Polifonia

Polifonia: a digital harmoniser for musical heritage knowledge, H2020

D3.2: Analysis of music repositories to identify musical patterns (V1.0)

Deliverable information	
WP	WP3: Mining Musical Patterns
Deliverable dissemination level	PU Public
Deliverable type	R Document, report
Lead beneficiary	NUIG
Contributors	UNIBO, CNRS, CNAM, KCL, OU, SORBONNE, KNAW
Document status	Final
Document version	V1.0
Date	December 20, 2021
Authors	James McDermott (NUIG) and Danny Diamond (NUIG) and Abdul Shahid Khattak (NUIG) and Peter van Krannenburg (KNAW) and Jacopo de Berardinis (KCL) and Andrea Poltronieri (UNIBO) and Albert Meroño Peñuela (KCL) and Raphaël Fournier-S'niehotta (CNAM) and Tiange Zhu (CNAM)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004746

PAGE INTENTIONALLY BLANK

Project Information

Project Start Date: 1st January 2021
Project Duration: 40 months
Project Website: https://polifonia-project.eu

Project Contacts

Project Coordinator

Valentina Presutti

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
Department of Language, Literature and Modern Cultures (LILEC)

E-mail: valentina.presutti@unibo.it

Project Manager

Marta Clementi

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
Research division

E-mail: marta.clementi3@unibo.it

POLIFONIA Consortium

No.	Short name	Institution name	Country
1	UNIBO	ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA	Italy
2	OU	THE OPEN UNIVERSITY	United Kingdom
3	KCL	KING'S COLLEGE LONDON	United Kingdom
4	NUI GALWAY	NATIONAL UNIVERSITY OF IRELAND GALWAY	Ireland
5	MiC	MINISTERO DELLA CULTURA	Italy
6	CNRS	CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE CNRS	France
	SORBONNE	SORBONNE UNIVERSITE (LinkedTP)	France
7	CNAM	CONSERVATOIRE NATIONAL DES ARTS ET METIERS	France
8	NISV	STICHTING NEDERLANDS INSTITUUT VOOR BEELD EN GELUID	Netherlands
9	KNAW	KONINKLIJKE NEDERLANDSE AKADEMIE VAN WETENSCHAPPEN	Netherlands
10	DP	DIGITAL PATHS	Italy

Project Summary

European musical heritage is a dynamic historical flow of experiences, leaving heterogeneous traces that are difficult to capture, connect, access, interpret, and valorise. Computing technologies have the potential to shed a light on this wealth of resources by extracting, materialising and linking new knowledge from heterogeneous sources, hence revealing facts and experiences from hidden voices of the past. Polifonia makes this happen by building novel ways of inspecting, representing, and interacting with digital content. Memory institutions, scholars, and citizens will be able to navigate, explore, and discover multiple perspectives and stories about European Musical Heritage.

Polifonia focuses on European Musical Heritage, intended as musical contents and artefacts - or music objects - (tunes, scores, melodies, notations, etc.) along with relevant knowledge about them such as: their links to tangible objects (theatres, conservatoires, churches, etc.), their cultural and historical contexts, opinions and stories told by people having diverse social and artistic roles (scholars, writers, students, intellectuals, musicians, politicians, journalists, etc), and facts expressed in different styles and disciplines (memoire, reportage, news, biographies, reviews), different languages (English, Italian, French, Spanish, and German), and across centuries.

The overall goal of the project is to realise an ecosystem of computational methods and tools supporting discovery, extraction, encoding, interlinking, classification, exploration of, and access to, musical heritage knowledge on the Web. An equally important objective is to demonstrate that these tools improve the state of the art of Social Science and Humanities (SSH) methodologies. Hence their development is guided by, and continuously intertwined with, experiments and validations performed in real-world settings, identified by musical heritage stakeholders (both belonging to the Consortium and external supporters) such as cultural institutes and collection owners, historians of music, anthropologists and ethnomusicologists, linguists, etc.

Executive Summary

Deliverable D3.2 is titled: *Analysis of music repositories to identify musical patterns*. It is due month 12, and linked to Task 2 of WP3: *pattern recognition and definition in monodic and polyphonic music*. The task leader is NUIG and the participants are: UNIBO, OU, KCL, CNRS, CNAM, KNAW. This deliverable and task are strongly linked with D3.1 *software tools for pattern extraction* also due in month 12. Central sub-tasks include identifying frequent and significant patterns in music segments, and formally expressing these patterns and relations with other concepts including other patterns. Later deliverables in WP3 are planned at Months 18 and 24, D3.3 and D3.4.

Thus this document D3.2 and D3.1 both report on work in progress. It is organised into an introduction and literature review, followed by four chapters corresponding to four main strands of parallel research as listed below, and a brief conclusion.

Irish folk music An n-gram approach is used to characterise patterns in monophonic melodies, and frequent common patterns are identified. The approach has been informally validated: it identifies both well-known and new links between tunes in a corpus, and in discussion with (independent) domain experts, these links are confirmed to be meaningful. Simple machine learning algorithms for root-note detection in the same corpus have been developed. The code developed in this strand has already been generalised to work on Dutch corpora, enabling next steps in cross-tradition analysis.

Dutch folk music This research will build on extensive previous research with the Dutch Song Database and Meertens Tune Collections. Again it will be mostly monophonic and will find links within and between pieces and across corpora. This strand of research has begun recently (September), and so no new results are to be reported yet.

Search engine framework for patterning mining A faceted search engine for music data is developed, based on representing music using n-grams and feature descriptors, and applying standard search engine modules (indexing, searching, and ranking) and on-line identification of fragments which match the user query. Search across melody, harmony, and lyrics is possible. It is available on the Neuma platform and suitable for general use.

Harmonic similarity By reducing a segment of music to a representation of its harmony alone, common harmonic patterns can be identified. This method currently runs on several datasets including Isophonics for pop music, the JAAH jazz harmony dataset, and Schubert Winterreise for classical music, and succeeds in identifying interesting relationships within and between corpora. It was demonstrated at the SONAR festival 2021.

Document History

Version	Release date	Summary of changes	Author(s) - Institution
V0.1	27/09/2021	Outline released	James McDermott (NUIG)
V0.2	09/11/2021	Partial draft of intro and lit review, and lengthy Ch 3.	James McDermott (NUIG) and Danny Diamond (NUIG)
V0.3	15/11/2021	Initial complete draft	James McDermott (NUIG) and Danny Diamond (NUIG) and Abdul Shahid Khattak (NUIG) and Peter van Krannenburg (KNAW) and Jacopo de Berardinis (KCL) and Andrea Poltronieri (UNIBO) and Albert Meroño Peñuela (KCL) and Raphaël Fournier-S'niehotta (CNAM) and Tiange Zhu (CNAM)
V0.4	09/12/2021	Response to reviews. Added statements on FAIRness and reproducibility (end of Ch 3, short paragraph in Ch 4, subsection in Ch 5, subsection end of Ch 6). Added summary table of pattern types (end of Ch 2) and pattern types we do not use. Added motivation from pilots (Ch 2). Added short section on plans for evaluation (end of Ch 3). Added some introduction to specialist topics eg ABC notation, machine learning, search engine concepts, in the relevant sections. Added more references to other WPs and deliverables. Removed some FACETS-specific material from Ch 5.	As above.
V0.5	17/12/2021	Added details of further deliverables D3.3, D3.4. Added more on future work.	As above.
V1.0	20/12/2021	Updated to 1.0. Removed final page. Final version submitted to EU	UNIBO

Table of contents

1 Introduction

Pattern is a central concept in many fields. Mathematics has been called "the science of patterns" [1]; "pattern recognition" is almost a synonym for the field of machine learning; in architecture, Alexander [2] codified common design patterns, a term then adopted by software developers also. It is central in many forms of art, design, and music [3]. Probably this is because the human brain is characterised by "superior pattern-processing" ability [4], and these activities play with our pattern-recognition abilities.

However, the term has a certain ineffability. For one thing, pattern can be highly concrete ("in bar 3, the pattern in bar 1 is used again") or highly abstract (patterns among patterns). Pattern can be present within an object, or as a relationship between two or more objects. The term can be used both as a count noun ("there are three main patterns in this song") and a mass noun ("pattern is present"). A danger is that a discussion of pattern may involve people talking about different things without realising it.

Focussing on music, one reasonable definition is that pattern is present when there is *repetition*:

> A number of influential music analysts and music psychologists have stressed that discovering the important repetitions in a passage of music is an essential step towards achieving a rich understanding of it. For example, Heinrich Schenker claimed that repetition "is the basis of music as an art" [3, p. 5], Ian Bent proposed that "the central act" in all forms of music analysis is "the test for identity" [4, p. 5] and Lerdahl and Jackendoff [2, p. 52] state that the importance of parallelism [i.e., repetition] in musical structure cannot be overestimated. The more parallelism one can detect, the more internally coherent an analysis becomes, and the less independent information must be processed and retained in hearing or remembering a piece. – [5, original citations included].

This repetition need not be literal or precise, for example in music if a sequence of notes is repeated with transposition, it will certainly be regarded as a pattern. Other related terms include: *invariance* (i.e., something stays the same, though implicitly other things change), *regularity* (i.e., based on rules, which allows for pattern to be present even when highly indirect), or *organisation* (Varèse called music "organised sound" [6]).

When repetition is present but in an abstract way, a complex mapping between the occurrences may be needed, and this is a type of *analogy* [7]. Alexander [2] was motivated by seeing a repetition of approaches across many projects.

Another view of pattern does not necessarily involve repetition. In a search scenario, a user might specify a segment of music, such as a phrase of two bars, to be searched for (the "search pattern") in a large corpus. The search engine will return a result even if the segment occurs only once. However, Bent's "central act" of testing for identity or invariance remains in this scenario.

Within Polifonia, "the overall goal of WP3 is to devise approaches to analysing large repositories of music (tunes, songs, etc.) to identify common, meaningful patterns that are indicative of their identity, filiation or cultural association (genres, origin, etc)." In particular, WP3 requires us to:

- Choose representations for musical data;
- Define algorithms for pattern extraction;
- Define metrics of pattern quality, i.e. measure success;
- Extract patterns in various pieces and corpora;
- Use the patterns and pattern definitions to formally express possible relationships between pieces and corpora.

All this work may be informed by background knowledge (present also in other WPs) with a longer-term goal of creating machine-readable data in the form of ontologies and knowledge graphs (see Deliverable D2.1).

In this report we report on progress and results in WP3 during the first 11 months of the Polifonia project.

In Chapter 2, we start by briefly exploring various *dimensions* of pattern, in order to situate both previous work and the individual strands of Polifonia WP3 work to be reported in later chapters.

2 Common Background and Literature Review

2.1 Dimensions of Pattern

Pattern can be present in any of the "elements" or "layers" of music, including melody, harmony, rhythm, and timbre. Pattern can be on a short time-scale or a longer one. These are two examples of what we might call the *dimensions* of pattern. What are the other dimensions?

Layers Pattern can be present in melody, in harmony, in rhythm, and in timbre. Pattern can also be present in spatial location of audio, and in words where present (see Deliverable 4.1 for Polifonia research on music textual corpora). Importantly, pattern can occur in interactions between layers also.

Time-scale Some patterns might consist of just a few notes, while others are concerned with the large-scale structure or form of a piece (sonata, ABA form, etc.).

Multiplicity If a pattern is defined by a user as a search pattern, then it may occur zero or more times in a given piece. If a pattern is defined by its occurrence then at least one repetition is needed.

Intra-voice versus inter-voice In a monophonic piece of music, repetition may occur immediately, or delayed. In a polyphonic piece of music, repetition may also occur simultaneously, between multiple voices.

Intra-opus versus inter-opus versus inter-corpus Repetition may occur within a single piece, or between pieces. With *inter-opus* pattern, we may be interested in something that occurs once in Opus 1 and occurs once in Opus 2 (for example a particular cadence), or we may only be interested in something that occurs many times in Opus 1 (and thus is already a pattern in its own right) and many times in Opus 2. Inter-opus pattern may be between a pair of pieces, or across a corpus, or even between multiple corpora (now the focus may for example be on differing frequency of usage of particular patterns between the corpora).

Abstraction Some patterns are highly concrete, e.g. a sequence of notes that is repeated multiple times verbatim. Others are more abstract, e.g. in rock music it is common for the drummer to play a fill at the end of each 4- or 8-bar section; even if the content of the fill differs each time, it is an abstract pattern. While concrete patterns are the more common subject of musicology and music information retrieval (MIR), more abstract types of pattern are also of interest and may require more sophisticated algorithms for detection and abstraction.

Similarly, if pattern is (non-literal) repetition, we can see the *degree* of non-literalness as corresponding to abstraction. The following situations all indicate pattern at different (approximately increasing) levels of abstraction:

1. Bar 2 is identical to bar 1;
2. Bar 2 is identical to bar 1, if we ignore dynamics;
3. Bar 2 is identical to bar 1, if we subtract a chromatic or diatonic pitch offset (the latter being a typical transposition);
4. Segment 2 is identical to segment 1, if we ignore the choice of instrument;
5. Segment 2 is identical to segment 1, if we ignore rhythm (e.g. in canon by augmentation as used by J. S. Bach);
6. Segment 2 is identical to segment 1, if we ignore everything *except* rhythm;
7. Segment 2 is identical to segment 1, if we ignore the ending of each;
8. Segment 2 is identical to segment 1, if we ignore everything except chord progression (e.g. parts of *My Way* and *Life on Mars*[1]);
9. Piece 2 is identical to piece 1, if we ignore all content and consider only the "verse-chorus-verse-chorus-bridge-chorus" structure;
10. Piece 2 is similar to piece 1, if we consider only histograms of pitches and note durations [8].

Thus, we see that identifying abstract pattern is synonymous with *ignoring* some information which is (for the moment) irrelevant. Knowing which information is relevant, and which is irrelevant, in a given situation and for the purpose of a given analogy, has been argued to be central to fluid, human-like intelligence [7].

Mechanism of repetition Closely related to the question of abstraction is the *mechanism of repetition*. We will explain by example. Suppose we have two musical segments X and Y, and we perceive that Y is a repetition in some sense of X, i.e. there is pattern. This repetition may arise, or may be perceived to arise, in several different ways. Y may be perceived as a transformation of X. Both X and Y may be perceived as identical after reduction, or as identical if allowing for a few small edits. Both X and Y may be perceived as generated by instances of the same process (this process may be parameterised or non-deterministic, allowing for distinct outputs). Or X and Y may be perceived as generated by *different* processes from the same abstract underlying data.

Exact versus fuzzy Sometimes a pattern is repeated exactly, and in other cases the repetition is varied. This is (slightly) distinct from the reduction/abstraction or "ignoring information" point of view advanced above, as here the focus is on a literal repetition of the concrete music with a small fuzz-factor or *edit distance*, whereas above it is about an exact match (zero edit distance) on some abstracted representation of the music.

Signal versus symbolic data The raw data on which a pattern is defined (i.e., on which a pattern detection algorithm runs) may be an audio signal, or it may a symbolic music representation such as MIDI. Because audio signals are so low-level, it is generally necessary to process them before any meaningful patterns can be detected, e.g. with averaging, spectral transformation, or pitch detection (thus transforming to a symbolic representation).

[1] https://www.youtube.com/watch?v=dd-b8GbOPKg

Musically meaningful Pattern may be present without being meaningful. In a corpus of rock music, we will find many bars with 4 beats. Algorithms free of background knowledge might discover this pattern, but users will know they are not worth reporting. Ideas such as TF-IDF (term frequency – inverse document frequency [9]) quantify our intuition that important patterns are (1) common in the piece, but also (2) distinctive to the piece, not ubiquitous across all pieces. As Conklin [10] writes, "[a] pattern is distinctive if it is over-represented with respect to an anticorpus".

Since pushing boundaries is a fundamental ingredient of many art forms, there are no types of patterns (e.g. highly abstract patterns) which are inherently "off-limits": any type of pattern an artist chooses to use is potentially meaningful. Since listeners may be highly sophisticated and may themselves be artists, there is no asymmetry between artists and listeners in this: any pattern that is perceived by listeners is, again, meaningful.

However, it is also possible that pattern may be present without being intended by the artist or perceived by the listener. When patterns are very common in a tradition or a genre, a composer may use them without conscious intention. Meredith [5] gives an example to show that repetition may be present by coincidence, and thus not meaningful.

Implicit versus explicit A further scenario where this "unperceived pattern" could arise is when we define pattern using the tools of algorithmic information theory (Kolmogorov complexity, minimum description length), as some scientists do, declaring that pattern is present when a compression of the musical signal is possible (taking advantage of previous content of the signal). For example, Dubnov [11] uses neural network and other methods to measure the information rate of the audio signal. Such methods result in the detection that pattern exists, without exhibiting any specific pattern as a result.

Definition versus algorithm In principle, listeners may agree that a pattern fitting a certain definition is present, even if no algorithm is known for extracting such a pattern.

2.2 Goals

The above dimensions relate to the definition of pattern. A different aspect is the *goal* of defining patterns. Sometimes the goal is to identify common origins of pieces. In the work of Lartillot [12] the goal is motivic and thematic analysis. But in the work of Dixon et al. [13], for example, the patterns are to be used in a machine learning genre classification model.

In Polifonia, the goals and motivation for our pattern research arise mostly through the *pilots*:

TUNES: analysis and classification This pilot primarily concerns research with the Dutch folk song collections (to be described in more detail in Chapter 4). The central research question is "To what extent are melodies from Dutch 17th and 18th century sources connected with melodic repertoires from elsewhere in Europe?" These connections may take the form of familial relationships between pieces of music, and pattern analysis is a natural way of uncovering this. Thus, this motivates the cross-corpus research to be described in Chapters 3-4.

FACETS: Exploration of music scores collections through statistical features This pilot will build a search engine suitable for exploration of music scores. One central way to explore such scores is through user queries for the occurrence of music segments. Now, pattern-matching in musical scores is required, motivating the research described in Chapter 5.

INTERLINK Interlinking of collections in digital music libraries and audiovisual archives In this pilot the goals are to achieve a commonality and interoperability between many archives, corpora, and datasets. The research reported in Chapter 6, also related to the SONAR demonstration, begins to establish links across diverse corpora so that they are interoperable. There is a strong link here with Deliverable D2.1, "Ontology-based knowledge graphs for music objects". To go beyond this and find *interesting* links, it demonstrates that harmonic patterns can be used.

