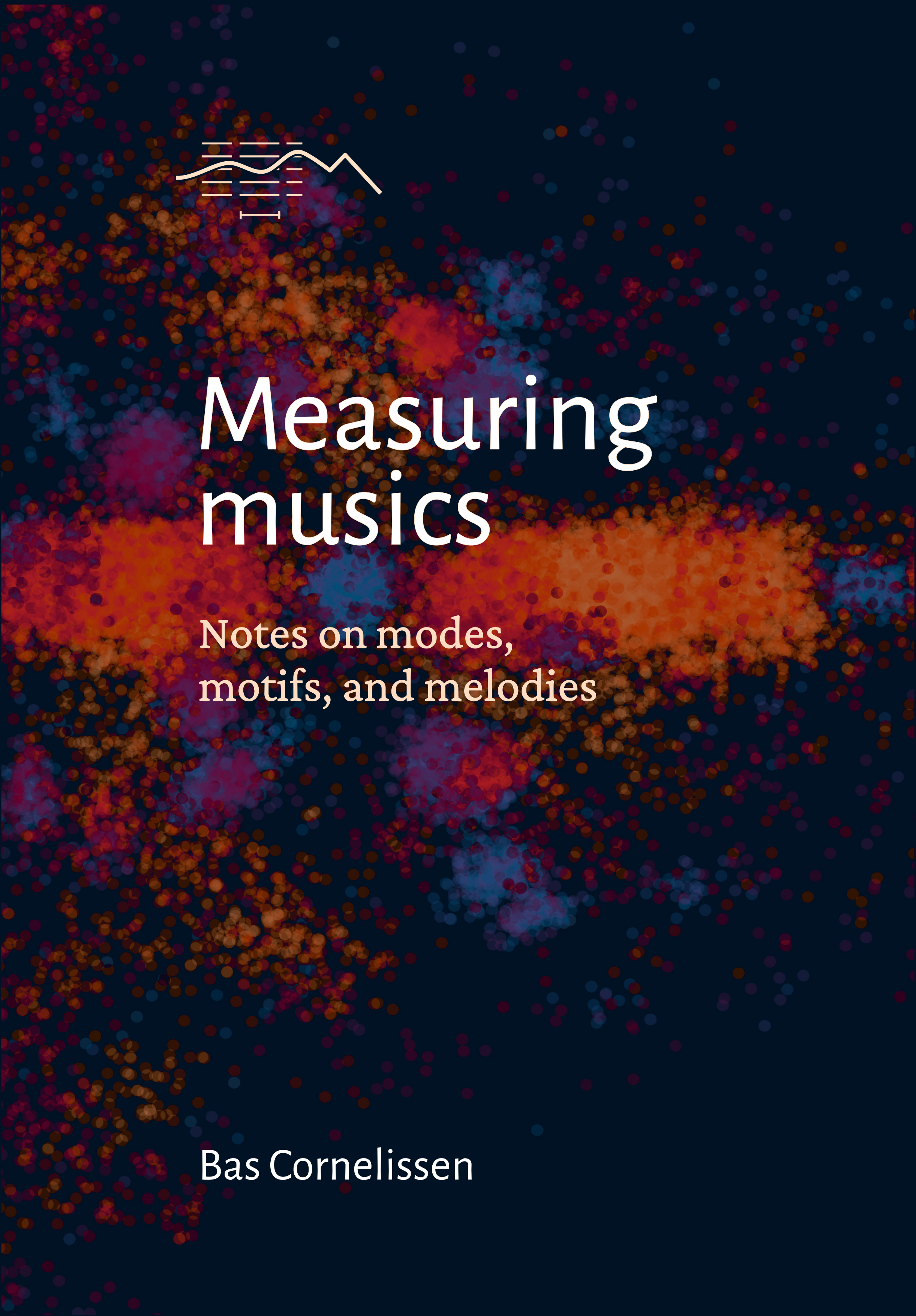


Measuring musics

Notes on modes, motifs, and melodies

Bas Cornelissen

Measuring musics

Notes on modes, motifs, and melodies

Bas Cornelissen

ILLC Dissertation Series DS-2024-02

Institute for Logic, Language and Computation

For further information about illc publications, please contact the Institute for Logic, Language and Computation at the University of Amsterdam:
email illc@uva.nl homepage illc.uva.nl phone +31 20 525 6051
address Science Park 107, 1098 xg, Amsterdam, The Netherlands

cover and typesetting by Bas Cornelissen/Studio Metrum. This dissertation is typeset in Crimson Pro (designed by Jacques Le Bailly/Fonthausen) and Alegreya Sans (designed by Juan Pablo del Peral/Huerta Tipográfica), both licensed under the sil Open Font License.

printed by Ipskamp Printing on recycled, fsc-certified paper.
isbn 978-94-6473-363-1
version 2.7

Measuring musics

Notes on modes, motifs, and melodies

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in de Aula der Universiteit
op vrijdag 23 februari 2024, te 11.00 uur

door

Bastiaan Jan Maarten Cornelissen

geboren te Utrecht

Promotiecommissie

PROMOTOR

dr. W.H. Zuidema	Universiteit van Amsterdam

COPROMOTORES

dr. J.A. Burgoyne	Universiteit van Amsterdam
prof. dr. H.J. Honing	Universiteit van Amsterdam

OVERIGE LEDEN

dr. F. Wiering	Utrecht University
prof. dr. I. Quinn	Yale University
prof. dr. J.J.E. Kursell	Universiteit van Amsterdam
dr. B.D. ten Cate	Universiteit van Amsterdam
dr. I.A. Titov	Universiteit van Amsterdam
prof. dr. K. Sima'an	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

For the fun of it

Contents

CHAPTER 1

Prelude

Prelude

We are an extraordinary species. In just about any climate, from the freezing colds of the Arctic to the dazzling heat of the Sahara, human populations thrive. On virtually every stretch of land you could visit, you would find more of us: humans. Far from a colorless, uniform mass, it is a spectacular display of cultural diversity. You can find people with hundreds of gods and those with no gods; people living in skyscrapers and those who carry their homes; those who live in mountains, and those who fare seas. But no matter where you go, you will find that they have language and music.

Music and language exemplify the richness and diversity of human life. There are over six thousand languages in the world, and orders more must have existed since our ancestors started to talk. There are languages without sounds and languages with dozens of them; languages that pack into one word, which would require several sentences in another. Some use east and west instead of left and right; in others, verbs do not live in the past, present, or future. There are languages with an endless inventory of number terms and those without any. But all these languages have at least one thing in common: children pick them up spontaneously and seemingly without effort. The case for music is not that different. You can find music without melody and music without a beat. Music with dozens of notes in an octave or just two. Music with conflicting meters and music without meters. What sounds pleasant in one type of music could be awfully dissonant in another. But while most of us can move to a beat or hum a melody, your dog can't—and that is certainly no lack of exposure.

Language and music are not the answer to what it means to be human, but they are an important part of it. Why so? Why did humans evolve to make music? And how so? I will not answer such 'big questions' in this dissertation: they are the questions that have motivated some of the smaller questions that I *will* address. The idea, in short, is this. If you want to understand the evolution of music, it helps to study what sorts of music exist, how musical traditions—or *musics*—relate, and how they compare. Are there properties that all musical traditions share, or hardly any share? To answer such questions, you have to start *measuring musics*: manually, perhaps, or automatically, using computational methods. And that is the central topic of this dissertation: developing computational methods to measure musics, from the modality of chants or shapes of melodies to inventories of rhythmic motifs and even an intricate rarity.

1.1 Music and musicality

But first—music? I imagine that a biologist would characterize music as a behavior. Comparable perhaps to how some musicologists prefer to see music as an activity, not an object, and refer to it with the verb *musicking* (Small, 1998). Musical behavior can take many forms: singing, dancing, playing an instrument, listening, or perhaps just silently studying sheet music or preparing a performance.

Much of our musical behavior is learned socially from other individuals and shared by a group. In biology, such behavior is known as *cultural behavior*, in contrast to for example instinctive behavior (e.g., Hoppitt & Laland, 2013). It is typical for much of human behavior but can also be found in other species, from whales and dolphins (Whitehead & Rendell, 2015) to perhaps even bees (Alem et al., 2016; Loukola et al., 2017). Cultural behavior results in a *dual inheritance*: individuals inherit not only their genetic makeup (biological inheritance) but also some of their behaviors (cultural inheritance). When thinking about cultural behavior, it may be helpful to distinguish the cultural phenomenon from the biological abilities that underly it. This distinction is commonly made for language, perhaps one of the better examples of a "cultural system that runs on biological infrastructure" (Levinson & Dediu, 2013).

