published 1998).

O'Shea, H. (2008). *The Making of Irish Traditional Music*. Cork University Press, Cork, Ireland.

Pikrakis, A., Antonopoulos, I., and Theodoridis, S. (2004). Music Meter and Tempo Tracking From Raw Polyphonic Audio. In *Proceedings of* *the 5th International Society for Music Information Retrieval Conference*, Barcelona, Spain.

Pikrakis, A., Kroher, N., and Días-Báñez, J.-M. (2016). Detection of Melodic Patterns in Automatic Flamenco Transcriptions. In *Proceedings of the 6th International Workshop on Folk Music Analysis*, pages 14–17, Dublin, Ireland.

Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and Ellis, D. P. (2014). Mir_eval: A Transparent Implementation of Common MIR Metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 367–372, Taipei, Taiwan.

Salamon, J. and Gómez, E. (2012). Melody Extraction From Polyphonic Music Signals Using Pitch Contour Characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770.

Salamon, J., Gomez, E., Ellis, D. P., and Richard, G. (2014). Melody Extraction from Polyphonic Music Signals: Approaches, Applications, and Challenges. *Signal Processing Magazine*, 31(2):118–134.

Salamon, J., Serrà, J., and Gómez, E. (2013). Tonal Representations for Music Retrieval: From Version Identification to Query-by-Humming. *International Journal of Multimedia Information Retrieval*, 2(1):45–58.

Sellers, P. H. (1980). The Theory and Computation of Evolutionary Distances: Pattern Recognition. *Journal of Algorithms*, 1(4):359–373.

Serrà, J., Gómez, E., and Herrera, P. (2008). Transposing Chroma Representations to a Common Key. In *IEEE CS Conference on The Use of Symbols to Represent Music and Multimedia Objects*, pages 45–48.

Serrà, J., Gómez, E., and Herrera, P. (2010). Audio Cover Song Identification and Similarity: Background, Approaches, Evaluation, and Beyond. In *Advances in Music Information Retrieval*, volume 274, pages 307–332. Springer, Berlin, Heidelberg.

Stasiak, B. (2014). Follow That Tune-Adaptive Approach to DTW-based Query-by-Humming System. *Archives of Acoustics*, 39(4):467–476.

Sturm, B. and Ben-Tal, O. (2018). Let's Have Another Gan Ainm: An Experimental Album of Irish Traditional Music and Computer-Generated Tunes. Technical report, KTH Royal Institute of Technology, Stockholm, Sweden.

Sturm, B., Santos, J. F., and Korshunova, I. (2015). Folk Music Style Modelling by Recurrent Neural Networks With Long Short Term Memory Units. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (Late-Breaking Demo Session)*, Málaga, Spain.

Su, N. M. and Duggan, B. (2014). TuneTracker: Tensions in the Surveillance of Traditional Music. In *Proceedings of the ACM Conference on Designing Interactive Systems*, pages 845–854, Vancouver, Canada.

Temperley, D. (2001). *The Cognition of Basic Musical Structures*. MIT Press, Cambridge, Mass.

Toiviainen, P. and Eerola, T. (2006). Autocorrelation in Meter Induction: The Role of Accent Structure. *The Journal of the Acoustical Society of America*, 119(2):1164–1170.

Tzanetakis, G., Kapur, A., Schloss, W. A., and Wright, M. (2007). Computational Ethnomusicology. *Journal of Interdisciplinary Music Studies*, 1(2):1–24.

Vallely, F. (2011). *The Companion to Irish Traditional Music*. Cork University Press, Cork, Ireland, second edition.

Varewyck, M., Martens, J.-P., and Leman, M. (2013). Musical Meter Classification with Beat Synchronous Acoustic Features, DFT-based Metrical Features and Support Vector Machines. *Journal of New Music Research*, 42(3):267–282.

Wagner, R. A. and Fischer, M. J. (1974). The String-to-String Correction Problem. *Journal of the ACM*, 21(1):168–173.

Wallace, B. (2018). SoundTracer: A Brief Project Summary. Technical report, University of Oslo, Oslo, Norway.

Weissenberger, L. (2017). Stories, Songs, Steps, and Tunes: A Linked Data Ontology for Irish Traditional Music and Dance. In *International Society for Knowledge Organization*, London, United Kingdom.

Williams, S. (2013). *Focus: Irish Traditional Music*. Routledge, New York, USA.

Wu, S. and Manber, U. (1992). Fast Text Searching: Allowing Errors. *Communications of the ACM*, 35(10):83–91.