In Polifonia, goals are also partly defined through *user stories* and *personas*. The following examples illustrate some of the goals of WP3 pattern research. Descriptions are from the Polifonia *stories* repository[2].

Keith, event and music producer "Keith wants to compile programmes of music, e.g. for music festivals. This involves putting together related pieces of music; in some cases the relationship may be non-obvious" (see Chapter 6).

Mark, computational musicologist "Mark is interested in understanding how Dutch folk tunes relate to other music [...] Can we compare music from different collections, e.g. from different countries to show connections/influences between musical styles? [...] Can we identify a tune (e.g. from an oral tradition) in our collection with music in another documented collection, e.g. RISM, NEUMA, ABC. [...] Can we visualize interconnections, e.g. of tunes which share melodic patterns or geographical origin?" (See Chapters 3-4 for work preliminary to this)

William, art historian working as a curator for the Europeana platform "William is looking for connections between composers, compositions and performers across various collections of 20th century music in Europe, particularly folk music." (See Chapters 3-4 for work preliminary to this)

Sethus, music theorist, composer and teacher, specialized in late Renaissance music. "Sethus uses a search tool that allows to explore the corpus at hand based on several criteria. An initial search can be done based on melodic profile (say, a typical cadence in a soprano line)." (See Chapter 5)

Sonia, lecturer, music producer and festival director interested in finding new music "As each piece plays, the playlist app visualises interconnections to past and future steps in the musical pathway. Shuffle mode can make connections according to a wide range of features such as the composer, musicians, lyrics, melodic patterns, locations and historical events."

[2] https://github.com/polifonia-project/stories

Sophia, musicologist and practising musician "Sophia is analysing a Frescobaldi composition and notices a particular motif that accompanies a reference to birdsong. She decides to see where else this motif can be found in the compositions of Frescobaldi, the compositions of his contemporaries and also investigate the language accompanying the motif." (See Chapter 5)

2.3 Previous work

In this section we briefly mention some highlights of previous work on pattern in music in general. Previous work which is specific to particular subsequent chapters is postponed to those chapters.

As already mentioned, the idea of *ignoring irrelevant information* is central to uncovering pattern. With this perspective, the concept of *reduction* becomes important, as in Schenkerian analysis and the Generative Theory of Tonal Music [14, 15]. Simonetta et al. [16] apply a gradual reduction to melody data and create a graph representation suitable for measuring melodic similarity. Hand-in-hand with the idea of reduction goes the idea of a *trade-off*: any definition which allows us to identify some concrete patterns by reducing some information necessarily fails to identify some other types of pattern.

In folk music, common patterns among tunes have been noted since the 19th century [17]. "Tune families" were identified on the basis of common patterns with assumed ur-melody origins [17, 18, 19], and these ideas have often informed computational ethnomusicology [19].

Still focussing on the melodic layer, repositories exist of common patterns used by jazz soloists. For example, Coker [20] writes that "we seldom hear an improviser's solo that does not contain melodic fragments or patterns", which could be taken from various sources, and so they provide a repository of patterns to fill a "need for a collection of patterns to be practiced diligently" by jazz improvisers. This is a rather different goal from the goals of musicology, e.g. finding common origins of tunes. Interestingly, they choose to present most patterns in eighth notes, "without rhythmic variation", i.e. ignoring some information in order to achieve commonality. Among many other works, Frieler et al. [21] present applications for exploring patterns in jazz solos, drawing from the Weimar Jazz Database [22]. Distinctions are drawn between related terms *patterns* ("repeated sub-sequences") and *formulas*, *licks*, *stock-phrases*, and *riffs*. For Coker [20] and for Frieler et al. [21], patterns are quite concrete and involve short time-scales, defined at the level of pitch sequences but allowing for transposition and rhythmic variation. But by the nature of the genre, patterns in jazz can reach unbounded levels of abstraction and complexity also.

Meredith [5] proposes a geometric "point-set" algorithm for detecting repetition in symbolic music. It considers note pitch and onset time only. It is geometric in the sense that it considers the score as a grid of two axes (diatonic pitch and time), and while the goal is to detect repetition and allow for variation, the variations that are allowed are only those easily expressed by geometric transformations in this grid: delay (= shift in the time axis), transposition (= shift in the pitch axis), elongation (= scale in the time axis). It uses heuristics named coverage, compactness and compression ratio to try to discover only interesting patterns. Extensions to Meredith's basic algorithms have also been proposed [23, 24].

Conklin [10] describes an approach to extracting "distinctive" patterns in symbolic music. Again a musical grid is assumed. Features such as pitch, diatonic pitch, pitch contour (with values of =, +, and -) and duration are calculated at each note-event. Patterns consist of commonly-occurring sequences of such information, even across layers (e.g. [+ + + dur(4)], "a sequence of three upward jumps in pitch, followed by a note of duration 4 beats" might be a valid pattern). TF-IDF ideas are used to avoid identifying many uninteresting patterns, with thresholds for important parameters in the definition of "interesting". Intra-corpus analysis is facilitated: the paper identifies the single most distinctive pattern in each of 6 corpora. Thus, Conklin's patterns are highly concrete, but are unusual in that they can span layers. Both Meredith and Conklin can work with polyphonic music.

In "high art", a composer will generally aim for originality, and if they choose to develop variations on another composer's theme they make this explicit in a title or subtitle, and thus it is available in metadata. In the study of corpora from "oral" traditions it is common to encounter monophonic data, i.e. just melodies, and a high degree of sharing of patterns between pieces which may be useful to identify known or unknown relationships between pieces.

Dixon et al. [13] use rhythmic patterns extracted from audio signals to classify music by genre. Thus, this is a departure from other work mentioned so far in that it is in the rhythm layer and begins with the audio signal rather than the symbolic representation of the music. It is also a departure in that the patterns identified are used in a classification task.

Finally, while our focus is on patterns in music, it is interesting to see both practical and theoretical work on patterns in other fields. Gangemi and Presutti [25] define "knowledge patterns" as something like template sub-graphs in knowledge graphs, whose components can vary while the template structure remains. Finding templates which occur often allows interesting applications. The problems that arise in finding them parallel those of musical patterns: the data (a knowledge graph, or a piece of music) is not "rectangular" but has relational and hierarchical aspects; there are an unlimited number of candidate patterns; and there are no clear boundaries.

2.4 Summary

The central goals of Polifonia WP3 include identification of patterns that are *useful in detecting relationships* between pieces of music, with particular focus on Europe's musical heritage. Taken together with our "dimensions" of pattern this allows us to narrow our focus in the research of the following Chapters 3-6.

We begin with commonalities: all strands of the research work at the symbolic level, and use algorithms to exhibit or use explicit patterns rather than merely detecting that pattern exists.

Next, we show some distinctions between the strands of research, with a summary of pattern dimensions in Table 2.1. We then briefly describe the four substantial chapters.

With a focus on monophonic Irish folk tunes, the work of NUIG (Chapter 3) will use sequence mining approaches on melodic data. Patterns will short-term, and in the melodic layer only. They will be quite concrete, but not at the level of raw pitches: instead melodies will be normalised by transposition, and both Parsons code (using up/down jumps but discarding the size of jumps) and accented notes only (discarding non-accented notes) will be explored. A "frequent n-gram" approach is used, thus multiplicity of patterns is high and is central to our judgement of which patterns are important. With a similar focus on monophonic Dutch traditional music, the work of KNAW (Chapter 4) will also use quite concrete and short time-scale patterns.

In both Chapters 3 and 4, patterns are initially identified intra-opus, but are then used inter-opus to identify potential links between pieces.

In the work of CNAM (Chapter 5), we shift to focus on developing a general framework for pattern extraction, relying on a search-engine back-end (and again an n-gram approach, though slightly different). The user query is regarded as a pattern to be matched in the dataset. This is achieved by a careful data representation in which multiple descriptors extract features for later use in a search engine with indexing, searching, and ranking procedures. The representation allows for inexact, i.e. fuzzy, matching. New descriptors could be added, to allow for new searchable patterns. Patterns here are inter-opus from the system's point of view, in the sense that a user's search is immediately run across the entire dataset.

Finally, in the work of KCL and UNIBO (Chapter 6), we shift to focus on a harmonic similarity model. Thus we are now in the harmony layer. Exciting results emerge from inter-opus and inter-corpus use of patterns. The harmonic similarity measure defined here is positive even for a single instance of a shared pattern between two pieces, thus we are now considering multiplicity of one.

Thus, we can also mention the types of patterns we have not considered in this WP. We are not considering timbre or lyrical layers, or rhythm except indirectly. We have worked with symbolic representations of music only, never audio-level. We have not considered highly abstract patterns, though some exploration here is likely in future.

Table 2.1: Summary of pattern dimensions used in four strands of research in this deliverable. The 'x' characters indicate position in the spectrum indicated.

Irish traditional music, Ch 3

Layer: melodic
Source of pattern importance: multiplicity

Monophonic	x			Polyphonic
Short time-scale	x	x		Long time-scale
Exact	x	x		Fuzzy
Concrete		x		Abstract
Intra-opus	x		x	Inter-opus

Dutch folk music, Ch 4

Layer: melodic
Source of pattern importance: multiplicity

Monophonic	x			Polyphonic
Short time-scale	x	x	x	Long time-scale
Exact	x	x		Fuzzy
Concrete		x		Abstract
Intra-opus	x		x	Inter-opus

NEUMA search engine, Ch 5

Layer: melodic
Source of pattern importance: user query

Monophonic	x			Polyphonic
Short time-scale	x			Long time-scale
Exact		x		Fuzzy
Concrete		x		Abstract
Intra-opus			x	Inter-opus

Harmonic similarity, Ch 6

Layer: harmonic
Source of pattern importance: cross-corpus occurrence

Monophonic			x	Polyphonic
Short time-scale	x	x		Long time-scale
Exact		x		Fuzzy
Concrete		x		Abstract
Intra-opus		x	x	Inter-opus

3 Patterns in Irish Traditional music

In this chapter we describe research on patterns in Irish traditional dance music. This topic is partly motivated by the TUNES pilot (understanding the relationships among a corpus of pieces, with focus on the Dutch tradition but secondary focus on Irish). It clearly fits under the heading of “a digital ecosystem for European Musical Heritage”. The methods developed here will be applicable to other traditions, such as the Dutch tradition to be studied in the TUNES pilot, and vice versa.

3.1 Literature Review

Readings are ongoing in musicology and ethnomusicology, specifically relating to Irish traditional music. Additional ongoing readings seek to establish a general overview of relevant contemporary work in the fields of music information retrieval and computational ethnomusicology and computational musicology more generally [26, 27].

A precedent for codification of patterns within and between melodies in folk music corpora has long existed. In the Irish tradition, the existence of common patterns between tunes was noted in general terms by 19th century collectors George Petrie and William Forde [17]. From the early 20th century onwards, American ethnomusicologist Samual Bayard developed the theory of ‘tune families’: that many traditional melodies could be traced across space and time, in the manner of a family-tree, to ur-melody originators [17, 18, 19]. This theory, which was further developed by Bertrand Bronson’s work on English-language folk song melodies [18, 28], informs not only much subsequent research in the Irish tradition, but in computational ethnomusicology generally [19].

A key foundational resource for our research is the body of work created by Breandán Breathnach. Breathnach was a collector and ethnomusicologist, and a key figure in the establishment of Irish traditional music as an area of formal academic study from the mid-20th century onwards [29]. Although informed by the work of Bayard and Bronson, his work focussed narrowly on the Irish tradition, seeking to derive conclusions directly from the music material rather than imposing an external theoretical perspective [30].

From 1965-1977 Breathnach worked as a State-funded independent music collector and researcher. During this time he wrote on Irish music for both popular and academic audiences, founded and edited the journal *Ceol*, published three volumes of tunes from his collections, and laid the groundwork for the publication of a further two volumes after his death in 1985 [29]. Despite following a relatively narrow policy for tune-inclusion informed by Breathnach’s cultural-nationalist perspective [31], Breathnach collected, identified and indexed a corpus of over 5,000 unique traditional melodies [32], of which 1,208 were ultimately published in print.

He developed an innovative card index system, which numerically indexed every melody in his collection by a sequence of eight integer values, representing the diatonic pitch classes of the first eight rhythmically accented notes [32]. This card index system, held today in the Irish Traditional Music Archive, was important in informing the preliminary phase of our research; and the five published collections from the Breathnach collection form the basis for our working testbed corpus [33, 34, 35, 36, 37].

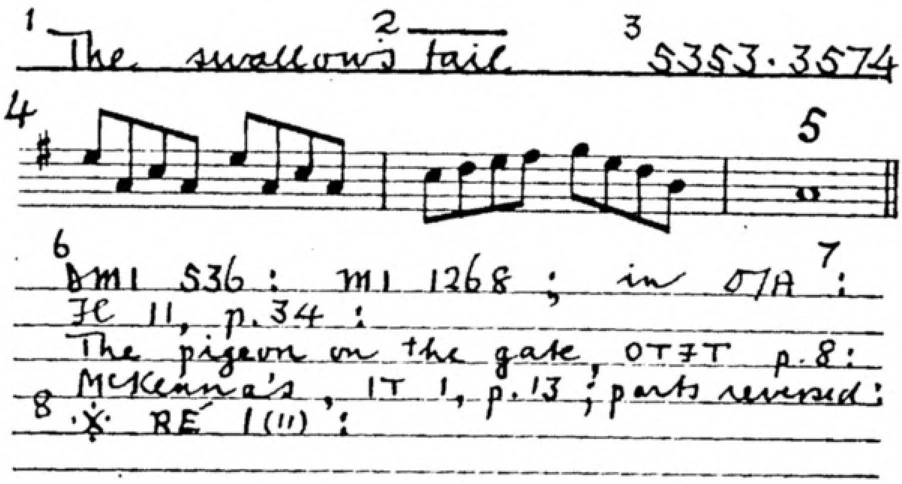

Figure 3.1: An example from Breathnach's card-index system (Breathnach, 1982)

From the 1980s onwards, the 'tune family' theory has been contested as overly simplistic, particularly in its application to Irish traditional music [17, 18, 38]. No consensus has emerged around any single model of Irish traditional tune structure, but many researchers agree to the significance of the pitch and duration of rhythmically accented and cadential notes in defining a superstructure for a given tune, along which interchangeable short melodic and rhythmic blocks are arranged in variable patterns [18]. These blocks or motifs are often common across multiple tunes and are typically between 1/2-bar and 4 bars in duration [18].

Mid- and high-level patterns have also been established: Breathnach noted the importance of the incipit, the first two bars of a given tune, in defining a melodic identity [32]. He also identified the 4-bar phrase as a key melodic unit of Irish instrumental dance tunes, and noted that often these phrases are internally arranged in a call-and-response structure [30]. As early as the 18th century, writers had identified and commented on the common high-level patterns of repetition of 8 or 16 bar 'parts' or sections within a tune, most often AABB, AABA (sonata form), or ABAB [18, 30].

It should be noted that the tonality of music in the Irish tradition is generally modal in character [30, 31]. This is evolving gradually as contemporary compositions integrate Western Classical influences, and influences from other musical traditions, but still largely holds true [31]. Due to this modal, essentially pre-Classical basis, along with the specific structural conventions outlined above, it cannot be assumed that computational musicological analyses developed for Western Classical, popular music, and non-modal Western European folk music will translate to the study of Irish corpora. As such, in the course of reading on computational ethnomusicology, we have consulted pattern analysis work on non-Western folk music, such as [39, 40], alongside the more widely-occurring work on Western classical and Western folk traditions.

Identification and analysis of all but the highest level musicological patterns was largely theoretical and/or anecdotal until the advent of computational musicology from the 1960s onwards [41]. In recent decades the field of computational musicology has expanded from a focus on popular and Western classical music into the analysis of folk forms [42]. The application of computational analysis to folk music particularly has given rise to the subfield of computational ethnomusicology [19, 42], which has been the focus of the readings outlined below.

Contemporary computational ethnomusicology studies encountered in the literature survey generally divide into the following steps:

- Selection of music corpus for study (audio or symbolic representation)
- (Occasional:) Musicological annotation of corpus
- Conversion of musical information to machine readable representation
- Pattern extraction (can be global, local, or more usually a combination of both)
- Pattern analysis (usually multiple methods, the results of which are compared and aggregated)
- (Occasional:) Evaluation of results by domain experts

The work of the Meertens Instituut in The Netherlands, a partner organisation (KNAW) in the Polifonia project, has been a key reference point. Meertens has a tradition of computational musicology research stretching back more than 40 years [19, 43]. Research by Dr Peter van Kranenberg and Dr Berit Jaansen into pattern analysis of the Meertens MTC-ANN musicologically annotated corpus and the Dutch Song Corpus proved very informative: the numeric representations of melodic sequences used in their studies directly informed our decisions re music representation, which are described below.

Another important reference point was the Tunepal music information retrieval app, developed and maintained by a team led by Dr Bryan Duggan in TU Dublin [44, 45, 46]. The app allows users to record via a mobile device microphone, converts the audio to ABC notation via FFT, normalises and simplifies the transcription into a text string. The string is searched across Tunepal's online ABC database of 10,000+ Irish tunes, using an edit distance algorithm [47] to return ranked matches. The search results provide the tune name, MIDI playback of the melody, and transcriptions in ABC and Western music notation.

A third foundational computational ethnomusicology reference was [48], which includes high-level computational analysis of a subset of the Breathnach corpus. This work updates Breathnach's paper-based research into the computational era; along with Tunepal it is one of the few studies tailored specifically to the modal tonality and non-Western structures within the Irish instrumental dance music tradition. Ó Maidín's numeric music representations and his calculations of global key/mode through pitch class histograms were particularly useful in informing our work.

Additional reading involved a broad survey of approaches to music representation and pattern-mining in contemporary computational ethnomusicology studies. It was found that studies generally originate with either an audio corpus or a symbolic music corpus. If using a symbolic corpus, the most popular format is MIDI [49]. Other formats encountered in the literature review include MusicXML [50, 51], ABC music notation [52] and **kern [19].

This input corpus is most often converted to numeric feature sequence representation for pattern extraction and analysis (e.g.: [19, 43, 46]. A minority of studies, such as [44, 52] convert to a text string for pattern analysis. If the input corpus is made up of audio recordings, an extra conversion step is necessary, converting to symbolic representation. Per [44, 45], this is usually achieved via Fast Fourier Transform.