As with languages, accumulating evidence suggests that *musics* build on a biological infrastructure known as *musicality* (Honing, 2018). Musicality does not refer to some special ability only gifted musicians have. It refers to the common, widely shared abilities that allow humans to engage in musical behavior, whether playing (production) or just listening (perception). The abilities are thought to be so common that people lacking them have become of scientific interest: the tone-deaf or those unable to hear a beat. Honing (2018) defines musicality as "a natural, spontaneously developing set of traits based on and constrained by our cognitive and biological system". This definition adopts a *multi-component* perspective: it suggests thinking of musicality as composed of multiple components or traits, such as beat perception, relative pitch perception, or vocal learning. A prominent research agenda in the field now aims to determine which components underly human musical behavior or, differently put, to characterize the *musicality phenotype* (Honing, 2018).

Turning to “the musical systems of cultures” (Nettl, 2005), what precisely are *musics*? In terms of cultural behavior, a music would be something like the totality of musical behavior shared by a cultural group. Its rich cross-cultural diversity makes it notoriously hard to pinpoint what counts as musical behavior—not to mention the many efforts to stretch its boundaries. A pragmatic escape assumes that musical behavior can be reliably recognized by members of a cultural group or trained ethnomusicologists. To further characterize musics, one could adopt *a typological perspective on musics*, in analogy with a multi-component perspective on musicality. Just like a multicomponent perspective breaks down musicality into different parts and studies those across species, a typological perspective breaks down musical behavior into a set of characters or features and examines their variability across musics.

This is by no means a novel agenda: it was a central concern of the discipline of *comparative musicology* that blossomed almost a century ago (Nettl, 2005; Savage, 2019), and resulted in typologies that are still used today, such as the Hornbostel–Sachs instrument classification. After the Second World War, the field adopted the new name of ‘ethnomusicology’ and moved focus to in-depth fieldwork and culture-specific description. In the words of Bruno Nettl (2005), “we study each music in its own terms, and we try to learn to see it as its own society understands it” (Nettl, 2005, p. 13). Comparison still had an important role to play, but a more relativistic one. Interest in cross-cultural comparison has recently resurfaced and even led to an attempt to revive comparative musicology (see e.g., Savage & Brown, 2013). This new comparative musicology also takes a typological perspective of musics, and is heavily invested in classifying, and comparing musical traditions. But as it aims for global comparisons, it is bound to understand musics not on their own terms, but in *general terms*: it develops concepts that are applicable cross-culturally.

In this dissertation, we focus on a few musical features and aim to characterize these computationally: primarily mode and contour, but also rhythmic and melodic motifs. We will focus almost exclusively on musical scores, which means that the term ‘music’ will be used in a narrow sense: that specific *product* of musical behavior that can be captured in a musical score. And as any musician used to notated music knows, that is fairly limited. It strips the rich behavior of much of its context and meaning. However, while musical behavior may be much more than a formal structure, it still *has* formal structures, and we will focus on those in this dissertation.

1.2 Outline

The form of this dissertation is somewhat unusual. Indeed, I prefer to call this work a ‘dissertation’ and not a ‘thesis’ as I do not put forward one central thesis for which each of the chapters provides arguments. Instead, the core of this dissertation consists of a series of interconnected articles.

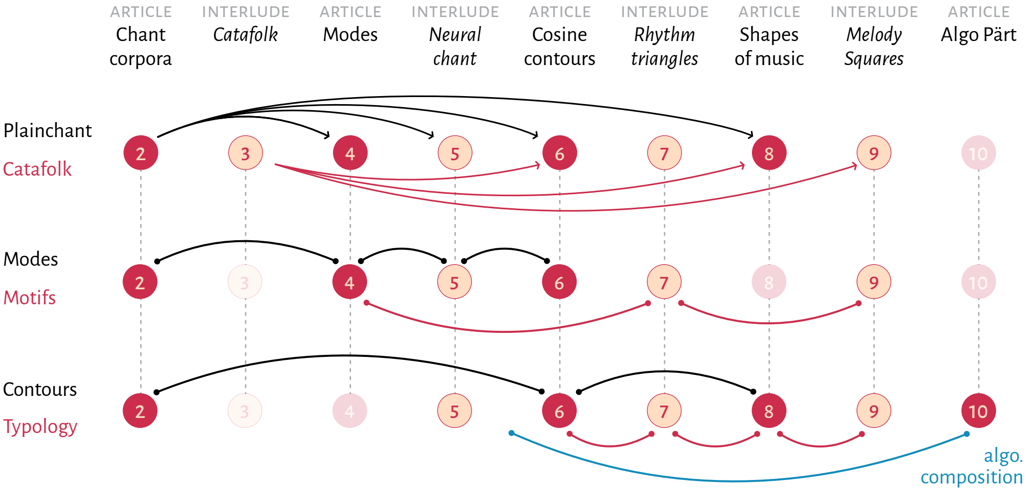


FIGURE 1.1 – Outline. This dissertation develops a series of computational methods to 'measure musics'. It is organized as a series of articles interleaved with interludes. Central themes as melodic modes, motifs, contours, and typology, appear in multiple chapters, often relying on the chant and folk music corpora introduced in the first two chapters. The figure highlights some thematic connections between the chapters.

I have chosen to interleave those articles with interludes, as illustrated in Figure 1.1. The articles, evenly numbered, can be quite formal, while the odd interludes are written more freely. Although one could think of the interludes as *divertimenti*, they are more than academic amusement: the interludes describe the projects that have not yet fully matured but still deserve a place in this dissertation.

CHAPTER 2 The first two chapters lay out the groundwork by introducing the main corpora studied in this dissertation. Chapter 2 presents two plainchant corpora, the Cantus Corpus and the GregoBase Corpus, along with a Python package that parses plainchant formats. The corpora and software are illustrated in two small case studies. One of the case studies confirms the *melodic arch hypothesis* in plainchant: phrases from this repertoire indeed tend to be arch-shaped, as the hypothesis suggests. The case study also paves the way for more in-depth studies of melodic contour representations in chapter 6 and chapter 8.

CHAPTER 3: INTERLUDE This interlude discusses the Catafolk project that aims to collect consistent metadata from folk song corpora, allowing one to bundle many corpora into one larger cross-cultural corpus. The project is a proof of concept and primarily contains subsets of the Essen Folksong Collection and the Densmore Collection, which will be used in later chapters.

CHAPTER 4 This chapter attempts to measure the central organizational structure of plainchant: the eight modes. Modes are melody *types* that lie somewhere between abstract scales and concrete melodies. We compare three different ways to classify musical mode: two approaches that largely view mode as a scale and one distributional approach that focuses on its melodic character. We find that this latter approach can still determine mode fairly accurately even when all pitch information has been discarded. However, this only really works when the mode is segmented in the 'right' way: in units corresponding to textual units such as syllables and words. We also propose a simple attribution method that visually explains why a chant may have been classified to a particular mode. All in all, the chapter confirms that mode is a melodic phenomenon, but it also suggests that this repertoire is built up from small melodic units, comparable perhaps to how a sentence is composed of syllables and words.

CHAPTER 5: INTERLUDE Following the linguistic analogy, this interlude takes on plainchant using a neural language model, partly because such a model would also be capable of generating artificial chant. But the interlude also tries to understand what kind of representations the model learns. Although preliminary, the learned chant representations suggest that mode and genre are the two primary axes along which chants are organized.

CHAPTER 6 The next chapter takes a more general perspective on the axes along which melodies and their shapes are best described. We analyze the principal components of melodies, represented as fixed-length pitch sequences and find that the principal components closely approximate cosine functions of increasing frequency. After explaining why the variance in melodies may be best explained by cosines, we propose a new contour representation that we call *cosine contours*. We illustrate the representation in three small case studies.