After conversion to feature sequence representation, secondary features can be derived from the primary features of note onset, duration, and pitch [41, 53]. Widely used secondary features include: pitch class; pitch interval class; key-invariant pitch; and key-invariant interval. Global features can also be calculated from both primary and secondary features for a given sequence [41, 53]. These include: total duration; total number of note events; pitch class histograms; melodic contour; time signature. Studies generally make use of both local and global data, though there is consensus in the literature that local values are more effective in pattern extraction and analysis [19, 53].

Once a corpus has been converted to feature sequence representation, patterns can be extracted. In some studies, including those where the input corpus has been musicologically annotated, the corpus is searched for pre-identified candidate patterns and the closeness of resulting matches are ranked, most often via edit distance or compression algorithms (e.g.: [19, 43]. In other studies the candidate patterns must themselves be identified and extracted algorithmically from numeric feature sequences. Typically this is achieved using pattern-mining algorithms (e.g.: [54] or n-grams (e.g.: [55, 56]).

Some studies calculate a single numeric weighted sequence, aggregating multiple features, then extract patterns from this sequence. This approach is termed early fusion in the literature [49]. More commonly, patterns are separately extracted from multiple feature sequences, then analysed and compared. This approach is termed late fusion [49]. For both early and late fusion, pitch and duration sequences are the most commonly-used inputs, but we also encountered studies which derived patterns from melodic contour and interval sequences [41, 57].

Once patterns have been extracted from feature sequences, they typically are algorithmically compared against each other and against the corpus, to extract and rank similar patterns. In the literature this measurement of similarity is most often conducted via alignment algorithms (e.g.: [19, 41, 44, 45]; occasionally using compression algorithms [58], n-gram probability models (Hilleware et al, 2014) or by geometric analysis [59]. Some studies also engage musicologists and/or musicians to subjectively judge the validity of identified patterns.

3.2 Methodology

The 1,208 melodies collected in the five published volumes of Breandán Breathnach's *Ceol Rince na hÉireann* were selected as a viable test corpus for our work. This material is monophonic and is entirely comprised on traditional Irish instrumental dance music. A transcription of the entire collection into ABC notation was created by American musician and researcher Bill Black, and this version of the collection is freely available from his website for re-use [60]. ABC notation is a simple text format particularly used for monophonic folk music which includes a symbolic representation of notes as letter names, and some optional metadata. This version of the collection, which has previously been used in [44, 45, 46, 52], is the basis for our corpus.

Figure 3.2: Clean ABC corpus example: Lord McDonald's reel

As a first step, the ABC corpus was manually cleaned. This work entailed: removal of non-valid ABC characters; removal of standardisation of repeat markers; and removal of alternate tune versions introduced during transcription into ABC notation. Root note (as chromatic pitch class) was also manually assigned for every tune in the corpus– for use in calculation of key-invariant secondary feature sequences. After this pre-processing work, the clean corpus was converted from ABC notation to MIDI using EasyABC 1.3.7 [61]. The resulting dataset is available for re-use (see Section 3.5).

The clean MIDI corpus was then converted to primary numeric feature sequences representing pitch, onset, duration, and velocity. This conversion used the Music21 Python library [62] and additional code, structured per Figure 3.3.

Additional secondary feature sequences were derived from these primary features, including: interval, key-invariant pitch, pitch class, pitch class interval, inter-onset interval, bar number, and

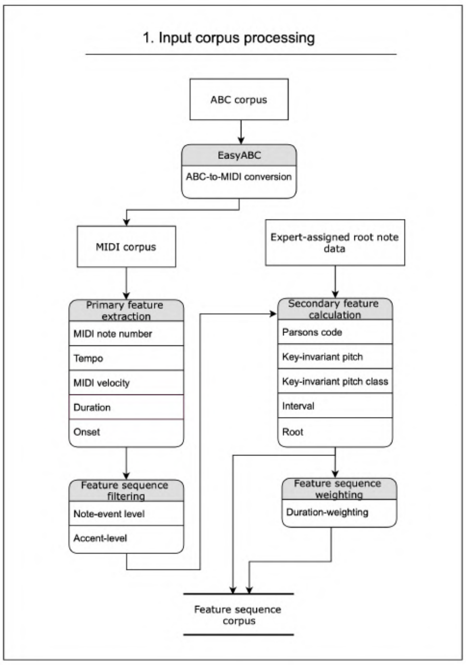

Figure 3.3: Flow chart 1: Corpus setup

simple melodic contour represented in Parsons code [63] The Parsons code values are formatted per: 1 = upward pitch movement relative to previous note; -1 = downward movement relative to previous note; 0 = repetition of previous pitch. All feature sequence calculations were carried out in Python, with the output for each tune stored in Pandas [64] dataframes.

Lord McDonald's (reel)

	note_number	start_time	duration	velocity	tempo	parsons_num	parsons_contour	interval	pitch	pitch_class	pitch_class_interval	inter_onset_interval	duration_eighth_notes	bar_count	cumulative_duration
0	67	0.002	0.498	105	180.0	0	0	0	-12	0	0	0.5	1.0	0	1.0
1	62	0.502	0.498	105	180.0	-1	-1	-5	-17	7	7	0.5	1.0	1	2.0
2	67	1.002	0.498	80	180.0	1	0	5	-12	0	-7	0.5	1.0	1	3.0
3	71	1.502	0.498	80	180.0	1	1	4	-8	4	4	0.5	1.0	1	4.0
4	67	2.002	0.498	80	180.0	-1	0	-4	-12	0	-4	0.5	1.0	1	5.0
5	74	2.502	0.498	95	180.0	1	1	7	-5	7	7	0.5	1.0	1	6.0
6	67	3.002	0.498	80	180.0	-1	0	-7	-12	0	-7	0.5	1.0	1	7.0
7	71	3.502	0.498	80	180.0	1	1	4	-8	4	4	0.5	1.0	1	8.0
8	67	4.002	0.498	80	180.0	-1	0	-4	-12	0	-4	0.5	1.0	1	9.0
9	72	4.502	0.998	105	180.0	1	1	5	-7	5	5	1.0	2.0	2	11.0
10	71	5.502	0.498	80	180.0	-1	0	-1	-8	4	-1	0.5	1.0	2	12.0
11	72	6.002	0.498	80	180.0	1	1	1	-7	5	1	0.5	1.0	2	13.0
12	69	6.502	0.498	95	180.0	-1	0	-3	-10	2	-3	0.5	1.0	2	14.0
13	67	7.002	0.498	80	180.0	-1	-1	-2	-12	0	-2	0.5	1.0	2	15.0
14	64	7.502	0.498	80	180.0	-1	-2	-3	-15	9	9	0.5	1.0	2	16.0
15	67	8.002	0.498	80	180.0	1	-1	3	-12	0	-9	0.5	1.0	2	17.0
16	62	8.502	0.498	105	180.0	-1	-2	-5	-17	7	7	0.5	1.0	3	18.0
17	67	9.002	0.498	80	180.0	1	-1	5	-12	0	-7	0.5	1.0	3	19.0
18	67	9.502	0.998	80	180.0	0	-1	0	-12	0	0	1.0	2.0	3	21.0
19	69	10.502	0.498	95	180.0	1	0	2	-10	2	2	0.5	1.0	3	22.0
20	67	11.002	0.498	80	180.0	-1	-1	-2	-12	0	-2	0.5	1.0	3	23.0

Figure 3.4: Feature sequence data for Lord McDonald's reel (accent-level values highlighted)

Feature sequences were extracted at both note event- and accent-level (i.e.: for every note event in every melody; and for rhythmically-accented notes only). These two parallel approaches follow the consensus in ethnomusicological work on the Irish tradition: that patterns of accented notes are of particular importance in defining a melodic superstructure, which is then filled in using lower level melodic motif patterns [18, 38, 65]. Duration-weighted versions of the pitch and pitch class sequences have also been calculated, but have not yet been studied in our research so far (duration weighting converts a sequence of feature values per note event to a sequence of feature values per temporal unit, in our case per eighth note).

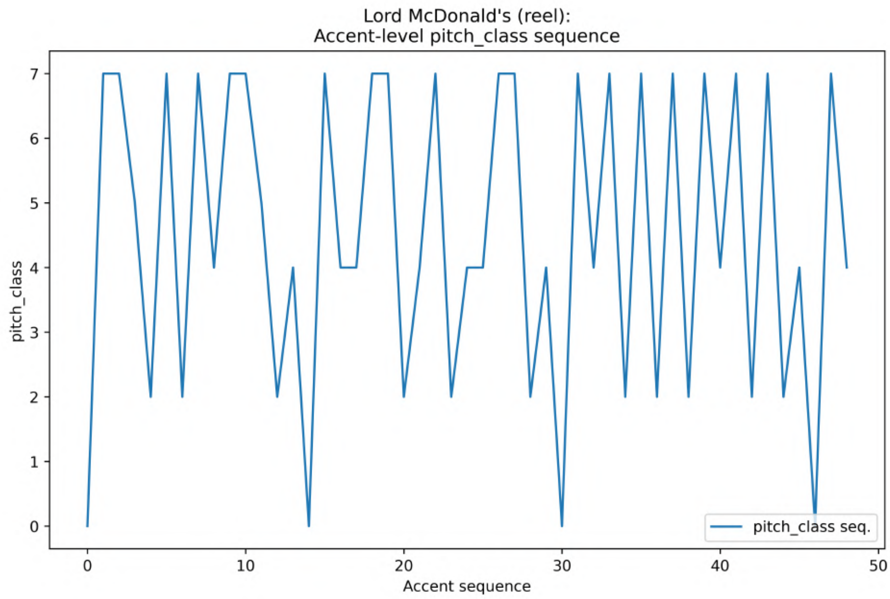

Figure 3.5: Visualisation of accent-level key invariant pitch class sequence for Lord McDonald's reel

We experimented with pattern extraction, firstly using a Python implementation of the PrefixSpan sequence mining algorithm [66], and then using a simple *n*-gram extraction script based on that of Triglia [67]. The focus has so far been on *n*-grams but it is our intention to revisit sequence mining as work progresses. *n*-grams for $3 \leq n \leq 12$ were extracted from every melody in the corpus for five feature sequences: melodic contour, interval, pitch, pitch class interval, and pitch class.

Initially concentrating on accent-level pitch class 6-grams (i.e.: patterns of six accented pitch classes), unique *n*-gram patterns occurring in each melody were counted, ranked by tf–idf [68, 69], and stored in Pandas dataframes. These dataframes then were concatenated to provide a corpus-level table of unique *n*-gram patterns, ranked by idf.

This ranked database of patterns could now be used for pattern similarity searching. Initial testing involved geometric distance measurements (Cosine, Euclidean), compression (zlib, Lempel-Ziv) and edit distance algorithms (Levenshtein, Smith-Wateman, Gotoh, Damerau-Levenshtein). The Damerau-Levenshtein algorithm [70] was selected for further investigation.

Ongoing exploratory work involves extraction of the top pattern(s) in a candidate melody as ranked by tf–idf, per Figure 3.7. These patterns are used as input search candidates. Using the Damerau-Levenshtein edit distance algorithm as implemented in the *fastDamerauLevenshtein* Python library [71], similar patterns to the candidate(s) are identified in the corpus-level table of

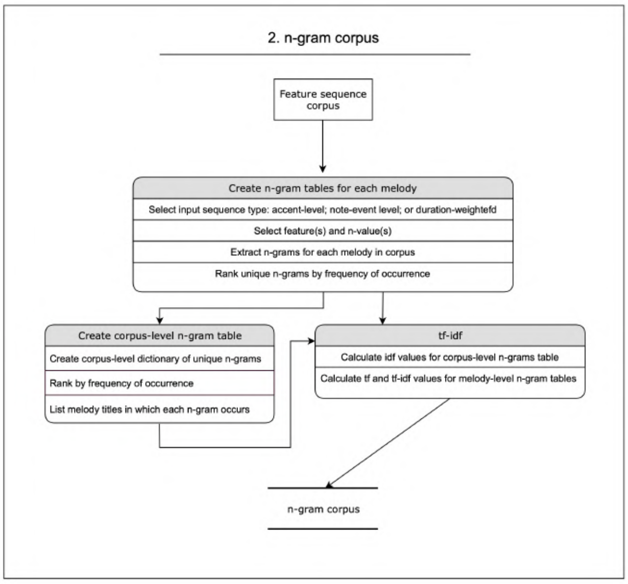

Figure 3.6: Flow chart 2: *n*-gram and tf-idf calculations

patterns. The Damerau-Levenshtein approach allows fuzzy searching via custom edit distance penalties for sequence item rotations, substitutions, and for variations of sequence length. This is important for Irish material, where, due to the oral nature of transmission in the tradition, there can be great variation between transcriptions of the same or similar pieces of music [31].

Once similar patterns to our candidate pattern(s) are identified per the above, melodies across the corpus in which any of the similar patterns occur are identified. The number of similar patterns occurring within each of these melodies is counted: E.g.: If a melody contains three patterns identified as similar by the Damerau-Levenshtein algorithm, its count value is three. The melodies are ranked in descending order according to this count, and these results tables are currently under evaluation as a potential indicator of overall melodic similarity.

The frequency of occurrence of each pattern in each melody is not yet included as a similarity metric, though this is intended as the next addition to the process.

A GitHub repository containing work-in-progress code to run the methodology above in full is available here as a Polifonia deliverable D3.1 component.

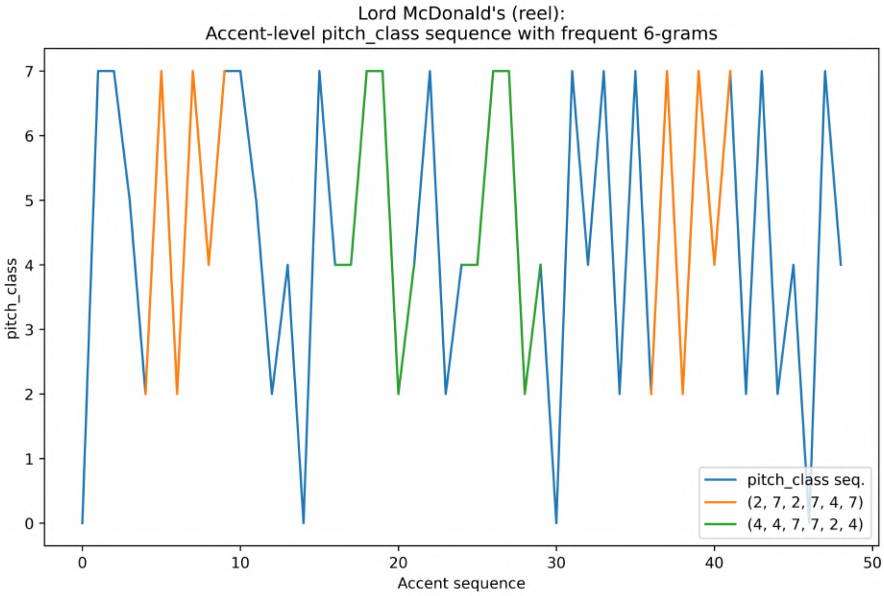

Figure 3.7: Visualisation of frequent pitch class 6-grams for Lord McDonald's reel

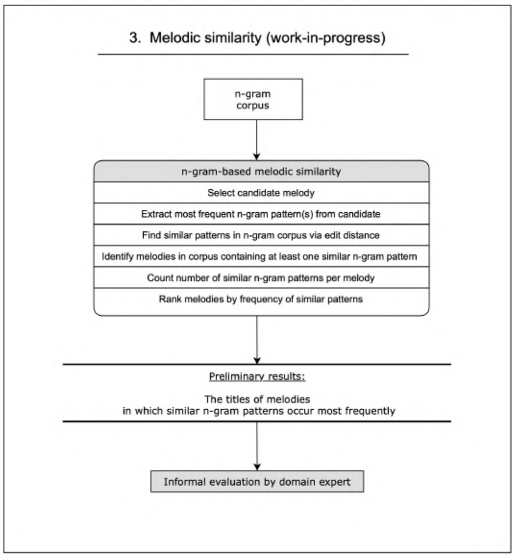

Figure 3.8: Flow chart 3: deriving *n*-gram-based melodic similarity results

3.3 Results

Results for test melodies *McDonald's (reel)*, *Anderson's Reel*, and *Anac Cuan* have returned musically plausible melodic connections, as informally verified by a domain expert.

For example, Figure 3.9 lists melodies containing similar 6-gram patterns to the pitch class sequence (2, 7, 2, 7, 4, 7), which, per Figure 3.7 above, occurs twice in *Lord McDonald's (reel)*.

Frequency of occurrence of similar n-grams for (2, 7, 2, 7, 4, 7)

	Title	Frequency
1	Lord McDonald's (reel)	8
2	Tim Mulloney's (reel)	5
3	Kiss the Maid Behind the Barrel (reel)	2
4	View from Black Head , The	2
5	Maid Of Mount Kisco (reel), The	2
6	Anderson's Reel	2
7	Greig's Pipes (reel)	2
8	Monasterevin Fancy (reel), The	2
9	O'Shaughnessy's Reel	2
10	Repeal of the Union (reel), The	2

Figure 3.9: *n*-gram-based similarity results for Lord McDonald's (reel)

Figure 3.10 shows the melody of the top-ranked tune, *Tim Mulloney's (Reel)*, which contains five similar accent-level pitch class patterns to (2, 7, 2, 7, 4, 7), as measured by Damerau-Levenshtein distance. This may be compared to the melody of *Lord McDonald's (reel)*, as illustrated in Figure 3.2, which exhibits obvious melodic similarity.

The importance of the opening 4 bars in defining the theme of Irish traditional melodies will be investigated further as, from our exploratory work, patterns such as the candidate under investigation above, which occur early in a melody, appear to have the greatest significance in defining similarity. This corresponds with Breathnach's observations on melodic structure in traditional Irish instrumental dance tunes [30]. Conversely, patterns occurring outside of the candidate melody's opening thematic section have also been tested, and in these cases the similarity results returned were not verifiable by domain expert.

It should be noted that further work is required to build upon the initial exploratory findings reported and discussed above. In particular, we plan to gather a small dataset of known links between pieces ("tune families" as already described), and use it for objective/quantitative evaluation of performance.