CHAPTER 7: INTERLUDE Cosine contours can be seen as a form of *continuous* music typology: they describe the musical feature 'contour shape' in a continuous fashion. The next three chapters continue this line of thought. First, we look at rhythmic motif frequencies. We visualize rhythmic data from music and animal vocalizations by plotting all motifs of three successive temporal intervals in a so-called rhythm triangle. Thinking in terms of motifs leads us to a measure of *isochronicity*—how steady, pulse-like a rhythm is—that generalizes the nPVI, a more commonly used measure. Our measure of isochronicity produces a cross-section of the rhythmic variability in music and animal vocalizations. Throughout the interlude, we discuss a question that has attracted attention recently: are rhythms in a given dataset *categorical*? This effectively questions the presence of statistical modes in some continuous space: the rhythm triangle, in this case.

CHAPTER 8 The next chapter also investigates the presence of statistical modes, but now in the space of melodic contours. What sort of typology one can best use to describe the distribution of melodic phrase contours? Rephrasing this as a clustering problem, we propose a way to measure the presence of statistical modes—but find none. This suggests that melodic phrase contours do not cluster into separate types, rendering any discrete typology somewhat arbitrary. This, combined with shortcomings in commonly used discrete typologies, suggest that one should instead view melodic contour as a continuous phenomenon.

CHAPTER 9: INTERLUDE After measuring rhythmic motifs in chapter 7, this interlude measures *melodic* motifs. It visualizes *melodic* units of three successive notes (or two intervals) in what might be called a *melody square*. Though simple, such squares are informative enough to group corpora by their rough area of origin. More importantly, the melody squares readily suggest common melodic patterns as well as rare ones. The music of Arvo Pärt may be a musical *rarum*, as it hides melody squares with a surprising symmetry.

CHAPTER 10 The final chapter is a case study of a single piece, *Summa* by Arvo Pärt. So far this dissertation developed formal ways to measure 'informal' music, but formal methods may well be indispensable to the study of certain *formal* music. To show why, we attempt to reconstruct *Summa* using formal procedures: an algorithmic reconstruction. This finale also closes the circle: Pärt's *tintinnabuli* style takes inspiration from the plainchant with which we will soon start this dissertation.

Several of these chapters are directly based on published articles, while others have not been presented elsewhere before. At the end of every chapter, I have included references to the relevant publications, pointers to data and code, and also listed author contributions.

CHAPTER 2

Article

Chant corpora

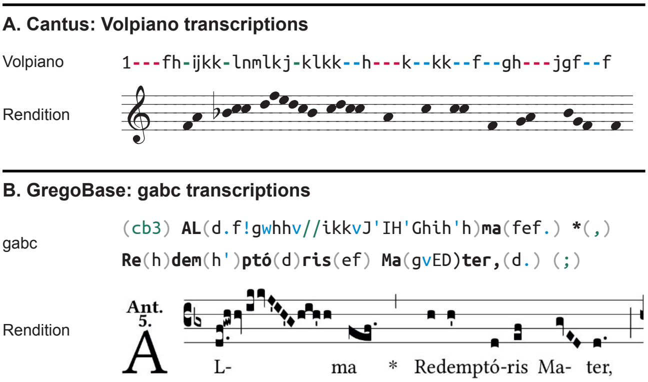
This chapter presents chant21, a Python package to support the plainchant formats gabc and Volpiano in music21, together with two large corpora of plainchant. The Cantus-Corpus contains over 60,000 medieval melodies collected from the Cantus database, encoded in the Volpiano typeface. The GregoBaseCorpus contains over 9,000 transcriptions from more recent chant books in the gabc format. Chant21 converts both formats to music21 while retaining the textual structure of the chant: its division into sections, words, syllables, and neumes. We present two case studies. First, we report evidence for the melodic arch hypothesis from the GregoBase Corpus. Second, we analyze connections between differentiæ and antiphon openings in the Cantus Corpus and show that the systematicity of the connection can be quantified using an entropy-based measure.

2.1 Introduction

If one thing stands out about our species' musical behavior, it is its ubiquity: all cultures seem to make music (Mehr et al., 2019). Yet, our understanding of music from corpus studies is almost entirely based on Western classical or popular music (Savage, 2022). Part of the explanation might be the scarcity of large corpora from other traditions. Recent efforts have been addressing this, often under the header of *computational ethnomusicology* (Tzanetakis et al., 2007). We contribute to the efforts to diversify by converting two existing databases of Christian plainchant into a form suitable for corpus analysis in popular tools: the medieval Cantus Corpus and the more recent GregoBase Corpus. We also release the Python package Chant21 for working with these corpora in music21. Finally, we present two case studies illustrating their usefulness. First, we show that melodic phrases have arch-shaped contours in the GregoBase Corpus, confirming the general *melodic arch hypothesis* (Huron, 1996). Second, we focus on a particular problem in chant scholarship and revisit the relation between so-called differentiæ and antiphon openings (Shaw, 2018) in the Cantus Corpus.

The plainchant on which we focus is, indeed, another European tradition. But it is sufficiently distant from Western classical and popular music, if not in time, then certainly in its musical language, to be studied as a separate tradition (Jeffery, 1992). The music goes back well over a thousand years, to the ninth century, when the first melodies appear in manuscripts. Multiple chant traditions had coexisted in Europe before then, with their own variants of music and texts, but many were (partly deliberately) displaced by what became known as Gregorian chant. The monophonic melodies are rooted in the recitation of sacred Latin texts, which formed the backbone of the liturgy. The first manuscripts therefore only record the text, but later sketches of the melodies appear between the lines of the text. These sketches consisted of so-called *neumes*, figures indicating the contour of small melodic motifs but not their exact pitches. Later, these neumes were placed on staff lines to also indicate their exact pitches. This developed into both the modern five-line notation and the four-line *square notation* used in chant books today. The corpora we present employ both types of notation (Figure 2.1).

The chant repertoire was, sometimes actively, organized along several lines (Hiley, 2009). First of all, chants were classified into a system of eight *modes*, usually grouped in four pairs (Dorian, Phrygian, Lydian, Mixolydian). Two paired modes use the same final note but differ in their typical range: the so-called *authentic* one moves mostly above the final, and the *plagal* one around it. This already shows that modes are *melody types*, more than just the church scales to which they are sometimes associated (Powers et al., 2001). We discuss modes in more detail in chapter 4. Second, different parts of the liturgy use different chant *genres*, from the short, syllabic *antiphons* to the elaborate responsories. Some genres, like antiphons, consisted of freely composed melodies, but others, like psalms,



A. Cantus: Volpiano transcriptions
Volpiano
Rendition
B. GregoBase: gabc transcriptions
gabc
Rendition
Ant.
5.
A
L-
ma
*
Redemptó-ris
Ma-
ter,

```
1---fh-ijkk-lnmlkj-klkk--h---k--kk--f--gh---jgf--f
```

```
(cb3) AL(d.f!gwhhv//ikkvJ'IH'Ghih'h)ma(fef.) *(,)
Re(h)dem(h')ptó(d)ris(ef) Ma(gvED)ter,(d.) (;)
```

FIGURE 2.1 – **Volpiano and gabc.** Two versions of *Alma redemptoris mater*. **(A)** The Cantus Corpus contains melodic transcriptions from medieval manuscripts notated in Volpiano: a simple five-line notation. **(B)** The GregoBaseCorpus contains scores from recent chant books in the gabc format, an elaborate format for four-line square notation.

used standard melodic formulae: a reciting tone decorated by an opening and closing gesture particular to the mode of the chant.

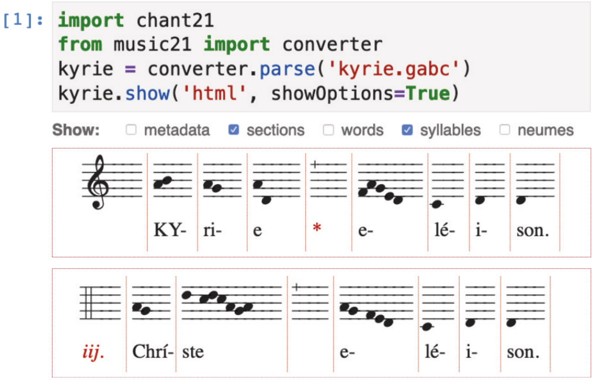
Most computational studies of plainchant have been concerned with optical music recognition of medieval manuscripts. But several recent studies have addressed more musicological questions, also in other chant traditions: Panteli and Purwins (2013) analyzed scale intonation in Byzantine chant, and Biró et al. (2012) studied cadences in Torah trope. Closer to the present work, van Kranenburg and Maessen (2017) used perplexities under an *n*-gram model to classify five early Christian chant traditions. We hope that the two corpora and software we will now present inspire more computational studies of plainchant.

2.2 Corpora

The first corpus we present, the Cantus Corpus, is in essence a cleaned-up export of the Cantus database (Lacoste et al., 1987–2019). This is an online index of the many medieval manuscripts kept in libraries across the world. As of this writing, it contains 497,071 chants; the database contains records for almost all, with information on where they are found in which manuscript, but also on things like their incipit, liturgical genre, feast, mode, and a *Cantus ID* to be able to identify the same chants across manuscripts and databases. For 63,628 chants (13%) the melody has also been (partially) transcribed using *Volpiano*.[1]

Volpiano is a typeface that renders text as notes on five staff lines and was specifically developed for notating plainchant. Several conventions are commonly adhered to, such as the use of three, two, and one hyphen(s) to indicate word, syllable, and neume boundaries respectively (Figure 2.1A). This allows the music to be aligned to the manuscript text, which is transcribed separately. Many of these conventions have been fixed in the elaborate transcription guidelines of the Cantus database, and this is what we refer to as the *(Cantus) Volpiano format*. The guidelines and editorial reviews ensure a high transcription quality (Helsen & Lacoste, 2011).

[1] Of the transcribed chants, 37% contain fewer than 30 notes and are probably incipits.



```
[1]: import chant21
     from music21 import converter
     kyrie = converter.parse('kyrie.gabc')
     kyrie.show('html', showOptions=True)
```

Show: □ metadata ☑ sections □ words ☑ syllables □ neumes

FIGURE 2.2 – Chant21 in action. Chant21 improves the support for plainchant in Music21 with converters for gabc and Volpiano. It uses a chant representation that divides the chant into sections, words, syllables, and neumes. This structure can be interactively explored in Jupyter notebooks.

The Cantus database is easy to use for chant scholars, but not necessarily for computational purposes: it is continuously updated, which is actually inconvenient when replication is a concern. We, therefore, scraped the database via its API and converted it to a set of clean CSV files which we release as the Cantus Corpus. Releases are versioned as we plan to occasionally release newer versions.

Our second corpus, GregoBase Corpus, again repackages and versions an existing database: GregoBase (Berten, 2013–2020), which provides a complementary perspective on chant. Whereas the Cantus database maps the complexity of medieval manuscripts in a simplified notation (Volpiano), GregoBase consists of modern reinterpretations of the Gregorian repertoire: the one found in chant books like the *Liber Usualis*. Such books are indented for practical use and use the full scope of square notation, including things like breathing marks, different note shapes, rhythmic signs, and clef changes.

The GregoBase website currently hosts 9139 chant transcriptions from 29 books, including the complete *Liber Usualis*. The transcriptions are written in *gabc* (Figure 2.1B), a plain text format for square chant notation, developed for the typesetting system *Gregorio*. We converted the GregoBase database to a set of easy-to-use CSV files, but also to separate gabc files that include metadata such as the mode, liturgical genre, and all books a chant appears in.

2.3 Chant21

To make it easier to work with the two corpora we present the Python package chant21 which improves the support for gabc and Volpiano in music21 (Cuthbert & Ariza, 2010), by now the go-to toolkit for symbolic computational musicology. Chant21 consists of parsers for (1) gabc and (2) Volpiano; (3) a way to align text to music notated in Volpiano; (4) a chant representation that retains the subdivision in sections, words, syllables, and neumes; (5) a way to export this representation to HTML, which allows for fast visualization in Jupyter notebooks.

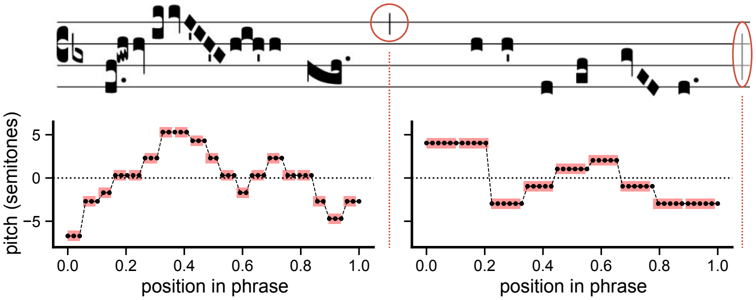


FIGURE 2.3 – Contour representation. Contours consist of 50 pitches, sampled after normalizing the phrase duration and transposing the phrase by its mean pitch. This is illustrated in the first two phrases of the antiphon *Alma mater redemptoris*. The plots below the score show the contours in black over a red piano roll.

Writing parsers for the elaborate gabc syntax and the informal Volpiano guidelines is not straightforward. After experimenting with custom parsers, we decided to specify the syntax of both formats as *parser expression grammars* (PEGS) (Ford, 2004).[2] Specifying the syntax in a grammar makes it transparent and much easier to maintain. PEGS resemble context-free grammars but use a deterministic choice operation to make parse trees unambiguous. After specifying the grammar, we delegate the actual parsing to the PEG parser *Arpeggio* (Dejanović et al., 2016). The resulting parsers are reliable: their error rates are well under 1% when evaluated on the Cantus Corpus and GregoBase Corpus and most failures are caused by syntax errors.

The parse trees of both gabc and Volpiano strings are then converted to music21 objects, but using a custom, hierarchical chant representation that groups the music into sections, words, syllables, and neumes. This structure can be useful in computational studies (as we will see in chapter 4) but is also needed to align Volpiano to the text. The Cantus database has guidelines for full-text transcriptions: how to for example mark section boundaries, or missing pitches. We use another PEG-based parser to parse the text and then split all words in syllables using the Latin syllabifier from the *Classical Language Toolkit* (Johnson et al., 2014–2021). After all this, the text is divided into sections, words, and syllables, which we match to their counterparts in the music.

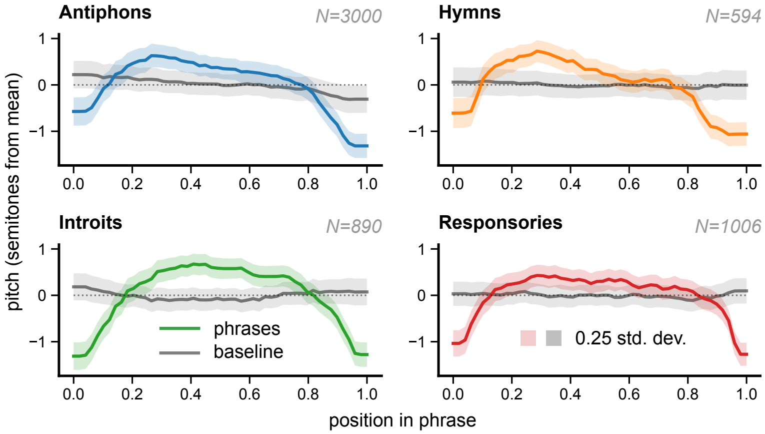
Finally, inspired by the Cantus website, chant21 can export the hierarchical chant representation to HTML, using Volpiano to display the music. This is particularly useful in Jupyter notebooks: it results in much faster typesetting and allows you to interactively explore the structure of the chant. After installing Volpiano and running `pip install chant21`, chant21 is ready to be used (Figure 2.2).

2.4 Case study 1: The melodic arch

To illustrate the usefulness of the presented corpora and software, we discuss two case studies. The first concerns the *melodic arch hypothesis*: the claim that the pitch contour of musical phrases across cultures tends to be arch-shaped. David Huron (1996) was the first to present quantitative support for this phenomenon, based on an analysis of 6000, mostly Ger-

[2] This idea was borrowed from `gabc-parser`, but we had to completely rewrite the grammar as `gabc-parser` only implements the basic features of gabc and left many chants unparseable.