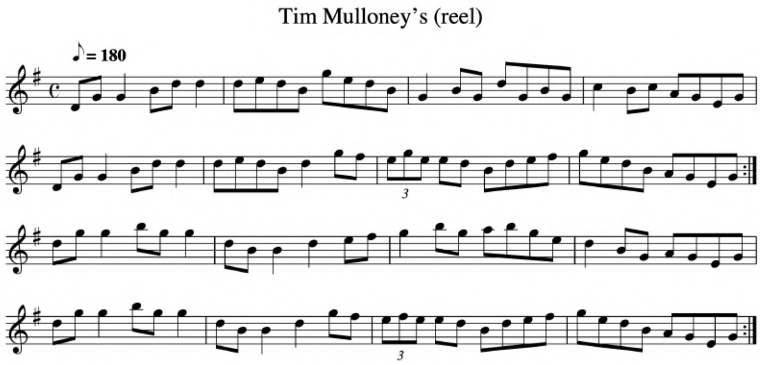

Figure 3.10: Tim Mulloney (reel)

3.4 Root Note detection

Accurate detection of pitch class patterns relies fundamentally on accurate key-invariant pitch sequence data. Initially Breathnach's approach was adopted, generating key-invariant pitch sequences relative to the final note of the melody, which was assumed as the root [32]. This was revisited due to concerns over accuracy, and the root of each melody in the corpus was manually assigned by a domain expert. During this pass, the corpus was also cleaned to remove non-valid ABC characters, inaccurate repeat markers, and additional versions of the melodies which had been appended during transcription to ABC format. This work produced a clean, annotated version of the ABC corpus, which is provided as a component for Polifonia deliverable D3.1.

Working on the clean corpus, Breathnach's final-note-as-root approach was found to agree with the expert-assigned root in 71 percent of cases. Transcriber-encoded root note values were also available from the ABC Notation metadata, and were found to agree with the expert-assigned root in 86 percent of cases. The Music21 Python library contains a built-in implementation of the Krumhansl-Schmuckler pitch class intersection algorithm [72], with various built-in weightings available as listed below. Theses algorithms were individually tested on the corpus, but all had under 80% agreement with the expert-assigned root values. It is thought that the low accuracy of the Music21 algorithms when applied to the Irish corpus were due to the modal nature of a significant proportion of melodies in the Irish tradition, as discussed in [30] and [46]. As the Music21 algorithms function on the basis of tonal major and minor keys only, it is unsurprising that they operate with reduced accuracy on modal music. Due to the relatively low accuracy levels of these existing, easily accessible options for determining root values, a new approach was instigated as detailed below.

Note: The Humdrum Toolkit Keycor key detection tool, as described in [73], runs the same Krumhansl-Schmuckler-based algorithms as Music21 via a command-line interface, and offers simple access to customisation options such as user-defined weightings for non-tonal keys. It is intended to run Keycor key detection on the Irish corpus and to include the results in the above comparison.

The Krumhansl-Schmuckler pitch class intersection algorithm [72], as implemented in the Music21 Python library [62], was used to detect root values for all melodies in the corpus. The algorithm was run five times, each with a different set of weights applied; additional root detection metrics were also calculated as listed below.

This data was saved in a table with each measure as a column, for input into an ensemble method machine learning algorithm. Figure 3.11 shows a graphical depiction of the proposed architecture. Each tune was processed to extract numerous aspects, as seen in the Figure, and this was also reported above.

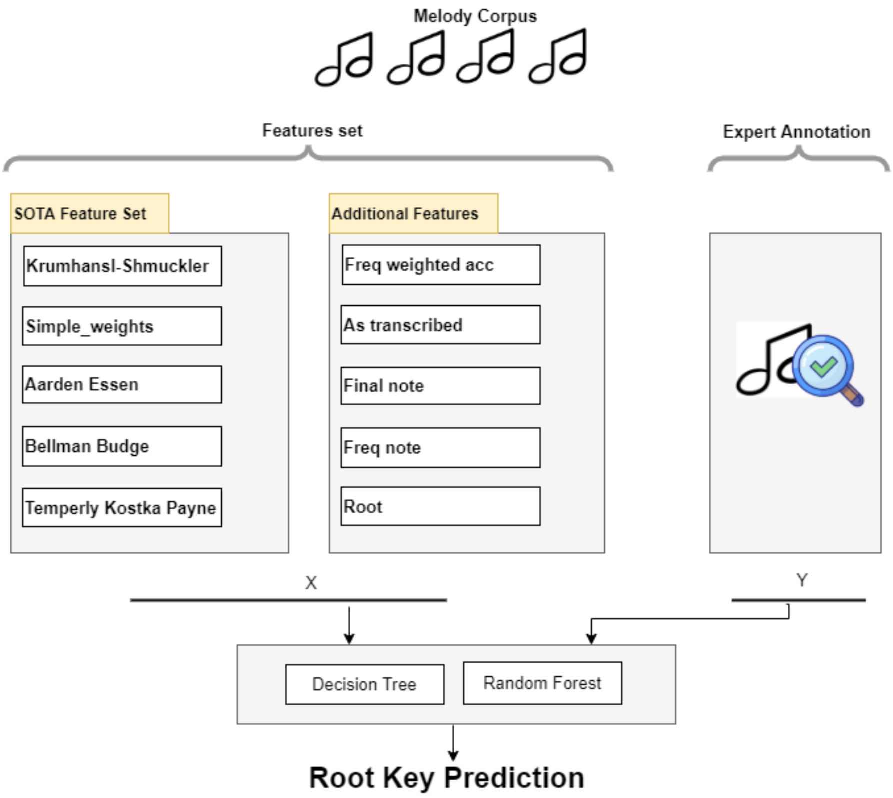

Figure 3.11: Proposed Ensemble Method of Machine Learning for Root Note Detection

The features in figure 3.11 have been divided into two categories: 1) those extracted using state-of-the-art techniques, and 2) those extracted using supplementary methods. The following is a full description of each feature:

State-of-the-art Key finders as features:

- **Krumhansl-Shmuckler** is the result of the Krumhansl-Shmuckler key finding algorithm, with default/original weights [72].
- **simple_weights** is the result of the Krumhansl-Shmuckler key finding algorithm, with Craig Sapp's simple weights from keycor toolkit[73].
- **Aarden Essen** is the result of the Krumhansl-Shmuckler key finding algorithm variation with Aarden-Essen weights [74].
- **Bellman Budge** is the result of the Krumhansl-Shmuckler key finding algorithm variation with Bellman Budge weights [75].
- **Temperly Kostka Payne** is the result of the Krumhansl-Shmuckler key finding algorithm variation with Temperly Kostka Payne weights [76].

Additional Features:

- **As transcribed** is the pitch class of the transcriber-assigned root, extracted from MIDI file. It is worth noting that this is human-supplied, but can be in error relative to our expert annotations, thus we are justified in using it as a feature in predicting the true key
- **Final note** is the pitch class of the final note in the melody.
- **Freq note** is the most frequently occurring value in the note-level pitch class sequence.
- **Freq weighted acc** is the most frequently occurring value in the duration-weighted accent-level pitch class sequence (see section 3.2 above).

It was important to explore the data to get more insights into it. Therefore, Exploratory Data Analysis (EDA) was conducted to understand the underlying structure and relationship among different features used in the dataset. For a machine learning model, it is preferable to have a balanced dataset. Our proposed model will be predicating a key, and therefore, it was important to understand the overall coverage of each key in the whole corpus. Table 3.1 shows the total count of distinct keys assigned to melodies in the corpus.

Table 3.1: Coverage of distinct root notes in the whole corpus

Expert Assigned Root Note	Symbolic Representation	Total Count
2	D	481
7	G	415
9	7	181
4	E	126
0	C	14
11	B	6
5	F	1

A number of methods exist to assist in determining the root note of a melody. However, none of the described methods are capable of accurately determining the root note to a tune. We saw in the previous paragraph that there could be a lot of factors at play when determining the root note of a melody. Important characteristics must be chosen for an ideal set of features for a machine learning model. Furthermore, feature selection is critical because more features do not always imply better outcomes. As a result, choosing fewer features that are more correlated may improve the overall results. As a result, to choose the subset of the most relevant attributes, a filter method was applied. After the features are ready, a model can be developed around them. Filtering was done in this experiment using a correlation matrix, which is most typically employed with Pearson correlation. As illustrated in Figure 3.12, we created a Pearson correlation heatmap. To better understand the relationship between features and independent factors, a thorough study was carried out. Finally, only those features were considered that have a strong correlation with a class label (expert annotation). The correlation values of each variable can be observed in the heatmap given in Figure 3.12. Based on strong correlation, the selected features were "Krumhansl-Shmuckler", "simple weights","Aarden Essen", "Bellman Budge","Temperly Kostka Payne", "as transcribed", "final_note", "freq note".

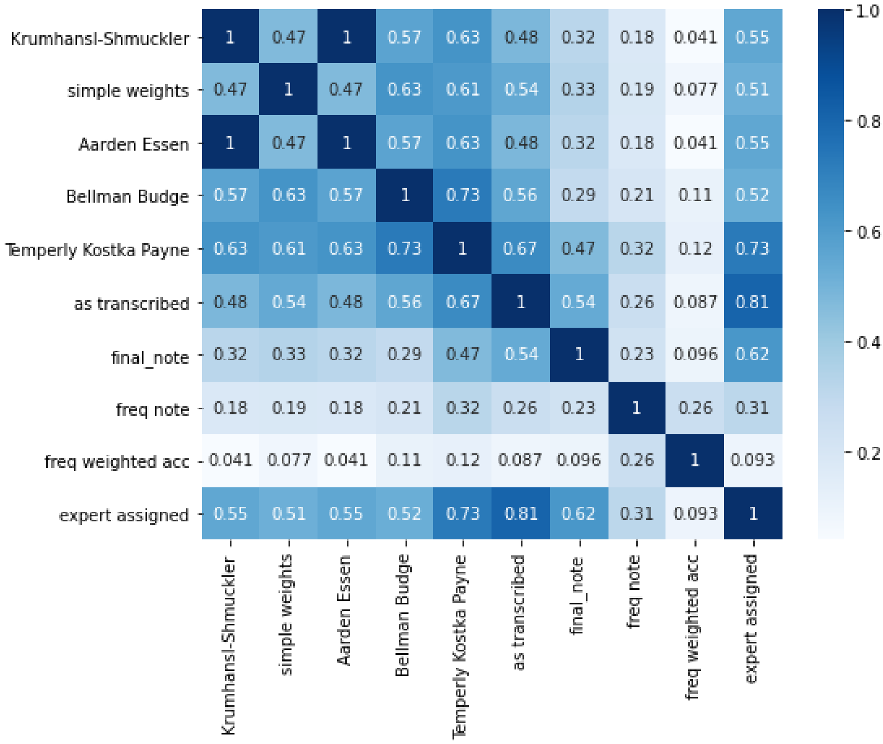

Figure 3.12: A Correlation heatmap of dependent and independent variables

Two fundamental machine learning classification algorithms, Decision Tree and Random Forest classifiers, were used for the trials. There were 1220 records in the dataset, and we used 80 percent (979 records) for training and 20 percent (245 records) for testing. The results of each classifier are presented separately below.

Decision Tree:

Decision Tree (DT) offer different parameters for tuning to enhance its accuracy. For example, one most important parameter is related to finding a splitting parameter. There are a couple of commonly known methods i.e. information gain and Gini index. There is not a large difference between them, however, the Gini-index is less computation-intensive as it does not compute logarithmic values and it also minimizes the misclassification rate. Therefore, in this experiment, we used the Gini index. However, the information gained should be checked to ensure that it has no negative impact on the overall results. The max depth of the tree is the next hard parameter. A high value for the max depth parameter may cause the model to overfit. Similarly, limiting its value to a low value may cause your model to miss trends and patterns in the dataset, i.e. to underfit. However, after running our model several times and observing the best results, we set max depth to 10. With these settings, we ran our model and were able to attain an accuracy of 88 percent for melodic root note classification.

Random Forest:

Random Forest (RF) is another cutting-edge classifier that can be utilized for a variety of classification problems. Random forest performance, like DT, can be improved by tweaking a few parameters. One of the most essential characteristics in RF, for example, is the number of trees used. We started with a number of trees of 1 and increased step by step until we reached the best accuracy at 9 trees. We found that as the number of trees increased over 9, the model's accuracy remained constant. As a result, the total number of trees has been set to nine. The relationship between no of trees and the accuracy of the model is shown in Figure 3.13. It can be observed from the figure that increasing number of trees is not affecting the overall accuracy of the model.

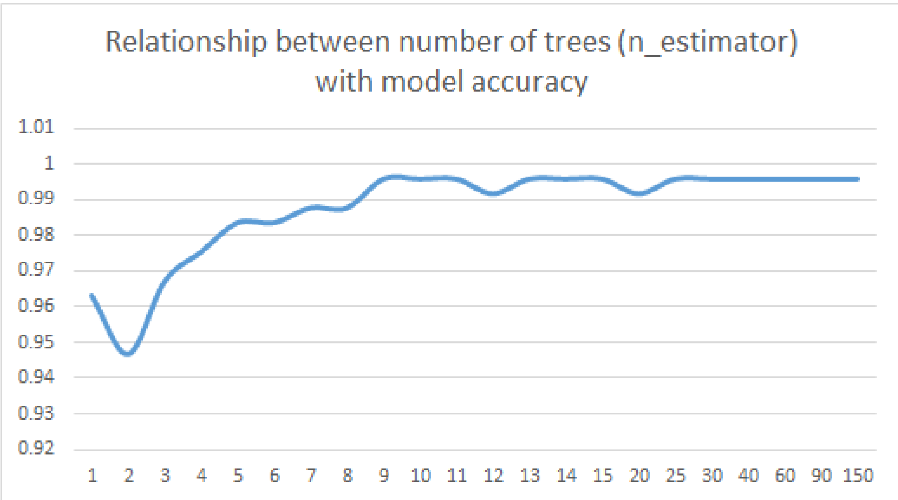

Figure 3.13: The line graph shows the relationship between number of trees (n_estimators) and accuracy of the Random Forest model for Root note detection

Finally, the classification accuracy was measured, and it was found to be 99 percent. The Figure 3.14 shows the classification report for each key. The outcomes are self-evident. On this dataset, our proposed ensemble approach performs remarkably well. However, there are a few concerns that should be addressed before drawing any conclusions: 1) this model should be tested on other datasets, 2) the class distribution is not balanced, so enough datasets of each root note could be generated either manually or synthetically, and 3) parameter tuning should be done in a more rigorous manner.

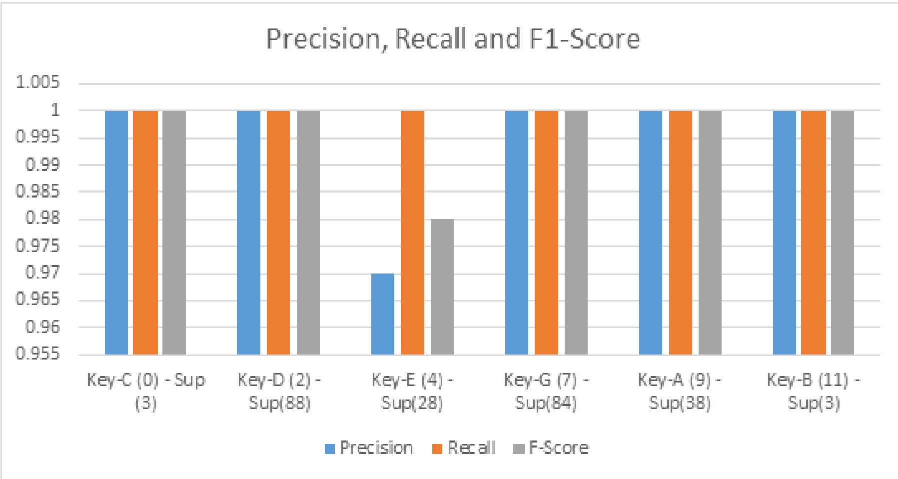

Figure 3.14: Precision, recall, and F1-measure of Root note detection of a Melody using Random Forest classifier

3.5 FAIRness and reproducibility

Our work in this chapter complies with the FAIR principles and is according to the Polifonia First Data Management Plan (D7.1).

As described, the central data used in this research is the *Ceol Rince na hÉireann* corpus. A transcription of the entire collection into ABC notation was created by American musician and researcher Bill Black, and this version of the collection is freely available [60]. It has been widely used in previous work. We have carried out some cleaning and conversion to MIDI (see above). Thus our data is a secondary source as per D7.1 Section 2.1.1. The changes will be notified to the "upstream" maintainer.

The data model here is a mere collection of files in MIDI format. Such a file is effectively an event-list (where each event specifies time, pitch, velocity, and duration of a note) plus some optional meta-data including author, title, and key.

The resulting corpus is available from our GitHub repository `https://github.com/polifonia-project/folk_ngram_analysis` for findability and accessibility. Since the data is in a common format (MIDI), it is interoperable and reusable. The repository has a Zenodo identifier (`https://zenodo.org/record/5768216#.YbNExFnLe01`), which ensures persistent access to the version of the data described here. The dataset is released under a permissive Creative Commons license. Our experiments and results can be reproduced using our freely-available code and data. However, our early evaluation of the system involves informal evaluation by experts, and this aspect is not intended to be reproducible. In future, it is hoped to codify some expert judgements in a way that allows reproducible evaluation of the research.

The data is a *corpus*, not a *dataset* in the terms of D7.1 Section 2.1.2, since it has not been converted to a Knowledge Graph or Linked Data format. However this is envisaged as part of future work.

No personal or sensitive information is involved in this strand of research and so there are no security or privacy concerns.

4 Patterns in Dutch traditional music

4.1 Introduction

In this chapter, we introduce the background for the work that will be done on patterns in Dutch traditional music. Since this work, which is connected to the TUNES pilot, only started per September 2021, there are no results to report, as of yet. We will introduce the data set and provide a brief overview of previous work we will build on.

4.2 Data

4.2.1 Dutch Song Database

The Dutch Song Database is a digital repository documenting Dutch song culture throughout the ages [77]. The main feature of the database is the availability of collected references to occurrences of Dutch songs. At the moment, the database contains metadata on c. 180 thousand occurrences of Dutch songs in a variety of sources dating from the twelfth century up to the present day. The database documents many kinds of (folk) songs, including love songs, satirical songs, beggar songs, psalms, other religious songs, children's songs, St. Nicholas songs and Christmas songs. The main sources in which these songs were found are songbooks, songsheets (broadsides), song manuscripts, and fieldwork recordings. The cataloged record for each song contains information about the source in which the text and/or the melody occurs. In most cases direct links are provided to the complete song text, to a scan of the source, to the notated sheet music, or to a recording of an individual song. The database can be accessed by the general public through an online interface at http://www.liederenbank.nl.

4.2.2 Meertens Tune Collections

A large number of song texts and melodies have been digitized and are currently accessible. Through the search facilities of the web interface of the Dutch Song Database, the melodies can be studied individually. To facilitate corpus studies, the melodic data sets have also been released separately as the Meertens Tune Collections (MTC) [78]. The most important data sets for the current project are MTC-FS-INST-2.0 and MTC-ANN-2.0.1. The former contains full melodic contents and meta data for c. 20.000 melodies. The latter contains a rich set of annotations for a small selection of 360 melodies in 26 tune families. These annotations include occurrences of melodic patterns as perceived by collection specialists of the Meertens Institute.

For both MTC-FS-INST-2.0 and MTC-ANN-2.0.1, sequences of melodic features have been pre-computed and made available in a Python module MTCFeatures.[1]

4.3 FAIRness and reproducibility

The data and code are open, as discussed above. The pre-computed features have been released under a CC licence (Attribution-NonCommercial-ShareAlike). The data are available from Zenodo,[2]. The interoperability of this data set will be improved in future work in the TUNES and INTERLINK pilots. Both data and meta-data adhere to common data standards, notably the json-standard.

4.4 Previous Work

In a study on melodic similarity among folk songs, we examined the role of different musical features for the human categorisation of songs into tune families [79]. We asked the collection specialists of the Meertens Insitute to rate the similarity of pairs of melodies from the same tune family. The specialists rated the melodic similarity for various dimensions: contour, motifs, rhythm, and lyrics. From these ratings it became evident that the motivic similarity is the prevalent kind of similarity for the collection specialists to recognise a melody as member of a tune family. This finding is a strong indication to focus our research on melodic motifs, or melodic patterns. Apparently, these play an important role in establishing the perceived identity of a melody as member of a tune family. We asked the specialists to identify these motifs and annotate their occurrences in the songs. This resulted in the annotations that are included in data set MTC-ANN-2.0.1, consisting of more than thousand motif occurrences in more than hundred motif classes.

We performed a retrieval experiment [80] in which we evaluated the retrieval of tune family members from a large database of more than 4.800 melodies. The matching was based solely on sequential occurrence of melodic patterns. In total, we considered 15 patterns that were manually defined by closely examining the annotated patterns. We reached a recognition rate of 0.75, meaning that for 75% of the queries a member of the same tune family was at the first rank in the result list. Furthermore, we found that for 90% of the queries a member of the same tune family is among the first 10 melodies in the result list. This implies that occurrences of a small set of melodic patterns provide enough information to identify a query melody as member of a tune family.

Mining for Maximally General Distinctive Patterns (MGDP) [81] is another approach to gain better understanding of a melodic corpus in terms of recurring patterns. MGDPs are patterns that are overrepresented in a certain corpus of melodies compared to an anti-corpus. We performed this analysis for the MTC-ANN-2.0.1 data set [82]. This resulted in a set of 22 patterns in 14 tune families. These patterns correspond quite well with human annotated patterns, but the exact

[1] https://pvankranenburg.github.io/MTCFeatures/

[2] doi: 10.5281/zenodo.3551003

interpretation of the relation between the mined MGDPs and melodic motifs in the traditional conceptualisation remains a challenge.

5 A search-engine framework to mine patterns from digitised scores

The work presented in this section is a bit different than the other parts of this document, focusing on the development of a *search* framework capable of extracting patterns from several dimensions of musical content, not a statistical analysis of patterns in a given corpus.

5.1 Introduction

Search engines are essential tools for information retrieval in large digital libraries. They help to retrieve a ranked list of relevant content for a given query and they usually rely on scalable indexing structures and algorithms that allows instant response [83]. Notable successes have been obtained for text-based documents, and extended to multimedia collections [84, 85, 86]. Music libraries currently lacks well-founded information retrieval tools and content-based music retrieval still remains as a challenge.

Musical content is intricate and hard to describe textually in natural terms. Temporal aspects (tempo, metric, synchronisation) complicate the attempts to provide a synthetic representation. Moreover, musical contents are extremely versatile, with a wide range of aspects that yield a boundless number of genres, styles and forms. Last but not least, periods and locations (of composition or interpretation) also increase the variability of the material. When it comes to digital representations, one is confronted to highly diverse encoding paradigms. Music in audio formats are the most common. *Symbolic representations*, on the other hand, are oriented towards content-based description of musical pieces. Music scores are the most elaborated way of describing music at this symbolic level. Digitally-encoded scores are often represented in the format of piano roll [87], MusicXML [88] or MEI [89, 90] (XML-based variants for scores).

The *MusicSearch* framework lay the foundations of a fully-functioning and scalable faceted search engine (FSE) for music score collections, supporting explorations and discovery of scores of interest in large collections, based on features such as melodic, harmonic or rhythmic patterns, style, structure, instrumentation and metadata. Potential users are music researchers, instructors, or anyone that is interested in deep exploration of musical contents.

5.1.1 Related work

Our approach relies on an abstract music content model, from which musical features are extracted and represented in a text-based format to be integrated into a standard search engine, ready for pattern querying. The abstract content model consists of a tree-based decomposition of a music score and the associated grammar. This draws heavily from [91, 92, 93], which introduced into the musical literature some ideas and tools from the fields of databases systems and computer linguistics (*e.g.,* hierarchical decomposition of musical content and context-free grammars). Our work also relies on libraries like Music21 [94] to extract some features.

The idea to benefit from an inverted index and splitting musical sequences into n-grams has been experimented in several earlier proposals [95, 96, 97, 98]. While pattern mining is a long-standing effort in the Music Information Retrieval (MIR) community [96, 99, 100, 101], developing full-featured search engines is an emerging topic, because it's only during the last decade that large collections of digital music have been produced and made widely available. Some preliminary initiatives, such as Peachnote [102], only allowed for exact search. Recently, [103] pushed the state-of-the-art further, presenting a full language dedicated to searching into symbolic scores. However, it is based on regular expressions, which cannot scale to large collections, and the search input requires typesetting XML-based code. Our framework is scalable and features a modern and user-friendly interface.

5.1.2 Global architecture of the *MusicSearch* framework

The *MusicSearch* framework relies on a *Music Content Model*, which borrows from the principles of music notation, reduced to the aspects unrelated to the layout of music scores. Although strongly influenced by the Western music tradition, this model is general enough to represent a large part of the currently digitised music. We call *Music Content Description* (MCD) a description of a music document according to this model. It offers several operations that support major functionalities of a search engine, namely *transformations* corresponding to the classical linguistic operations in text-based search engines, and *ranking*. A set of *features* can be obtained from a MCD thanks to the above mentioned transformations. The feature set presented in the following is by no means exhaustive, and the framework design lets open the possibility to design other features (for instance, harmonic, texture or timbre annotations). The last parts are the core modules of a search engine: *indexing*, *searching*, *ranking*, and on-line identification of fragments that match a pattern query.

5.2 The Music Content Model

The *Music Content Model* (MDM) relies on music notation, seen as an expressive formal language. The MDM provides an abstract vision of digital music documents as structured objects, and supports indexing and search functionalities.

Pitches

We define the domain of musical symbols for representing pitches as follows:

Definition 1 (Domain of musical symbols)**.** *The domain* ***Mus*** *of musical symbols consists of*

1. *the set of pitch names* $\mathtt{e}$ *(A4, C5, F#3, etc.),*
2. *the rest symbol, noted* $\mathtt{r}$*,*
3. *the continuation symbol, noted _ .*

Rhythm structure

Music is a temporal organisation of sounds inside a bounded time range. Notes fall on a set of positions that defines a discrete partitioning of this range. More precisely, the grid results from a recursive decomposition of temporal intervals, yielding a rhythmic organisation which is inherently hierarchical.

Rhythmic decomposition rules can be expressed in a well-known formal language, namely *Context-Free Grammars* (CFG) [92]. As an illustration, the following grammar $\mathcal{G} = (V, \textbf{Mus}, R, S)$ is sufficient to model the rhythmic organisation of the example on Figure 5.1 (with time signature $4/4$). The set of non-terminal symbols is $V = \{S, m, b, q\}$, where S (the initial symbol) denotes a whole music piece, m a measure, b a beat and q a quaver. The terminal symbols belong to **Mus**, the set of music symbols (Def. 1), and R is the following set of rules:

1. $R_0 : S \rightarrow m|m, S$ (a piece of music is a sequence of measures)
2. $r_1 : m \rightarrow b, b, b, b$ (a measure is decomposed in four quarter notes / beats)
3. $r_2 : b \rightarrow q, q$ (a beat is decomposed in two quavers / eighth note)
4. A set $\mathcal{R}^m$ of rules $R^v_e : v \rightarrow e$ where $e \in$ **Mus** is a musical symbol.

Rule R_0 and the set $\mathcal{R}^m$ together determine the temporal structure of music:

- (i) a time range in divided in equal-sized measures, and ii) events only occur at timestamps determined by a parse tree of the grammar. Unambiguous grammars that feature R_0 and $\mathcal{R}^m$ are called *music content grammars* in the following.

Given a music content grammar, we can use its rules to build a hierarchical structure (a parse tree) that models the rhythmic organization of a sequence of musical events.

Definition 2 (Monodic content descriptor)**.** *Let* $\mathcal{G} = (V, \textbf{\textit{Mus}}, R, S)$ *be a music content grammar. A (monodic)* content descriptor *is a parse tree of G. The inner nodes constitute the* rhythm tree*, and the leaves are the (musical)* events*.*

From a content descriptor, it is easy to infer the following properties that will serve as a basis for the indexing process: pitch sequence, temporal partition, and event sequence.

Definition 3 (Pitch sequence)**.** *Let* D *be a content descriptor. The sequence of leaf nodes values in* D *is a string in* ***Mus**** *called the* pitch sequence *of* D *and noted* $PSeq(D)$*.*

Given a time range I, a content descriptor D defines a partitioning of I as a set of non-overlapping intervals defined as follows.

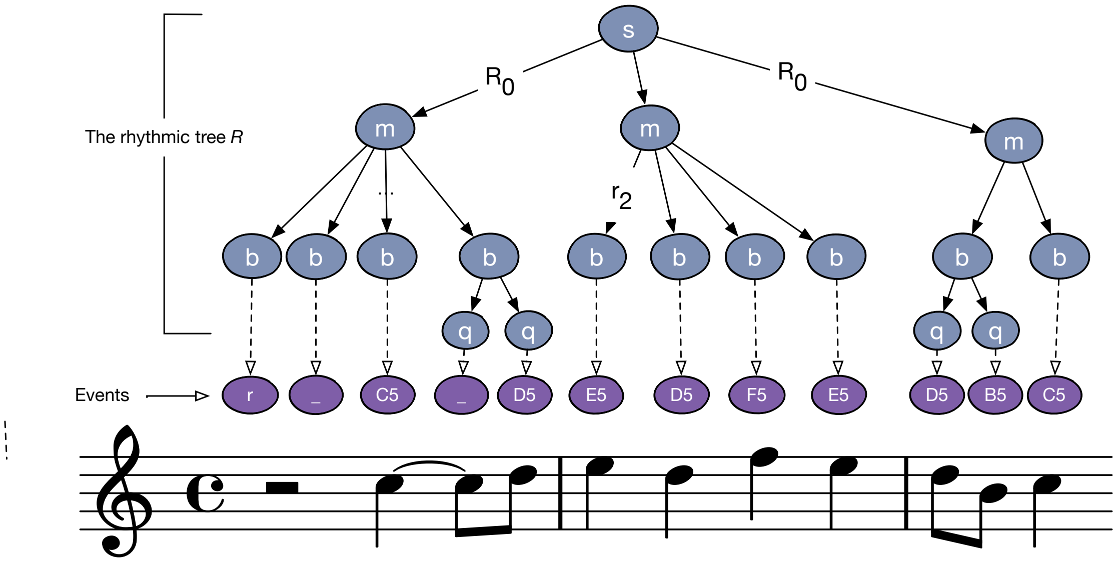

Figure 5.1: The content descriptor for the first notes of the German anthem, *Das Lied der Deutschen* by *Joseph Haydn* (1797), with its events and the rhythmic tree.

Definition 4 (Temporal partition)**.** *Let* $I = [\alpha, \beta[$ *be a time range and* D *a content descriptor. The* temporal partitioning $P(I, D)$ *of* I *with respect to* D *is defined as follows. Let* N *be a node in the rhythmic tree of* D *(recall that the rhythmic tree is* D *without the leaves level).*

1. *If* N *has no children,* $P(I, N) = \{I\}$
2. *If* N *is of the form* $N(N_1, \cdots, N_i)$*,* I *is partitioned in* n *sub-intervals of equal size* $s = \frac{\beta-\alpha}{n}$ *each:* $P(I, N) = \{I_1, \cdots, I_n\}$ *with* $I_i = [\alpha + (i-1) \times s, \alpha + i \times s[$

This partitioning associates to each internal node N of a content descriptor a non-empty interval denoted $itv(I, N)$ in the following and a duration denoted $dur(I, N)$. Each event (leaf node) covers the time interval of its parent in the rhythmic tree.

We adopt the following convention to represent temporal values: the duration of a measure is 1, and the music piece range is n, the number of measures. Both the duration and interval of a node result from the recursive division defined by the rules. The duration of a half note for instance is $\frac{1}{2}$, the duration of a quaver is $\frac{1}{4}$, etc. The duration of a leaf node (event) is that of its parent in the rhythmic tree.

One can finally obtain the *event sequence* by combining both information.

Definition 5 (Event sequence)**.** *Let* D *be a content descriptor and* $[L_1, \cdots, L_n]$ *be the pitch sequence of* D*. Then the sequence* $[(L_1, dur(L_1)), \cdots, (L_n, dur(L_n))]$ *where we associate to each leave its duration is the* event sequence *of* D*, denoted* $ESeq(D)$*.*

Each element in $ESeq(D)$ associates a symbol from **Mus** and a duration. One obtains the sequential representation commonly found in music notation. An explicit representation of the hierarchical structure is however much more powerful than the sequential one. We can use the tree structure for various simplifications, compute similarity measures (see below), or infer strong or weak timestamps from their corresponding branch in the tree. More generally, this general framework allows to derive *features* from content descriptors by extracting, transforming, normalizing specific aspects pertaining to rhythm, domain values, or both.

Non-musical domains, polyphonic music

Our modelling perspective can be extended to other value domains, beyond the class of music symbols, such as opera with lyrics. The content is then modelled with two content descriptors over distinct values domains (e.g., **Mus** and syllables, but chords, textures, or other types of annotation can fit in), which may differ.

The representation of polyphonic music simply consists in a set of monodic content descriptors sharing a same grammar.

Definition 6 (Polyphonic content descriptor)**.** *Given a music content grammar $\mathcal{G}$, a* (polyphonic) content descriptor *is a set of parse trees of $\mathcal{G}$ such that the number of derivations of rule R_0 (in other words, the number of measures) is constant.*

5.3 Features extraction

Several features can be produced from a music content descriptor, such as a chromatic interval feature, a diatonic interval feature, a rhythm feature, and a lyrics feature. This list is not closed, features pertaining to other aspects of music representation (*e.g.,* harmonic) or features obtained from an analytic process may be added, as long as they can be derived from our description model. However, those features are designed for a text-based search engine, each (type of) feature must therefore fulfill the following requirements:

- There exists an *analyzer* associated to each type of feature, that takes a content descriptor as input and produces the feature as output.
- There must exist a *serialization* as a character string, which makes possible the transposition of queries to standard text-based search supported by the engine.
- Finally, each feature must be equipped with a *scoring function* that can be incorporated into the search engine for ranking purposes.

We use the famous song *My way* as an example to illustrate our features (see Fig. 5.2)[1].

[1] The song is the English version of the French song *Comme d'habitude*, written by Claude François and Jacques Revaux (1967). The English lyrics are by Paul Anka (1969).

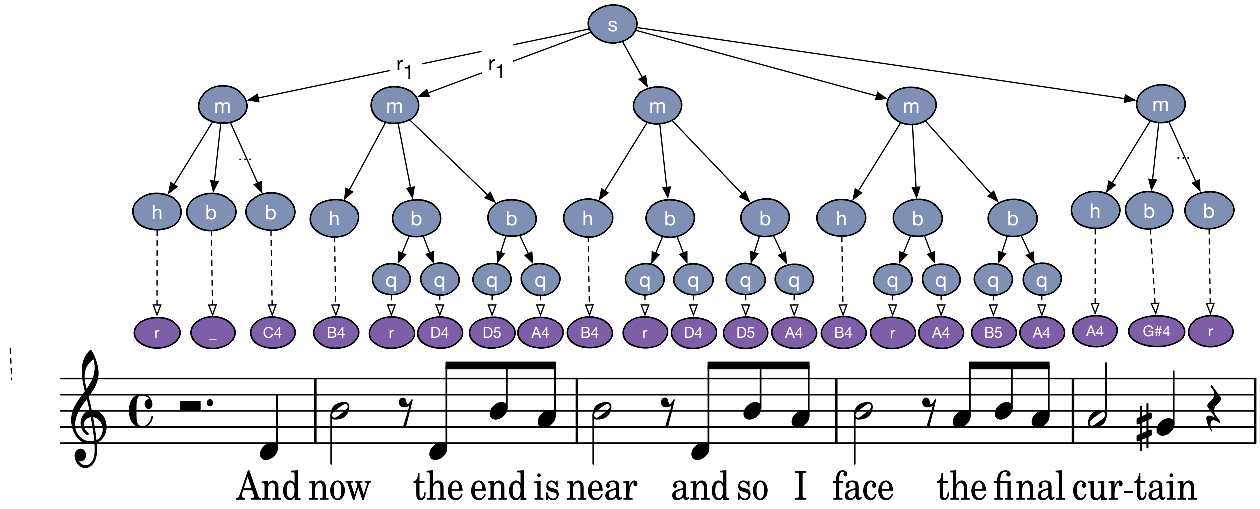

Figure 5.2: Content descriptor of *My way*

5.3.1 Chromatic interval feature (CIF)

The feature analyzer A_{CIF} relies on the following simplification of a pitch sequence:

1. All repeated values from $PSeq(D)$ are merged in a single one.
2. Rest and continuation symbols are removed.

One obtains a simplified descriptor that essentially keeps the sequence of non-null intervals. Fig. 5.3 shows such a sequence, resulting from the analysis of *My way*. Note that the two consecutive A4 near the end have been merged, and all rests removed.