FIGURE 2.4 – Average phrase contours. The melodic arch hypothesis seems to hold in Gregorian chant. Averaging all phrase contours results in arch-shaped contours (colored), whereas averaging random segments (grey) yields more or less flat contours. This is illustrated for four chant genres.



man folksongs from *Essen*. Later studies confirmed the hypothesis in the 2000 Chinese folksongs that were later added to *Essen* (Tierney et al., 2011), and a small global sample of 35 recordings from the Garland Encyclopedia (Savage et al., 2017). It has been suggested that the melodic arch is the result of general motor constraints (Tierney et al., 2011). Those make it easier to produce rising pitch contours at the start of a phrase when the pressure beneath the vocal folds is rising, and falling contours when the pressure drops towards the end. These constraints could imply a weak tendency for phrases to be arch-shaped (or descending) *on average*, even though individual phrases can take many shapes.

We analyze if these findings extend to Gregorian chant and focus on the *Liber Usualis* from the GregoBase Corpus (v0.4). We extracted phrases using the explicit breathing marks (*pausas*) in chant notation. As rhythmic interpretations of chant vary, we assigned all notes in chants equal duration. We removed duplicate phrases and phrases with fewer than 4 notes, and then randomly sample 3000 phrases per chant genre. Finally, we normalized all phrases to have duration 1 and mean pitch 0, and sampled 50 equally spaced pitches from the resulting contour (Savage et al., 2017; Tierney et al., 2011), as illustrated in Figure 2.3.

We average the 3000 normalized contours of a given genre and compare this to the following random baseline. We randomly segment every chant by successively sampling segment lengths from a Poisson distribution approximating the actual phrase lengths. The first and final (random) segments of each chant are omitted. This results in a set of random segments whose lengths are similar to actual phrases, but whose boundaries are unlikely to overlap with actual phrase boundaries. This keeps the melody intact and only shifts phrase boundaries—rather than shuffling all pitches (Savage et al., 2017).

Figure 2.4 shows the average phrase contours (coloured) compared to the average random segments (grey) for four chant genres. Whereas the actual phrases are clearly arch-shaped on average, the baseline is pretty much flat. The overall size of the arch is small (around 2 semitones), but similar to earlier findings (Savage et al., 2017; Tierney et al., 2011). The

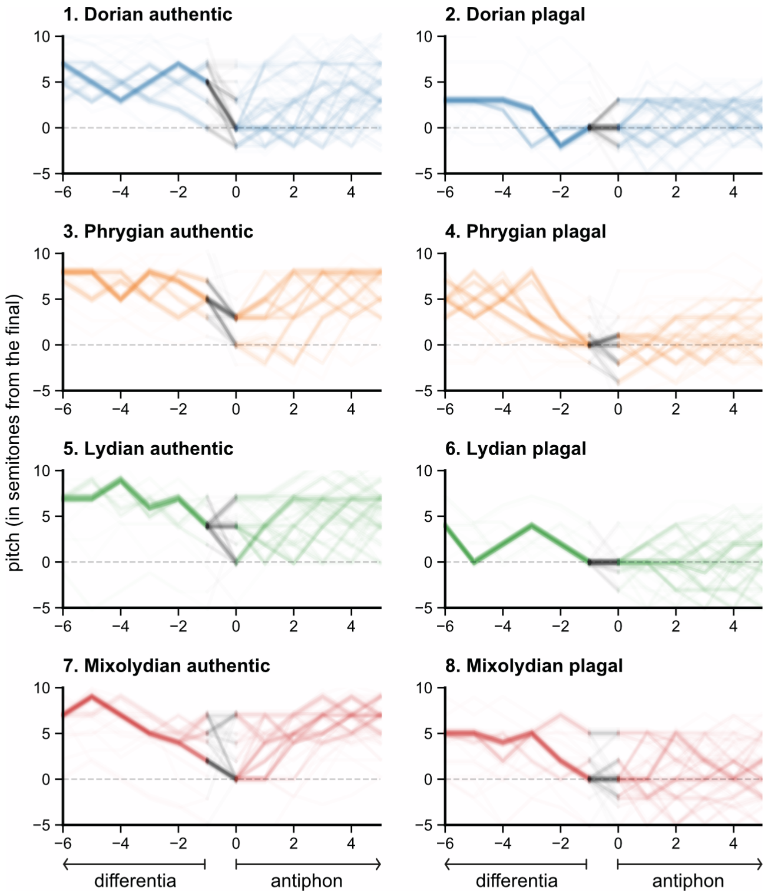


FIGURE 2.5 – Differentia-antiphon connections in all modes. Each line represents the last 6 notes of the differentia (colored), followed by the return to the antiphon (black), and 5 more notes of the antiphon (colored). We sample and show 200 connections per mode, jittered vertically to reveal clusters of overlapping contours.

average contours appear to differ across genres, but it requires further analyses to see if these differences are significant. The comparison with the random baseline does however make clear that phrase boundaries have a noticeable and consistent effect on the shape of phrase contours. In sum, these results from this corpus of plainchant are consistent with the melodic arch hypothesis.

2.5 Case study 2: Differentiæ

Our second case study revisits a particular problem in chant scholarship: the relation between so-called *differentiæ* and antiphon openings (Shaw, 2018). Every week, monks would sing a cycle of 150 psalms to melodic formulae known as psalm tones. An antiphon was sung before the psalm and repeated afterward. The *differentiæ* is the very end of the psalm, always set to the words *sæculorum amen* (abbreviated as *euouae*) and sung

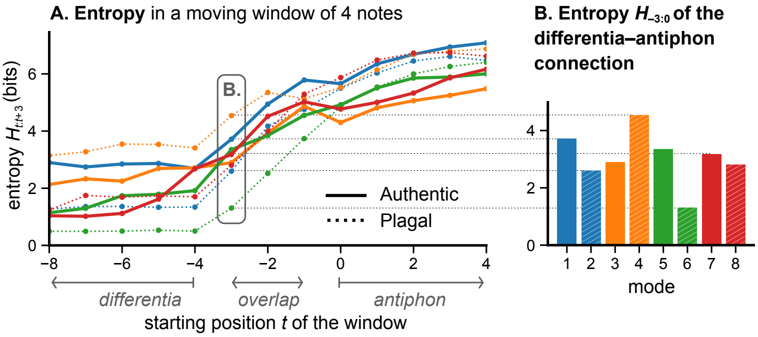


FIGURE 2.6 – Entropy of the chant. (A) We move a sliding window of 4 notes across the chant and estimate the unpredictability in the window using the entropy $H_{t:t+3}$ (details in the main text). This shows that differentiæ ($t \leq -4$) are more predictable than antiphons ($t \geq 0$). **(B)** Highlights the window containing the last 3 notes of the differentia and the first note of the antiphon, showing for example that the connection in mode 6 is more predictable than in mode 4.

directly before the repetition of the antiphon. The order, in short, was always antiphon–psalm–differentia–antiphon. A question dividing chant scholars is whether there is a systematic relation between differentiæ and antiphon openings: do certain psalm endings usually imply certain antiphon openings?

Shaw (2018) conducted the first large-scale data analysis and suggests that there is indeed a systematic connection for mode 1. Using chant21 we can extend this to all eight modes by visualizing the connections directly. We selected all 7102 antiphons from the Cantus Corpus (v0.1) that had a complete Volpiano transcription, lyrics ending on variants of *aeouae*, and a 'simple' mode (e.g., not transposed). We extract the last 6 pitches of the differentia and concatenate the first 6 notes of the antiphon to obtain the *(differentia–antiphon) connections*. We transpose all connections so that the final has pitch 0.

Figure 2.5 shows the connections for all modes. The systematicity seems to differ between modes. For example, mode 6 exhibits a very systematic connection: only one differentia is really ever used, and this virtually always leads to the same starting pitch of the antiphon (the final, F). Mode 5, on the other hand, also uses mostly one differentia, but this leads to three possible antiphon openings. This is certainly less systematic but still more predictable than a random transition.

We can quantify this difference in systematicity. For a given mode, consider all the segments $s_{-3:0} = (n_{-3}, n_{-2}, n_{-1}, n_0)$ spanning the last three notes of the differentia and the first of the antiphon. If $p(s_{-3:0})$ denotes the relative frequencies of all such segments, then we can measure the systematicity of a connection using its *entropy* $H(p(s_{-3:0}))$ or $H_{-3:0}$ for short. The entropy effectively measures the unpredictability of the chant in the segment from position -3 to position 0. A higher entropy indicates a less systematic connection, so we would expect mode 5, for example, to have higher entropy than mode 6. Figure 2.6B confirms this.