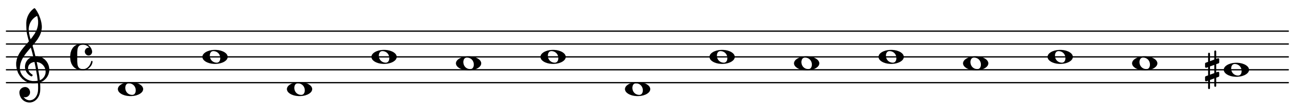

Figure 5.3: *My way*, after the feature extraction by the appropriate analyzer.

Definition 7 (Chromatic Interval Feature)**.** *Given a content descriptor D, the* chromatic interval feature *(CIF) $A_{CIF}(D)$ is the sequence of the semitones gaps between two consecutive pitches in the simplification of $PSeq(D)$.*

When the CIF analyzer A_{CIF} is applied to the sequence of Fig. 5.3, one obtains the following feature.

$$< 9, -9, 9, -2, 2, -9, 9, -2, 2, -2, 2, -2, -1 >$$

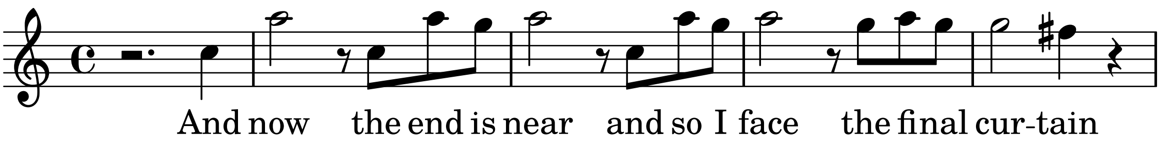

Figure 5.4: *My way*, transposed.

It is worth mentioning that we may obtain the same CIF from initially distinct music descriptors. Fig. 5.4 shows for instance a *transposed* version of *My way*, more suitable to a female voice. The CIF is invariant. The feature is also robust with respect to more involved changes. Fig. 5.5 shows the initial version of the tune. The lyrics in French imply a slightly distinct rhythmic structure. However, the sequence of intervals remains identical, and so does the CIF.

Figure 5.5: French variant of *My way* (*Comme d'habitude*, first phrase).

The descriptors shown in Figures 5.2, 5.4, and 5.5 *match* with respect to their respective chromatic features. The matching of two descriptors is highly dependent on the analyzer.

Among other possible features, we could have taken the sequence of pitch names, in which case transposed scores would not match (we would miss results that seem intuitive). Another feature would accept unisons (i.e., repeated notes yielding intervals with 0 semi-tones). Then, in our example, the French version (Fig. 5.5) would no longer match with the English version of *My way*.

5.3.2 Diatonic interval feature (DIF)

We keep the same simplification phase already used for A_{CIF}. Fig. 5.6 shows the second phrase, which slightly differs from the first one. If we compute the chromatic interval feature, one obtains the following sequence:

$$< 8, -8, 8, -1, 1, -8, 8, -1, 1, 4, -2, -5, 3, -1 >$$

which is distinct from that of the first phrase (see above).

Figure 5.6: *My way*, second phrase.

If we count the number of steps between pitches, and no longer the number of semi-tones, we observe that the second phrase starts with a 5-steps interval (from E4 to C5), continues with a descending one-step (from C5 to B4), etc. Counting the number of steps between the nominal value of pitches yields what is known as *diatonic intervals* in music theory. In this perspective, the first interval of the first phrase and of the second phrase do match: they are both sixths, major in the first case, minor in the second one. So does the second interval (a second, minor in the first case, major the second case). Both phrases are essentially similar, and would be perceived as such by a listener.

Definition 8 (Diatonic Interval Feature)**.** *Given a content descriptor D, the* diatonic interval feature *(DIF) $A_{DIF}(D)$ is the sequence of interval names between two consecutive pitches in the simplification of $PSeq(D)$.*

Assuming that the set of interval names is {U(nison), S(e)c(ond), T(hird), Fo(urth), Fi(fth), Si(xth), Se(venth) and O(ctave)} and that an ascending interval is coded with a $+$, a descending with a $-$, one may apply this definition to the descriptor of Fig. 5.3. The following sequence is obtained:

$$< Si+, Si-, Si+, Sc-, Sc+, Si-, Si+, Sc-, Sc+, Si-, Si+, Si-, Si- >$$

The first and second phrases of *My way* match with respect to this feature, and continue to match with any transposition (Fig. 5.4) or rhythmic variants (Fig. 5.5).

5.3.3 Rhythmic feature (RF)

We rely on a simplified rhythmic representation. Given a content descriptor R, its *temporal partition* (see Def. 4) gives the respective durations of the events. Consider once again the first phrase of *My way* (Fig. 5.2), ignoring the initial rest. It starts with a quarter note, followed by a half-note: the ratio (i.e., the multiplication to obtain the second duration value from the first one) is 2. Then comes a 1-eighth duration, hence a ratio equal to $\frac{1}{8}$, followed by three eight-notes, hence three times a neutral ratio of 1, etc. The sequence of these ratio is our description of rhythm.

Definition 9 (Rhythmic feature)**.** *Given a content descriptor D and its leaves $[L_1, L_2, \cdots, L_n]$, the* rhythmic feature *(RF) $A_{RF}(D)$ is a sequence $[r_1, \cdots, r_{n-1}]$ such that $r_i = \frac{dur(L_{i+1})}{dur(L_i)}, \forall i \in [1, n-1]$.*

The rhythmic feature of the first phrase of *My way* (ignoring the initial rest) is

$$< 2, \frac{1}{8}, 1, 1, 8, \frac{1}{8}, 1, 1, 8, \frac{1}{8}, 1, 1, 8, \frac{1}{2} >$$

5.3.4 Lyrics feature (LF)

The lyrics feature is the simplest concept : it consists of the text of the tune (if any). Since it consists of purely textual information, it is subject to the traditional transformations (tokenization, lemmatization, etc.) operated by search engines.

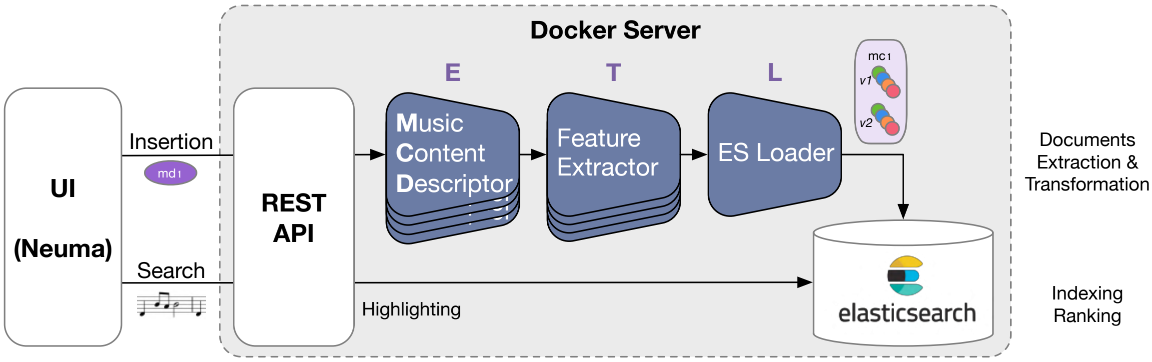

Figure 5.7: A global view of the *MusicSearch* architecture, for symbolic music.

5.4 Implementation

5.4.1 Implementation of MusicSearch framework

The current implementation of the *MusicSearch* framework is illustrated on Figure 5.7. The main components are: 1) an ETL (Extract/Transform/Load) process that receives music documents and produces their musical features, 2) an *Elasticsearch* server[2] that indexes music features, supports searches and ranks results, and 3) implementation of several utility functions, including the matching occurrences identification. All these modules are written in Python.

5.4.2 Reproducibility and Open Science

A live version of the search engine is available on the Neuma platform[3], where the pattern mining can be experimented on a wide range of opera, from Bach's chorals to more recent French popular tunes. We also offer a publicly available Docker image[4]. The work is fully reproducible, with open access to code on Github[5] under the GNU General Public License v3.0.

5.5 Conclusion

The framework presented in this section is detailed in a journal article currently under revision. Some details omitted in this report are presented in this article, which focuses on the search-engine aspect (and not only on *pattern-mining*).

While we simultaneously develop the software implementation and the theoretical model, we think it is already mature enough to be used as a large-scale pattern-mining tool. As we mentioned above, some patterns dimensions (features) have been developed, some new ones may

[2]Elasticsearch Engine: https://elastic.co
[3]See http://neuma.huma-num.fr.
[4]Docker Server image: https://hub.docker.com/repository/docker/traversn/scoresim
[5]Components implementation: https://github.com/cedric-cnam/scoresim

be written (e.g., to fit with the work from others Polifonia members). The extensibility and user-friendliness of this search engine make it a tool of choice for the Polifonia members (and, more generally, the MIR community).

However, some work on the dataset to be searched and the formats to be included in this framework are on-going. They will lead to very interesting statistical analysis reports, about various corpora (such as the ones from Neuma).

6 Harmonic similarity for the INTERLINK pilot

The work described in this chapter introduces a novel harmonic similarity function based on recurring chord structures that are shared by musical pieces. This method is part of `musilar` – a larger computational ecosystem for music similarity, that is currently under development for the INTERLINK pilot. This contribution was first tested and utilised for the implementation of the harmonic similarity agent that was presented, together with a prototype of the first Polifonia KG, at the Sonar Festival in October 2021, in the context of the Music and AI event[1].

6.1 Related work

Harmonic progression is one of the cornerstones of tonal music composition and is thereby essential to many musical styles and traditions. From a computational perspective, the ability to automatically establish links between musical pieces sharing harmonic patterns can potentially support information retrieval pipelines (e.g. querying a music corpus using harmonic progressions), the detection of cover songs (performances of the same composition) or plagiarism, but can also be used for music exploration and recommendation, the analysis of large musical corpora to test musicological hypothesis (or to collect insights to inform this process), and so forth.

Despite the numerous applications enabled by these approaches, the design of novel algorithms for harmonic similarity has not received particular attention in the last ten years. To the best of our knowledge, the current state of the art methods for this task are still those pioneered by Baas de Haas, namely, the tonal pitch step distance (TPSD) [104] and the chord sequence alignment system (CSAS) [105], which are outlined as follows.

The TPSD is a perceptually and musicologically-grounded distance function generalising Lerdahl's Tonal Pitch Space (TPS) [106] – a model of tonality that fits musicological intuitions and correlates well with empirical findings from music cognition [107]. More precisely, the TPSD implements a scoring mechanism using a part of the TPS as the main musical model to calculate the distance between two arbitrary chords. Intuitively, given two chords, the function considers the number of steps on the circle of fifths between their roots, and the amount of overlap/agreement between the corresponding chord structures, as well as their relation to the global key. When generalised over an entire chord progression – an ordered sequence of chords, the TPS distance is computed between every chord and the key of the sequence. This originates into a step function profiling the harmonic properties of a musical piece. As a consequence, the distance between two chord progressions is defined as the minimal area between their step functions over all possible horizontal circular shifts. This is done to take into account the "shape of the function" rather than their actual value – an effective way to achieve transposition invariance.

[1] https://aimusicfestival.eu

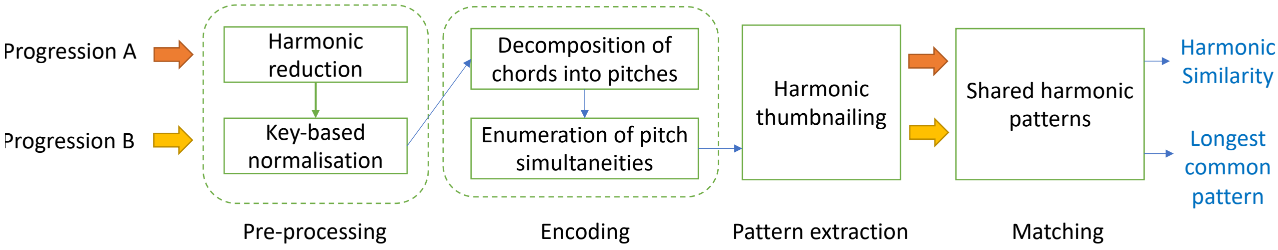

Figure 6.1: Schematic illustration of the main steps for the computation of the proposed harmonic similarity from two chord progressions.

Another notable approach is the Chord Sequence Alignment System (CSAS), using string matching techniques to compute similarity scores between strings (sequences of symbols) representing chords or distances between chords and key. In particular, the local alignment algorithm by Smith and Waterman [108] is used to locate and extract a pair of areas/regions from two given strings (generally defined as sequences of arbitrary symbols) that exhibit the highest similarity with each other. Thereafter, using a dynamic programming method, a similarity score for the two progressions is calculated based on the minimum number of elementary operations (deletion, insertion or substitution of a symbol) needed to transform one sub-string into the other.

From a systematic comparison of the two methods on the cover detection task, using chord encodings of different granularity, CSAS was found to achieve better performance than TPSD [107]. Nevertheless, TPSD has a more intuitive formulation, stronger musicological interpretation, and is also more efficient compared to the quadratic time complexity of CSAS.

In addition to the former approaches, [109] proposed a system using a generative grammar of tonal harmony to formally describe chord sequences. In doing so, a parser defined from the grammar would produce concrete syntactic trees representing harmonic analyses of a given chord progression. To enable the comparison of two parse trees, resulting from different pieces, the authors defined a method that constructs a tree containing all their shared tree structures. This allows to quantify tree-specific properties that can be further analysed yielding several similarity measures. Although this system is particularly appealing for musicological studies, the current grammar is not expressive enough to support certain chord sequence. As a result, chord sequence that are deemed as ungrammatical cannot be parsed, hence jeopardising the computation of the harmonic similarities.

6.2 Methodology

The state of the art methods reviewed in the previous sections have been demonstrated to be effective at detecting cover songs exclusively from their chord progressions – a specific task in music information retrieval (MIR). Nonetheless, the choice of harmonic similarity function strictly depends on the definition of similarity per se (i.e. what makes two harmonic progression similar to each other), and the kind of analyses the overall system should enable (i.e. what should inflate or deflate the similarity score). For instance, a local similarity measure – identifying specific regions from harmonic sequences that can be related to each other, would accommodate different use cases when compared to a global method – seeking to establish harmonic dependencies among tracks only when their harmonic profiles are globally aligned. In other words, a method that excels on a certain task does not necessarily generalise to accommodate other use cases.

The harmonic similarity proposed in this chapter is formulated in a way to emphasise shared repeated patterns among two arbitrary symbolic sequences, hence it provides a general framework for the analysis of symbolic streams based on their local structures. As illustrated in Figure 6.1, the domain-specific parts are the pre-processing and encoding steps, whereas the downstream part of the pipeline – including pattern extraction and matching operations, is completely general.

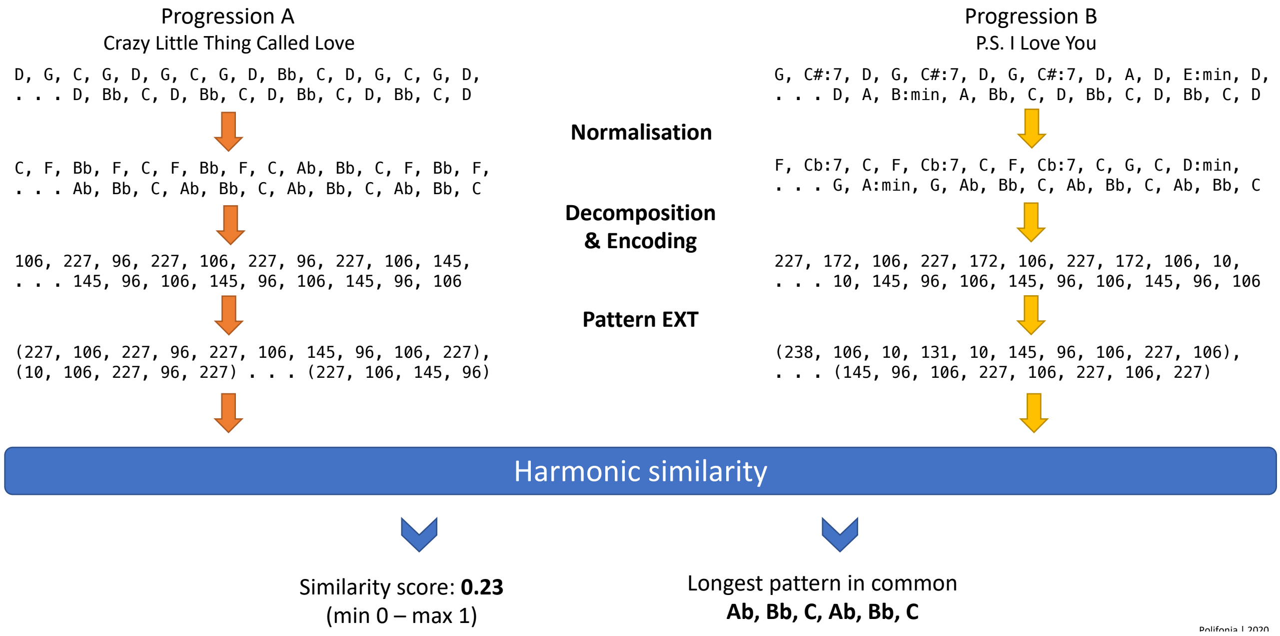

Figure 6.2: A real example of the workflow on two different harmonic progressions.

Harmonic reduction. Chord progression are simplified before comparison – the bass note is removed (as empirically demonstrated in [107], this operation improves the generalisation capabilities of the next steps, thereby producing more consistent similarity scores), and consecutively repeated chords are removed. This provides a "bird's eye view" on the global harmonic properties of each piece.