Next, we measure the entropy $H_{t:t+3}$ in a sliding window of four notes, starting at any position t and not only $t = -3$ as we did above. This allows us to analyze how unpredictable different parts of the chant are, which we do in Figure 2.6A. It is immediately clear that the more formulaic differentia ($t \leq -4$) are more predictable than antiphons ($t \geq 0$). But we also see that the moment we return to the antiphon, the entropy increases: $H_{-4:-1} < H_{-3:0}$. This suggests that across modes, differentia–antiphon connections are less predictable than differentiæ, but more predictable than antiphon openings.

2.6 Conclusions

This chapter presented two large corpora of Christian plainchant, the Python library chant21 which allows them to be used in music21, and two case studies. First, we showed that phrase contours in the GregoBase Corpus confirm the melodic arch hypothesis. Second, we show that the connection between differentiæ and antiphon openings is less predictable than the connection between notes within differentiæ, but more predictable than within antiphons. Moreover, the relation differs across modes. Both case studies only scratch the surface and raise further questions. We look again at differentiæ at the end of chapter 5, when classifying the mode of chants, and discuss melodic contour in much more detail chapter 6 and chapter 8. Those chapters broaden the perspective and look at melodic shapes in different musical traditions. But to do so, we need more music.

DATA AND CODE All data and code have been made available online:

- Cantus Corpus is available at github.com/bacor/cantuscorpus.
- GregoBase Corpus is available at github.com/bacor/gregobasecorpus.
- Chant21 can be found at github.com/bacor/chant21.
- For data and code related to the case studies, see github.com/bacor/DLfM2020

REFERENCE This chapter is based on Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2020b). Studying large plainchant corpora using Chant21. *7th International Conference on Digital Libraries for Musicology*, 40–44. DOI 10/ghmnrk

AUTHOR CONTRIBUTIONS BC designed and conducted the research and wrote the paper. JAB and WZ supervised the research and edited the manuscript.

ACKNOWLEDGEMENTS We would like to thank everyone involved in both the Cantus and GregoBase projects. In particular, we want to thank Jan Koláček for his help with downloading the full Cantus Database, and Olivier Berten for providing us with complete database exports of GregoBase.

CHAPTER

3

Interlude

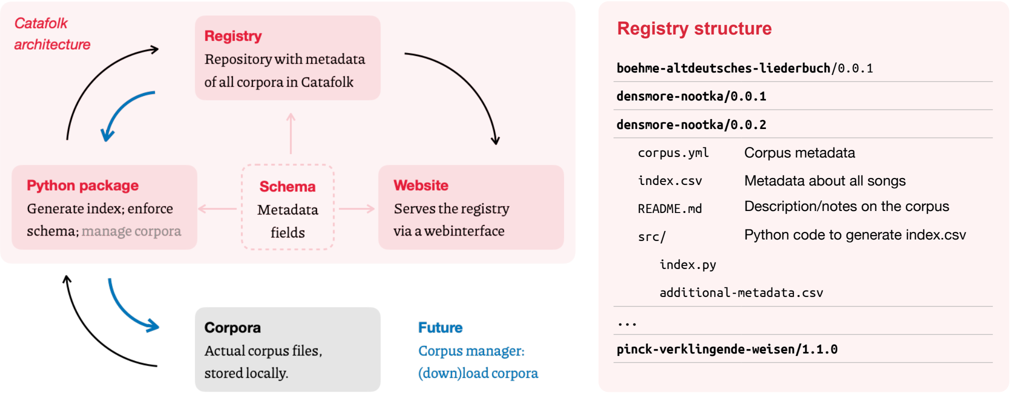
Catafolk

Cross-cultural computational studies often face an abundance of nothing: data scarcity. Two studies by Savage et al. (2015) and Mehr et al. (2019) are a case in point. Both rely on *global* samples to study the variety of music across the world. Although the studies differ in many respects, they have in common that they include at most a few songs from each tradition: Savage et al. (2015) used 304 recordings with a broad geographical spread from the *Garland Encyclopedia of World Music*, while Mehr et al. (2019) used only 118 songs covering a representative sample of societies.

Such samples may allow one to investigate musical diversity *across* cultures—the main objective of both studies—but not *within* cultures. A few songs can after all only accurately describe within-culture variability when one assumes that there is none: a problematic assumption that Savage and Brown (2013) have called the "one culture = one music" model. Describing diversity within a tradition usually requires a large *local sample* of its music. Such samples exist, but those widely available do not add up to a global sample—let alone a representative one.

Consider the *Essen Folksong Collection* (Schaffrath, 1995), perhaps the go-to corpus for cross-cultural musical scores. It contains large numbers of both German and Chinese folksongs that form an attractive contrast. The collection has accordingly been used in many studies, from cognitive to computational musicology, but—I suspect—usually not for principled, but *practical* reasons: other collections may not have been as readily available. In contrast, comparative research from the 50s and 60s, like the work of Mieczyslaw Kolinski (e.g., 1959, 1965a, 1965b), relied on a more varied range of musical traditions. The properties of interest were directly tabulated from the source publications, which meant a lot of manual work, but also a thorough understanding of the sample.

In that respect, the Essen Folksong Collection is not an ideal alternative. Where, for example, do the Chinese folksongs come from? Who collected them, where, and when? Are they all *songs*, or do they include instrumental


Catafolk architecture
Registry
Repository with metadata of all corpora in Catafolk
Python package
Generate index; enforce schema; manage corpora
Schema
Metadata fields
Website
Serves the registry via a webinterface
Corpora
Actual corpus files, stored locally.
Future
Corpus manager: (down)load corpora
Registry structure

```
boehme-altdeutsches-liederbuch/0.0.1
densmore-nootka/0.0.1
densmore-nootka/0.0.2
    corpus.yml            Corpus metadata
    index.csv             Metadata about all songs
    README.md             Description/notes on the corpus
    src/                  Python code to generate index.csv
        index.py
        additional-metadata.csv
...
pinck-verklingende-weisen/1.1.0
```

FIGURE 3.1 – Architecture of Catafolk. Catafolk consists of a *registry* holding the metadata of all corpora, a *Python package*, used amongst others to generate the registry, and a *website* to make the metadata easily accessible. At the core is a common schema, the set of metadata fields used by Catafolk.

melodies? Indeed, even the 'German' folksongs are rife with ambiguities, as Andrew Brinkman explains (Brinkman, 2020, 2021). There are mismatches between the songs in the collection and the songs in the source books, and it is unclear why the collection is divided into sections the way it is. More problematically, we do not know on what basis Schaffrath *added* information, such as a genre classification or phrase boundaries, neither of which appear in the source materials.

The relatively poor documentation of *Essen* contrasts sharply with the databases used in linguistic typology. Over the last decades, typologists have gathered vast amounts of research findings in projects such as the *World Atlas of Language Structures* (WALS) or the newly released Grambank.[1] Both datasets describe the grammatical structure of languages using a vast range of features. Notably, the absence or presence of every feature is accompanied by a reference to ensure the reliability of the data. Taking inspiration from linguists and hoping to address the issues raised above, I started to organize my research data more systematically. The project that grew out of that, Catafolk, is the topic of this interlude.

3.1 Catafolk

Catafolk aims to bridge the divide between sparse global samples and dense local samples by combining already available corpora. Since not all corpora may be freely shared, it instead focuses on providing metadata in a consistent format. The project is primarily a proof of concept that grew from how I organized my research data. As illustrated in Figure 3.1, Catafolk consists of three components bound by a common schema: a registry, a Python package, and a website.

COMPONENT 1: THE REGISTRY The *registry* is a Git repository containing metadata about all corpora and their songs. The most important part of the

[1] There are many more examples, see clld.org for an overview of cross-linguistic datasets.

registry is the *index* file, which lists the metadata for each entry in the corpus. The index is automatically generated by pulling data from multiple sources. Some metadata can be extracted from the source files (e.g., kern files), some data is constant for the corpus, and some may be included in additional metadata files. The code used to generate an index is also included in the registry. Every corpus is also versioned, and past metadata versions are kept in the registry.