Key-based normalisation. Chords labels/classes in a progression need to be contextualised according to the key of the piece (defined by the tonic and the scale) before any comparison is possible. Therefore, all chord progressions are transposed to the same key: C major. This last transformation concludes the pre-processing operations.

Decomposition of chords. The normalised harmonic sequences are then prepared for the encoding step, so that they can be used as input to the any computational procedure. Rather than further simplifying the symbolic musical content, a new encoding procedure was designed to retain the fundamental internal structure of each chord. More precisely, every chord label is decomposed into its pitch constituents – the individual pitches it is made of. For example, a C major is encoded as a multi-hot vector where the elements corresponding to the active pitches are equal to 1; all the others are 0.

Enumeration of pitch simultaneities. To reduce the complexity of any potential polyphonic model using such sparse local representations of chords, each unique decomposition is then assigned to an index (an integer value). As it can be observed, this is akin to the common encoding approach used in natural language processing for word tokens. This results into *chord tokens* defined over the vocabulary of all possible chord decompositions.

Harmonic thumbnailing. To identify the areas/regions of chord progressions that can be deemed as "harmonically memorable", we extract the n-grams of all possible orders – starting from $n \geq 3$ (i.e. from tri-grams), that repeat at least once within the progression. We call them "harmonic thumbnails", as they represent harmonic structures per se.

Shared harmonic patterns. Finally, chord progressions are compared for similarity based on the agreement between their harmonic thumbnails. In particular, the longest harmonic structures they share is compared to the order of the longest thumbnail that occurs within each progression, independently. Therefore, depending on the harmonic patterns the two chord progressions have in common – in relation to their internal structures, the similarity function will return a value between 0 and 1 (the higher the value, the stronger the similarity) together with the longest harmonic patterns they share.

The relatively simple and intuitive formulation of the similarity function – generalising to any symbolic sequence, does not pose any particular assumption in the pattern extraction and the matching process that could limit its applicability or specialise it to a certain musical genre/style. The major element of concern is represented by the data pre-processing and encoding steps, as music experts may argue that transposition to the same key would alter the musical texture in a way to compromise its consistency with the original (non-transposed) version. In addition, the decomposition encoding may also raise some concerns from a music theory perspective, as two distinct chord labels producing the very same sound (their pitch decomposition is identical) could have different harmonic functions/contexts. Both these critical points can be thought of as simplifications rather than contextual assumptions, although we are still in the process of validating their global suitability and collect expert feedback to extend/improve both steps, if necessary.

A concrete example of the main steps explained above is reported in Figure 6.2 for two pop/rock pieces: *"Crazy Little Thing Called Love"* by Queen and *"P.S. I Love You"* by The Beatles.

6.3 Preliminary results

To evaluate the proposed harmonic similarity method, two experiments were carried out on a collection of music datasets providing high-quality chord annotations of audio tracks belonging to different genres and styles (c.f. Section 6.3.1). Although the current experimental design does

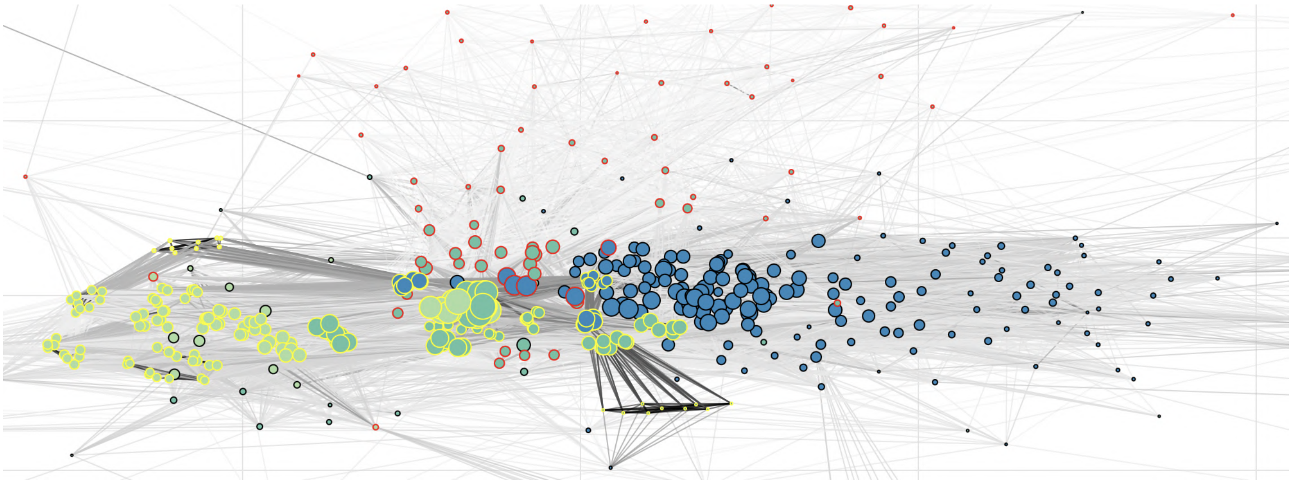

Figure 6.3: Illustration of the harmonic graph, encoding all the harmonic dependencies established between tracks in the three music collections, using the proposed similarity function.

not yet involve an expert evaluation of the similarity scores – needed to fully validate their plausibility and meaningfulness from a musicological perspective, we adapted and extended standard evaluation methodologies [104, 107] based on the cover song detection task. More precisely, the pair-wise harmonic similarity was computed between all tracks in the music datasets, and a graph encoding such similarities was created to formally describe this analysis (c.f. Section 6.3.2). In doing so, community detection methods were applied on the resulting network to discover groups of tracks sharing similar harmonic dependencies – a property that is expected to be intrinsically related to the genre/style of the music under analysis. In other words, the main research question behind this experiment can be framed as follows: "*can such groups be associated to a particular genre?*", which is addressed in Section 6.3.3. Second, we isolate those recordings that are associated to the same composition[2] and rank their harmonic similarity in relation to the other tracks. We expect performances of the same composition to all have the highest rank with each other. In the harmonic network, assuming that the harmonic similarity is effective, this would result in a number of cliques – one for each composition. This experiment is thus akin to the cover song detection task, and the results are briefly outlined in Section 6.3.4.

6.3.1 Chord datasets

The selection of datasets was driven by the need of having professional and validated chords annotations. This allowed to focus only on the harmonic similarity, without the need to deal with automatic chord recognition. Furthermore, given the purpose of this work, a further requirement of the experiment was to deal with tracks from different music genres and styles. Given these requirements, the choice fell on three datasets, each relating to a musical genre:

[2]For composition, we refer to a musical work usually represented in notated form (e.g. score) independently from all its possible interpretations or performances.

- Isophonics [110] for pop-rock music;
- Audio-aligned jazz harmony dataset (JAAH) [111] for jazz music;
- Schubert Winterreise [112] for classical music.

Isophonics [3] is a hosting of the software and the data produced by the Centre for Digital Music (C4DM) of the Queen Mary University of London. Among the datasets that Isophonics makes available, we decided to use three: Queen, Beatles and Michael Jackson. Each of these datasets contains chord, onset and segmentation annotations. Similarly, JAAH [113] provides meter, structure, and chords annotation for each audio track. Finally, the Schubert Winterreise dataset [114], provides various representations and annotations of Franz Schubert's song cycle Winterreise. For each of the recording of each representation, the dataset provides singer's lyrics, sheet music in different machine-readable formats, and audio recordings of nine performances, as well as musicological annotations describing tonal and structural characteristics, such as chord annotations, local and global annotations of musical keys, and segmentations into structural parts.

In total, 525 tracks were considered for this work, combining all datasets. Nevertheless, before proceeding with the analysis of the data, a cleaning and alignment phase had to be carried out for the chords annotations, all of which were transformed into JAMS format [115].

6.3.2 The harmonic network

As illustrated in Figure 6.3.2, in the harmonic network nodes correspond to tracks in our music datasets, whereas edges connect nodes if their value of harmonic similarity is greater than 0 (an harmonic match was found). To simplify the inspection, a grey-scale colourmap visually projects the value of harmonic similarity expressed by edges: from light grey (low similarity) to plain black (high similarity). Instead, nodes are sized according to their degree – the number of connections/edges they have[4]. A distinct border colour also differentiate nodes according to the music dataset they come from: yellow border for tracks in Schubert-Winterreise; black border for Isophonics; and red for JAAH.

6.3.3 Experiment 1: unveiling network structures

To address the first research question, a community detection algorithm was run on the harmonic network, and in particular, the Clauset-Newman-Moore modularity maximisation procedure [116]. This allowed to associate each node (track/progression) to a community/cluster based on the harmonic dependencies established with the other tracks. In Figure 6.1, this is illustrated by the (inner-)colour of each node, uniquely denoting a community.

From a closer view of the graph (Figure 6.4), we notice that clusters emerging from the community detection procedure can already express the relationship between nodes and their dataset –

[3] http://isophonics.net/

[4] As the graph is undirected, considering that similarity is symmetric by definition, the degree is equal to the number of tracks sharing harmonic structures.

meaning that our harmonic similarity function permits to implicitly detect the genres of tracks from their harmonic progression. In fact, each dataset gets clustered in separate communities, with a relatively small amount of overlap between them. In that regard, what is particularly interesting to inspect, is to look at those tracks that end up in a cluster different from that of their datasets.

6.3.4 Experiment 2: performance-composition detection

By exploiting the characteristics of the Schubert Winterreise dataset, which contains multiple performances of the same pieces, it was possible to validate the similarity function developed for this work. As can be observed from the graph (Figure 6.5), several performances of the same piece are grouped together in cliques. These groups of nodes within the graph are closely connected to each other, both in terms of the number of connections and the very high similarity value that binds them. However, it is possible to observe that the different performances result in small differences in the harmony of the piece, leading to a minimal shift of the nodes on the Cartesian axes of the graph.

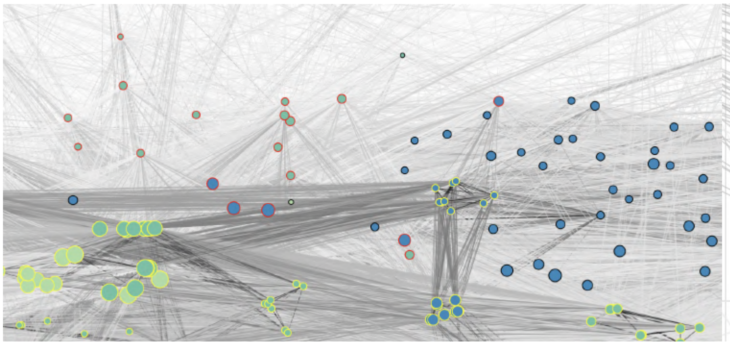

Figure 6.4: Collections as communities (clusters).

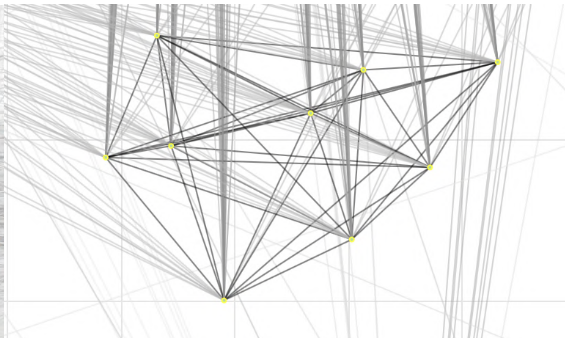

Figure 6.5: A composition clique.

6.4 FAIRness and reproducibility

All experiments presented in the previous sections were carried out using data from three different datasets, namely Isophonics, JAAH and Schubert Winterreise. All these datasets are available and downloadable from their respective repositories. The chord notation data of each dataset was then converted and transformed into JAMS format, a format designed to provide simplicity, structure and sustainability to the data. The code for converting the chords into JAMS format is available in a separate folder on GitHub`https://github.com/polifonia-project/mir-resources`.

All the resources used, i.e. three datasets and their metadata, as well as the converted chords are stored in a GitHub repository `https://github.com/polifonia-project/datasets` and can be freely accessed. In addition, the entire code developed for all described steps is available in the repository `https://github.com/polifonia-project/lharp`. Moreover, the repository has been associated to a Zenodo identifier (https://doi.org/10.5281/zenodo.5769546, which provides a unique link to the information in the repository and its version. All of the above code is also released under a Creative Commons (CC BY-NC 4.0) not-for-profit, which guarantees re-use of the code and data.

All the experiments proposed for this section are therefore reproducible in all their parts and the data to perform them is available. It is also possible to use the tools provided to run new experiments on other datasets, thus further evaluating the proposed approach.

7 Conclusions

We have reported on initial work in four main strands during the first 11 months of the project. Thus, none of these strands of work is declared as finished. However, we have put in place necessary foundations and achieved several early results:

- Review of existing work across several strands of research;
- Pattern algorithms for Irish folk music, with initial analyses of relationships among tunes and informal validation of these analyses, and initial generalisation of this code to run on Dutch corpora, addressing the *Mark* and *William* personas and the TUNES pilot; also foundational work on root note detection;
- A functional faceted search engine incorporating pattern matching, addressing the *Sethus* and *Sophia* personas and the FACETS pilot;
- A demo application which makes connections among pieces across multiple corpora, e.g. based on harmonic analyses, addressing the *Sonia* and *Keith* personas and the INTERLINK pilot.

7.1 Future work

In future work, each strand of work will continue to deliver new results according to its own methods. However, opportunities for links between projects will also arise. For example, the FONN software (Chapter 3) has been generalised to run on the Meertens Tune Collections (Chapter 4). The next step will be to use such software to investigate inter-corpus patterns. An example research question might be: which Irish tunes have their origins in Dutch tunes, which vice versa, and which Irish and Dutch tunes have common origins elsewhere?

7.1.1 Data formats

As the Polifonia ecosystem grows and becomes more interconnected, we expect to be able to ingest more input formats, e.g. MEI `https://music-encoding.org/about/`

7.1.2 Pattern Knowledge Graphs

Knowledge graphs are central to the Polifonia vision. In WP3 to date knowledge graphs have been used only in the strand of research on harmonic analysis (Chapter 6), which integrates with the Polifonia prototype knowledge graph (see WP2 and Deliverable D2.1). Other strands of research have worked directly with score-level data, e.g. abc or MIDI representations. In future, there are three possibilities to integrate knowledge graphs with pattern research in WP3. Firstly, our datasets can be converted to knowledge graph formats and this is planned in some cases. Secondly, algorithms for extracting patterns can create or help to populate knowledge graphs. The "dimensions" of pattern presented in Chapter 2 serve as a starting point for discussion of pattern ontology.

In the other direction, knowledge graphs which already contain some musical annotations (e.g. structural information) may enable smarter pattern extraction algorithms.

Bibliography

[1] L. A. Steen, "The science of patterns," *Science*, vol. 240, no. 4852, pp. 611–616, 1988.
[2] C. Alexander, *A pattern language: towns, buildings, construction*. Oxford university press, 1977.
[3] E. R. Toussaint and G. T. Toussaint, "What is a pattern," *Proceedings of Bridges*, pp. 293–300, 2014.
[4] M. P. Mattson, "Superior pattern processing is the essence of the evolved human brain," *Frontiers in neuroscience*, vol. 8, p. 265, 2014.
[5] D. Meredith, "Point-set algorithms for pattern discovery and pattern matching in music," in *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2006.
[6] R. F. Goldman, "Review of the record varèse: Ionisation; density 21.5; intégrales; octandre; hyperprism; poème electronique, by r. craft & e. varèse," *Musical Quarterly*, vol. 47, pp. 133–134, 1961.
[7] D. R. Hofstadter and E. Sander, *Surfaces and essences: Analogy as the fuel and fire of thinking*. Basic Books, 2013.
[8] P. Toiviainen and T. Eerola, "Self-organizing map of the essen collection," in *ISSCM*, 2001.
[9] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, 1972.
[10] D. Conklin, "Discovery of distinctive patterns in music," *Intelligent Data Analysis*, vol. 14, no. 5, pp. 547–554, 2010.
[11] S. Dubnov, "Deep music information dynamics," in *Joint Conference on AI Music Creativity*, B. L. T. Sturm and A. Elmsley, Eds., 2020, https://boblsturm.github.io/aimusic2020/papers/CSMC__MuMe_2020_paper_15.pdf.
[12] O. Lartillot, "Automated motivic analysis: An exhaustive approach based on closed and cyclic pattern mining in multidimensional parametric spaces," in *Computational Music Analysis*. Springer, 2016, pp. 273–302.
[13] S. Dixon, F. Gouyon, G. Widmer *et al.*, "Towards characterisation of music via rhythmic patterns." in *ISMIR*, 2004.
[14] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. Cambridge, MA: MIT Press, 1983.
[15] A. Marsden, "Schenkerian analysis by computer: A proof of concept," *Journal of New Music Research*, vol. 39, no. 3, pp. 269–289, 2010.
[16] F. Simonetta, F. Carnovalini, N. Orio, and A. Rodà, "Symbolic music similarity through a

graph-based representation," in *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, 2018, pp. 1–7.

[17] J. R. Cowdery, "A Fresh Look at the Concept of Tune Family," *Ethnomusicology*, vol. 28, no. 3, p. 495, Sep. 1984. [Online]. Available: https://www.jstor.org/stable/851236?origin=crossref

[18] A. Hillhouse, "Tradition and innovation in Irish instrumental folk music," Master's thesis, The University of British Columbia,, Vancouver, Canada, 2005.

[19] P. van Kranenberg, "A computational approach to content-based retrieval of folk song melodies," phdthesis, Utrecht University, Utrecht, Netherlands, 2010, iSBN: 9789039353936 OCLC: 1000569214.

[20] J. Coker, *Patterns for jazz*. Alfred Music Publishing, 1970.

[21] K. Frieler, F. Höger, M. Pfleiderer, and S. Dixon, "Two web applications for exploring melodic patterns in jazz solos." in *ISMIR*, 2018, pp. 777–783.

[22] M. Pfleiderer, K. Frieler, J. Abeßer, W.-G. Zaddach, and B. Burkhart, Eds., *Inside the Jazzomat - New Perspectives for Jazz Research*. Schott Campus, 2017.

[23] R. Clifford, M. Christodoulakis, T. Crawford, D. Meredith, and G. A. Wiggins, "A fast, randomised, maximal subset matching algorithm for document-level music retrieval." in *ISMIR*, 2006, pp. 150–155.