COMPONENT 2: A PYTHON PACKAGE The second component of Catafolk, the *Python package*, assists in generating the index files. It ensures that the Catafolk schema (see below) is respected and maintains consistency in the registry. The package is currently used to maintain Catafolk but could, in the future, also be used to load metadata from the registry or organize locally installed corpora. Going further, one could turn Catafolk into the equivalent of a package manager: a *corpus manager* that would download both the corpora and their metadata while taking care of versioning, validating the integrity using checksums, and so on.

COMPONENT 3: THE WEBSITE The third component is the Catafolk *website*, which makes the registry available via a graphical interface. The website builds a knowledge graph from the index files, which allows one to query all corpora in Catafolk simultaneously.[2] But the website primarily aims to improve the accessibility and documentation of the corpora. For that reason, I have also included references to source publications and, where possible, linked individual songs to publicly available scans of the sources.

THE CATAFOLK SCHEMA Central to all these three components is the Catafolk *schema*: the list of metadata fields used by Catafolk. Catafolk's ontology currently contains 61 fields, spanning musical data such as title, key, tempo, or tune family to metadata on the collectors, encoders, or copyrights. More technical fields, such as file paths and md5 hashes, are also included so that the integrity of the corpus can be verified. Entries are geocoded as much as possible and linked to Glottolog, D-Place, eHRAF, and possibly to scans of the source publications. The fields in the schema are inspired by the metadata fields in the Kern Humdrum format, with various additions based on the Natural History of Song corpus.

3.2 Corpora

Catafolk is publicly available at bacor.github.io/catafolk. The project is in an early stage but already contains metadata for 15,507 songs from 22 datasets. The vast majority of those are symbolic transcriptions from KernScores, the Densmore collection (Shanahan & Shanahan, 2014), and the Finnish Folk Tunes collection (Eerola & Toiviainen, 2004). In particular, the following two collections will be used later in this dissertation.

[2] The website uses Gatsby, a static site generator, and React. Gatsby internally constructs a knowledge graph that can be queried using the query language GraphQL.

THE ESSEN FOLKSONG COLLECTION As discussed in Brinkman (2020) in more detail, the origins of the collection go back to 1982, when Helmuth Schaffrath started collecting folksongs in a format known as the *Essen Associative Code (EsAC)*. This resulted in the publication of 6,255 folksongs in 1995. After Schaffraths death in 1994, Ewa Dahlig-Turek coordinated the EsAC collection, to which much Polish and Chinese music has been added since. In 1995, David Huron converted the Essen Folksong Collection to his new **kern format, which is available via KernScores (kern.humdrum.org).

Parts of the Essen Folksong Collection have been included in Catafolk but as separate corpora, corresponding to the source publications. For example, Catafolk contains the following three large collections of German folksongs (from the `essen/europe/deutschl` directory in *Essen*):

- DEUTSCHER LIEDERHORT This is a collection of 1700 German folk songs, originally collected by Ludwig Erk and later edited by Franz Magnus Böhme (Erk & Böhme, 1893a, 1893b, 1894). This corpus corresponds to the `erk` directory in *Essen*.

- ALTDEUTSCHES LIEDERBUCH A collection of 309 folk songs collected by Franz Magnus Böhme (1877). This corpus corresponds to `altdeu1` and `altdeu2` directories.

- VOLKSTHÜMLICHE LIEDER DER DEUTSCHEN A collection of 704 German folk songs published by Franz (Böhme, 1895). This corpus corresponds to `boehme` directory.

THE DENSMORE COLLECTION Frances Densmore was a very prolific collector of Native American music. Employed by the Bureau for American Ethnology from 1907 onwards, she embarked on many field trips, making thousands of recordings from all over the United States (Neubarth et al., 2018; Shanahan & Shanahan, 2014). Many of these have been transcribed and published in her books (Densmore, 1910, 1913, 1918, 1922, 1929a, 1929b, 1932, 1939, 1943, 1957, 1958). After Paul von Hippel, David Huron, and Craig Sapp transcribed some of these books in Humdrum, Shanahan and Shanahan (2014) transcribed all remaining books and made these available as the *Densmore Collection*. Besides recording music, Densmore also drew up extensive tables listing the frequency of various musical features (e.g., scale, tempo, or mode). She used these to compare the music of various peoples. I have transcribed some additional metadata from Densmore's tables and added it to Catafolk.

There are ethical concerns when using recordings of this kind. As Shanahan and Shanahan (2014) point out, Densmore "lacked formal training as an anthropologist, and her attitude toward her subjects in the early part of her career is often described as condescending and patronizing." Some music may have been intended for particular occasions, not for broader display. Throughout this dissertation, I refer to Native American peoples using the names and spelling used nowadays rather than the names Densmore used.

Although Catafolk has proven helpful for the present dissertation, it is somewhat unsatisfactory that the usual suspects currently still make up for the largest part of it. Many collections are missing, even very obvious ones, such as the Meertens Tune Collections or the two chant corpora introduced in chapter 2. Curating a catalog like Catafolk turns out to be a lot of work that, in the end, requires a much larger scale. But I am convinced that the effort is worthwhile and hope that Catafolk will be an inspiration to further map the musical treasures out there.

We now return to one such treasure: plainchant. The question that motivated the study in the next chapter was inspired by folksong research. Songs are perhaps the obvious 'units' of cultural transmission in music. Indeed, one of the central notions in folksong research is that of a *tune family*: a group of closely related songs that are the product of a process of cultural change. But one may wonder whether there are smaller units, perhaps analogous to how phrases, words, and morphemes in language are every smaller replicators? What are, in other words, the smallest units, larger than notes, in a musical tradition?

DATA AND CODE Most of the code is publicly available:

- The Catafolk website can be found at bacor.github.io/catafolk.
- The source code of the Catafolk website is available on github.com/bacor/catafolk
- The registry can be found at github.com/bacor/catafolk-registry
- The Python package has not been released yet.

REFERENCE Catafolk was presented at SysMus21: Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2021b). Catafolk: Cataloguing folk music datasets for comparative musicology. In J. Stupacher & S. Hagner (Eds.), *Proceedings of the 14th International Conference of Students of Systematic Musicology (SysMus21)*. DOI 10/jx9r.

AUTHOR CONTRIBUTIONS BC designed and conducted the research, wrote the conference abstract, and drafted this chapter. JAB and WZ edited the abstract and commented on the chapter.