[24] E. Ukkonen, K. Lemström, and V. Mäkinen, "Geometric algorithms for transposition invariant content-based music retrieval," in *ISMIR*, 2003.

[25] A. Gangemi and V. Presutti, "Towards a pattern science for the semantic web," *Semantic Web*, vol. 1, no. 1, 2, pp. 61–68, 2010.

[26] G. Loy, *Musimathics: The mathematical foundations of music*. Cambridge, USA: The MIT Press, 2006, vol. 1.

[27] D. Meredith, Ed., *Computational Music Analysis*. Cham, Switzerland: Springer International Publishing, 2016. [Online]. Available: http://link.springer.com/10.1007/978-3-319-25931-4

[28] B. H. Bronson and F. J. Child, *The traditional tunes of the Child ballads: with their texts, according to the extant records of Great Britain and America*. Princeton: Princeton University Press, 1959.

[29] N. Carolan, "'Because it's our own': Breandán Breathnach 1912-85," 2005. [Online]. Available: https://journalofmusic.com/focus/because-it-s-our-own-breandan-breathnach-1912-85

[30] B. Breathnach, *Folk Music and Dances of Ireland*. Dublin, Ireland: The Talbot Press, 1971.

[31] F. Vallely, Ed., *The companion to Irish traditional music*, 2nd ed. Cork, Ireland: Cork University Press, 2011, oCLC: ocn772584731.

[32] B. Breathnach, "Between the Jigs and the Reels," 1982. [Online]. Available: http://msikio.online.fr/Breathnach/breandn.htm

[33] ——, *Ceol rince na hÉireann, cuid I*, ser. Ceol rince na hÉireann. Dublin, Ireland: Oifig an tSoláthair, 1963, no. 2.

[34] ——, *Ceol rince na hÉireann, cuid II*, ser. Ceol rince na hÉireann. Dublin, Ireland: Oifig an tSoláthair, 1976, no. 2.

[35] ——, *Ceol rince na hÉireann, cuid III*, ser. Ceol rince na hÉireann. Dublin, Ireland: An Gúm, 1985, no. 3.

[36] B. Breathnach and J. Small, *Ceol rince na hÉireann, cuid IV*, ser. Ceol rince na hÉireann. Dublin, Ireland: An Gúm, 1996, no. 4.

[37] ——, *Ceol rince na hÉireann, cuid V*, ser. Ceol rince na hÉireann. Dublin, Ireland: An Gúm, 1999, no. 5.

[38] M. Ó Suilleabháin, "The creative process in Irish traditional music," in *Irish musical studies 1: musicology in Ireland*, ser. Irish musical studies. Dublin, Ireland: Irish Academic Press, 1990, vol. 1, pp. 117–130.

[39] Z. Juhász, "A systematic comparison of different European folk music traditions using self-organizing maps," *Journal of New Music Research*, vol. 35, no. 2, pp. 95–112, 2006, publisher: Routledge _eprint: https://doi.org/10.1080/09298210600834912. [Online]. Available: https://doi.org/10.1080/09298210600834912

[40] D. Conklin and C. Anagnostopoulou, "Comparative Pattern Analysis of Cretan Folk Songs," *Journal of New Music Research*, vol. 40, no. 2, pp. 119–125, 2011, publisher: Routledge _eprint: https://doi.org/10.1080/09298215.2011.573562. [Online]. Available: https://doi.org/10.1080/09298215.2011.573562

[41] K. Neubarth and D. Conklin, "Contrast Pattern Mining in Folk Music Analysis," in *Computational Music Analysis*, D. Meredith, Ed. Cham, Switzerland: Springer International Publishing, 2016, pp. 393–424. [Online]. Available: https://doi.org/10.1007/978-3-319-25931-4_15

[42] G. Tzanetakis, A. Kapur, W. Schloss, and M. Wright, "Computational ethnomusicology," *Journal of Interdisciplinary Music Studies*, vol. 1, pp. 1–24, Jan. 2007.

[43] B. Janssen, "Retained or lost in transmission? Analyzing and predicting stability in Dutch folk songs," phdthesis, University of Amsterdam, Amsterdam, The Netherlands, 2018.

[44] B. Duggan, "Tunepal: the traditional musician's toolbox," in *Proceedings of the second workshop on eHeritage and digital art preservation - eHeritage '10*. Firenze, Italy: ACM Press, 2010, p. 25. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1877922.1877931

[45] B. Duggan and B. O'Shea, "Tunepal: searching a digital library of traditional music scores," *OCLC Systems & Services: International digital library perspectives*, vol. 27, no. 4, pp. 284–297, Oct. 2011. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/10650751111182597/full/html

[46] P. Beauguitte, "Music Information Retrieval for Irish Traditional Music Automatic Analysis of Harmonic, Rhythmic, and Melodic Features for Efficient Key-Invariant Tune Recognition,"

phdthesis, TU Dublin, Dublin, Ireland, 2019, publisher: Technological University Dublin. [Online]. Available: https://arrow.tudublin.ie/sciendoc/234

[47] R. A. Wagner and M. J. Fischer, "The String-to-String Correction Problem," *Journal of the ACM*, vol. 21, pp. 168–173, 1974.

[48] D. Ó Maidín, "A programmer's environment for music analysis," phdthesis, University of Limerick, Limerick, Ireland, 1995, publisher: Unpublished. [Online]. Available: http://rgdoi.net/10.13140/2.1.3387.2969

[49] F. Simonetta, S. Ntalampiras, and F. Avanzini, "Multimodal music information processing and retrieval: Survey and future challenges," Milan, Italy, 2019, pp. 10–18.

[50] P. Rigaux and N. Travers, "Scalable searching and ranking for melodic pattern queries," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*, Delft, Netherlands, 2019, pp. 801–808.

[51] F. Foscarin, F. Jacquemard, and R. Fournier-S'niehotta, "A diff procedure for music score files," in *6th International Conference on Digital Libraries for Musicology*. The Hague Netherlands: ACM, Nov. 2019, pp. 58–64. [Online]. Available: https://dl.acm.org/doi/10.1145/3358664.3358671

[52] P. Lavin, "A similarity matrix for Irish traditional dance music," Master's thesis, TU Dublin, Dublin, Ireland, 2011.

[53] R. Hillewaere, B. Manderick, and D. Conklin, "Global feature versus event models for folk song classification," in *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009*, Kobe, Japan, 2009, pp. 729–734.

[54] ——, "Alignment methods for folk tune classification," in *Data Analysis, Machine Learning and Knowledge Discovery*, M. Spiliopoulou, L. Schmidt-Thieme, and R. Janning, Eds. Cham, Switzerland: Springer International Publishing, 2014, pp. 369–377, series Title: Studies in Classification, Data Analysis, and Knowledge Organization. [Online]. Available: http://link.springer.com/10.1007/978-3-319-01595-8_40

[55] P. Rigaux and N. Travers, "Scalable searching and ranking for melodic pattern queries." in *ISMIR*, 2019, pp. 801–808.

[56] K. Frieler, "Computational Melody Analysis," in *Inside the Jazzomat. New Perspectives for Jazz Research*. California, USA: Schott Campus, 2018, pp. 41–84.

[57] C. Anagnostopoulou, M. Giraud, and N. Poulakis, "Melodic contour representations in the analysis of children's songs," in *3rd International Workshop on Folk Music Analysis*, ser. Wiskunde en Informatica, P. v. Kranenburg, C. Anagnostopoulou, and A. Volk, Eds. Amsterdam, Netherlands: Meertens Institute and Utrecht University Department of Information and Computing Sciences, Jun. 2013, pp. 40–43. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00826427

[58] C. Louboutin and D. Meredith, "Using general-purpose compression algorithms for music analysis," *Journal of New Music Research*, vol. 45, no. 1, pp. 1–16, 2016, publisher: Routledge _eprint: https://doi.org/10.1080/09298215.2015.1133656. [Online]. Available: https://doi.org/10.1080/09298215.2015.1133656

[59] J. Urbano, J. Lloréns, J. Morato, and S. Sánchez-Cuadrado, "Melodic similarity through shape similarity," in *Exploring Music Contents*, S. Ystad, M. Aramaki, R. Kronland-Martinet, and K. Jensen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 338–355.

[60] B. Black, "The Bill Black Irish tune archive homepage." [Online]. Available: http://www.capeirish.com/webabc/

[61] N. Liberg, "EasyABC 1.3.7." [Online]. Available: https://sourceforge.net/projects/easyabc/files/

[62] M. S. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, Utrecht, Netherlands, Aug. 2010, pp. 637–642.

[63] D. Parsons, *The directory of tunes and musical themes*. Cambridge, UK: S. Brown, 1975.

[64] W. McKinney, "Data structures for statistical computing in Python," Austin, Texas, 2010, pp. 56–61. [Online]. Available: https://conference.scipy.org/proceedings/scipy2010/mckinney.html

[65] M. Tourish, "In process and practice: the development of an archive of explicit stylistic data for Irish traditional instrumental music," phdthesis, TU Dublin, Dublin, Ireland, 2013.

[66] C. Gao, J. Wang, Y. He, and L. Zhou, "Efficient mining of frequent sequence generators," in *Proceeding of the 17th international conference on World Wide Web - WWW '08*. Beijing, China: ACM Press, 2008, p. 1051. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1367497.1367651

[67] S. Triglia, "Elegant n-gram generation in Python," 2013. [Online]. Available: http://www.locallyoptimal.com/blog/2013/01/20/elegant-n-gram-generation-in-python/

[68] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, pp. 11–21, 1972.

[69] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM Journal of Research and Development*, vol. 1, no. 4, pp. 309–317, Oct. 1957. [Online]. Available: http://ieeexplore.ieee.org/document/5392697/

[70] F. J. Damerau, "A Technique for Computer Detection and Correction of Spelling Errors," *Communications of the ACM*, vol. 7, no. 3, pp. 171–176, Mar. 1964, place: New York, NY, USA Publisher: Association for Computing Machinery. [Online]. Available: https://doi.org/10.1145/363958.363994

[71] R. Grigoroiu, "fastDamerauLevenshtein," 2020. [Online]. Available: https://pypi.org/project/fastDamerauLevenshtein/

[72] C. Krumhansl, *Cognitive foundations of musical pitch*, ser. Oxford Psychology Series. New York, USA: Oxford University Press, 1990, no. 17.

[73] C. Sapp, "Key-profile comparisons in key-finding by correlation," Sapporo, Japan, 2008.

[74] B. Aarden, "Dynamic Melodic Expectancy," phdthesis, Ohio State University, Ohio, USA, 2003.

[75] H. Bellmann, “About the Determination of Key of a Musical Excerpt,” in *Proceedings of Computer Music Modeling and Retrieval (CMMR)*, Pisa, Italy, Sep. 2005, pp. 76–91.

[76] D. Temperley, *Music and probability*. Cambridge, USA: MIT Press, 2007.

[77] P. Van Kranenburg, M. De Bruin, and A. Volk, “Documenting a song culture: the dutch song database as a resource for musicological research,” *International Journal on Digital Libraries*, vol. 20, no. 1, pp. 13–23, 2019.

[78] P. Van Kranenburg, M. De Bruin, L. P. Grijp, and F. Wiering, “The meertens tune collections,” Meertens Institute, Amsterdam, Meertens Online Reports 2014-1, 2014.

[79] A. Volk and P. Van Kranenburg, “Melodic similarity among folk songs: An annotation study on similarity-based categorization in music,” *Musicae Scientiae*, vol. 16, no. 3, pp. 317–339, 2012.

[80] P. Van Kranenburg, A. Volk, and F. Wiering, “On identifying folk song melodies employing recurring motifs,” in *Proceedings of the 12th International Conference on Music Perception and Cognition*, E. Cambouropoulos, C. Tsougras, P. Mavromatis, and K. Pastiadis, Eds., 2012, pp. 1057–1062.

[81] D. Conklin, “Discovery of distinctive patterns in music,” *Intelligent Data Analysis*, vol. 14, pp. 547–554, 2010.

[82] P. Van Kranenburg and D. Conklin, “A pattern mining approach to study a collection of dutch folk-songs,” in *Proceedings of the 6th International Workshop on Folk Music Analysis*, P. Beauguitte, B. Duggan, and J. Kelleher, Eds., Dublin, 2016, pp. 71–73.

[83] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2008, online version at *http://informationretrieval.org/*.

[84] H. Samet, *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006.

[85] W. Zhou, H. Li, and Q. Tian, “Recent advance in content-based image retrieval: A literature survey,” *arXiv preprint arXiv:1706.06064*, 2017.

[86] W. Chen, Y. Liu, W. Wang, E. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew, “Deep image retrieval: A survey,” 2021.

[87] J. Rothstein, *MIDI: A Comprehensive Introduction*, ser. Computer Music and Digital Audio Series. A-R Editions, 1995. [Online]. Available: https://books.google.fr/books?id=ajDaXh-qgDUC

[88] M. Good, *”MusicXML for Notation and Analysis”*. W. B. Hewlett and E. Selfridge-Field, MIT Press, 2001, pp. 113–124.

[89] P. Rolland, “The Music Encoding Initiative (MEI),” in *Proceedings of the International Conference on Musical Applications Using XML*, 2002, pp. 55–59.

[90] “Music Encoding Initiative,” Music Encoding Initiative, 2015, accessed September 2021.

[91] F. Foscarin, F. Jacquemard, P. Rigaux, and M. Sakai, “A Parse-based Framework for Coupled Rhythm Quantization and Score Structuring,” in *MCM 2019 - Mathematics and Computation in Music*, ser. Proceedings of the Seventh International Conference

on Mathematics and Computation in Music (MCM 2019), vol. Lecture Notes in Computer Science. Madrid, Spain: Springer, Jun 2019. [Online]. Available: https://hal.inria.fr/hal-01988990

[92] F. Foscarin, R. Fournier-S'niehotta, and F. Jacquemard, "A diff procedure for music score files," in *Proceedings of the 6th International Conference on Digital Libraries for Musicology (DLfM)*. The Hague, Netherlands: ACM, 2019, pp. 7–11. [Online]. Available: https://hal.inria.fr/hal-02267454

[93] C. Antila, J. Treviño, and G. Weaver, "A hierarchic diff algorithm for collaborative music document editing," in *Third International Conference on Technologies for Music Notation and Representation (TENOR)*, 2017.

[94] M. S. Cuthbert and C. Ariza, "Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2010, pp. 637–642.

[95] J. S. Downie, "Evaluating a simple approach to music information retrieval: Conceiving melodic n-grams as text," Ph.D. dissertation, Univ. Western Ontario, 1999.

[96] G. Neve and N. Orio, "Indexing and Retrieval of Music Documents through Pattern Analysis and Data Fusion Techniques," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2004, pp. 216–223.

[97] S. Doraisamy and S. M. Rüger, "A Polyphonic Music Retrieval System Using N-Grams," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2004, pp. 204–209.

[98] C.-W. Chang and H. C. Jiau, "An Efficient Numeric Indexing Technique for Music Retrieval System," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2006, pp. 1981–1984.

[99] E. Cambouropoulos, M. Crochemore, C. S. Iliopoulos, M. Mohamed, and M.-F. Sagot, "A Pattern Extraction Algorithm for Abstract Melodic Representations that Allow Partial Overlapping of Intervallic Categories," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2005, pp. 167–174.

[100] J. Beltran, X. Liu, N. Mohanchandra, and G. Toussaint, "Measuring Musical Rhythm Similarity: Statistical Features versus Transformation Methods," *Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, pp. 1–23, 2015.

[101] C. Isikhan and G. Ozcan, "A survey of melody extraction techniques for music information retrieval," in *Proceedings of 4th Conference on Interdisciplinary Musicology (SIM'08), Thessaloniki, Greece*, 2008.

[102] V. Viro, "Peachnote: Music score search and analysis platform." in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2011, pp. 359–362.

[103] M. Gover and I. Fujinaga, "A notation-based query language for searching in symbolic music," in *International Conference on Digital Libraries for Musicology*, ser. DLfM '19. New York, NY, USA: ACM, 2019, p. 79–83. [Online]. Available: https://doi.org/10.1145/3358664.3358667

[104] W. B. De Haas, R. C. Veltkamp, and F. Wiering, "Tonal pitch step distance: a similarity measure for chord progressions." in *ISMIR*, 2008, pp. 51–56.

[105] P. Hanna, M. Robine, and T. Rocher, "An alignment based system for chord sequence retrieval," in *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, 2009, pp. 101–104.

[106] F. Lerdahl *et al.*, *Tonal pitch space*. Oxford University Press, USA, 2001.

[107] W. B. De Haas, M. Robine, P. Hanna, R. C. Veltkamp, and F. Wiering, "Comparing approaches to the similarity of musical chord sequences," in *International Symposium on Computer Music Modeling and Retrieval*. Springer, 2010, pp. 242–258.

[108] T. F. Smith, M. S. Waterman *et al.*, "Identification of common molecular subsequences," *Journal of molecular biology*, vol. 147, no. 1, pp. 195–197, 1981.

[109] W. B. De Haas, M. Rohrmeier, R. C. Veltkamp, and F. Wiering, "Modeling harmonic similarity using a generative grammar of tonal harmony," in *Proceedings of the Tenth International Conference on Music Information Retrieval (ISMIR)*, 2009.

[110] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler, "Omras2 metadata project 2009," in *12th International Society for Music Information Retrieval Conference*, ser. ISMIR, 2009.

[111] V. Eremenko, E. Demirel, B. Bozkurt, and X. Serra, "Jaah: Audio-aligned jazz harmony dataset," Jun. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1290737

[112] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. G. Grohganz, "Schubert winterreise dataset: A multimodal scenario for music analysis," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 14, no. 2, pp. 1–18, 2021.

[113] V. Eremenko, E. Demirel, B. Bozkurt, and X. Serra, "Audio-aligned jazz harmony dataset for automatic chord transcription and corpus-based research," in *ISMIR*, 2018.

[114] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. G. Grohganz, "Schubert winterreise dataset: A multimodal scenario for music analysis," *J. Comput. Cult. Herit.*, vol. 14, no. 2, May 2021. [Online]. Available: https://doi.org/10.1145/3429743

[115] E. Humphrey, J. Salamon, O. Nieto, J. Forsyth, R. Bittner, and J. Bello, "Jams: A json annotated music specification for reproducible mir research," in *Proceedings - 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 10 2014.

[116] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.