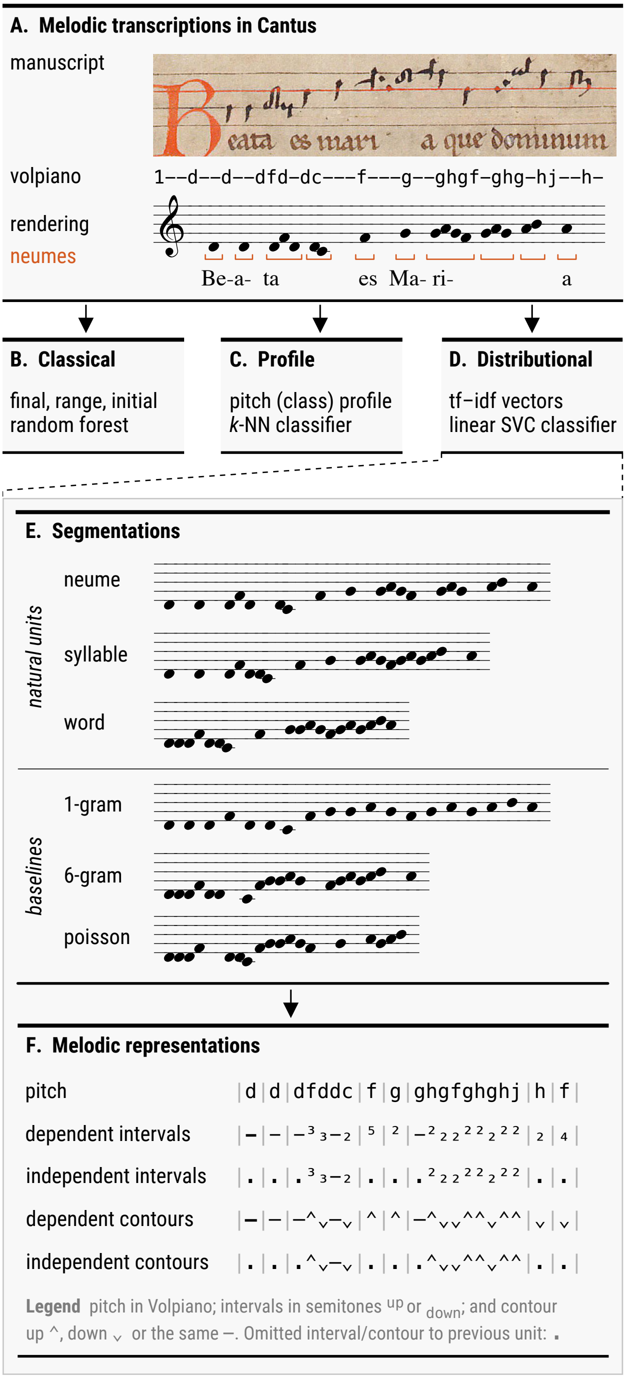
CHAPTER 4

Article

Modes

Many musics across the world are structured around multiple *modes*, which hold a middle ground between scales and melodies. We compare three approaches to classifying mode in a corpus of 20,865 medieval plainchant melodies from the Cantus database. The traditional 'textbook' approach and the only prior computational approach work well, but largely reduce modes to scales and ignore their melodic character. We propose a model using tf–idf vectors that reaches 93–95% F_1 score on mode classification, compared to 86–90% using traditional pitch-based methods. Importantly, it reaches 81–83% even when we discard all absolute pitch information and reduce a melody to its contour. Our model strongly depends on the choice of units: i.e., how the melody is segmented in motifs. If we borrow the syllable or word structure from the lyrics, the model outperforms all of our baselines. To better understand how the classifier works, we propose an attribution method, *witness coloring*, that highlights the motifs that strongly contribute to the resulting classification. Taken together, our results suggest that, like language, plainchant is made up of 'natural' units, in our case, between the level of notes and complete phrases.

FIGURE 4.1 – **Overview of this study.** We compare three approaches to mode classification in a corpus of Gregorian chant. Cantus contributors have transcribed a vast number of melodies from medieval manuscripts **(A)**. We classify mode based on the final, range, and initial in the *classical approach* **(B)**, and based on pitch (class) and repetition profiles in the *profile approach* **(C)**. Finally, in the *distributional approach* **(D)**, we use tf–idf vectors where we tweak two parameters: the *segmentation*, or which melodic units we use **(E)**, and the *representation* **(F)**, where we gradually discard information about the scale when we move from pitches to contours. In this way, we aim to capture the melodic, rather than scalar, aspect of mode.



4.1 Introduction

In his seminal Grove entry, Harold Powers et al. (2001) points out a remarkable cross-cultural generalization: many musics are structured around multiple *modes*. Modes are often associated with the major–minor distinction in Western music. Still, there are much richer systems of modes: examples include Indian *raga*, Arabic *makam*, Persian *dastgah*, *pathet* in Javanese gamelan music, and the modes of Gregorian chant. The specifics obviously vary, but all these phenomena share properties with both scales and melodies and are perhaps best thought of as occupying the continuum in between (Powers et al., 2001). On the one hand, a mode is more than a scale: it might imply a hierarchy of pitch relations or favor the use of characteristic motifs. On the other hand, it is not as specific as a particular tune: a mode instead describes a melody *type*. Modes are of central importance to their musical tradition, both as means to classify the repertoire and as practical guides for composition and improvisation. Characterizing modes computationally is, therefore, an important problem for *computational ethnomusicology*.

Several studies have investigated automatic mode classification in Indian *raga* (Chordia & Rae, 2007; Gulati et al., 2016), Turkish *makam* (Atalay & Yöre, 2020; Ünal et al., 2012) and Persian *dastgah* (Abdoli, 2011; Heydarian & Bainbridge, 2019). These studies can roughly be divided into two groups. First, studies emphasizing the scalar aspect of mode usually look at pitch distributions (Atalay & Yöre, 2020; Chordia & Rae, 2007; Heydarian & Bainbridge, 2019), similar to key detection in Western music. Second, studies emphasizing the melodic aspect often use sequential models or melodic motifs (Gulati et al., 2016; Ünal et al., 2012). For example, Ünal et al. (2012) train n-gram models for 13 Turkish makams and then classify melodies by their perplexity under these models. Going beyond n-grams, Gulati et al. (2016) use motifs, characteristic phrases, extracted from raga recordings to represent every recording as a vector of motif-frequencies. They weigh counts, amongst others, by the *inverse document frequency*, which balances highly frequent motifs and favors specific ones.

This chapter focuses on automatic mode classification in Western medieval plainchant. This has rarely been studied computationally, even though the term (if not the phenomenon) 'mode' originates there. At first glance, mode in plainchant is relatively clear, though certainly not entirely unambiguous. With a second glance, it has a musicological and historical depth that inspired a vast body of scholarship going back over one thousand years. The music is indeed sufficiently distant in time from most other musics, including Western classical and pop music, to provide an interesting cross-cultural comparison. And for once, data is abundant, thanks to the immense efforts of chant scholars.

Computational studies of chant have mostly been concerned with optical music recognition of medieval manuscripts: the SIMSSA project, for example, has used such systems to transcribe plainchant from the Cantus database (Helsen et al., 2014). Recent ISMIR conferences have also included analyses of Byzantine plainchant (Panteli & Purwins, 2013) and Jewish Torah tropes (van Kranenburg et al., 2011), and a comparison of five Christian chant traditions using interval n-grams (van Kranenburg & Maessen, 2017). But, to the best of our knowledge, the study by Huron and Veltman (2006) is the only computational study addressing mode classification in chant. The study used pitch class profiles and thus approached mode as a mostly scalar phenomenon. Wiering (2006) later criticized the study, partly for ignoring the melodic character of modes.

We aim to revisit this work on a larger dataset and model the melodic aspect of mode. Concretely, we compare three approaches to mode classification:

1. CLASSICAL APPROACH: based on a chant's range, final, and initial note.
2. PROFILE APPROACH: uses pitch, pitch class, and repetition profiles, inspired by Huron and Veltman (2006).
3. DISTRIBUTIONAL APPROACH: uses tf–idf vectors based on various segmentations and representations of the melody.

Besides evaluating mode classification, we ask how the task is solved. Using a linear classifier for the distributional approach allows us to explain the model behavior in more detail. In particular, we propose an attribution method to visualize which motifs contribute to the classification of a chant.

4.2 Methods

The design of this study is visualized in Figure 4.1.

DATA: THE CANTUS DATABASE We use chant transcriptions from Cantus Corpus (v0.2), a dump of the Cantus database tailored for computational research containing 497,071 chants (see chapter 2). We here only consider chants that have a Volpiano transcription (63,628 chants) and further filter out chants with incomplete or non-standard transcriptions, without a complete melody, without 'simple' mode annotation, and exact duplicates (see supplement A2). This resulted in 7031 responsories (966,871 notes, avg. length 138 notes) and 13,865 antiphons (825,143 notes, avg. length 60 notes). We fixed a 70/30 train/test split for all datasets and only used training data in exploratory analyses. Cantus often contains multiple variants of any particular melody, transcribed from different manuscripts (see supplement A10). One may wonder whether the simple train/test split is sufficient, or whether even more care is needed to avoid overlap between such melodic variants in the train and test sets. This is a difficult issue that also applies to other musical corpora (e.g., the Essen folk-song corpus), and for which there is no perfect solution. To assess the effects, we have also repeated our experiments on a subset without variants, which we discuss in supplement A12.

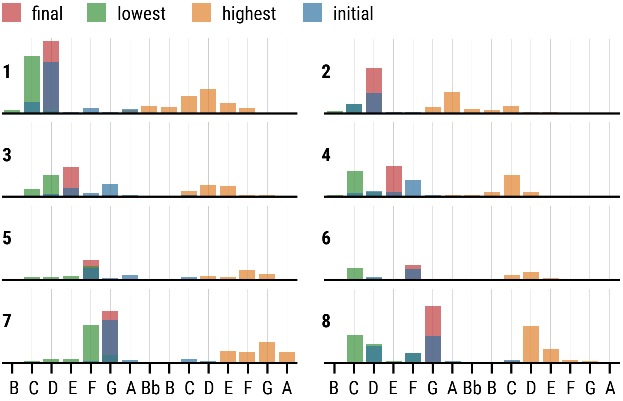


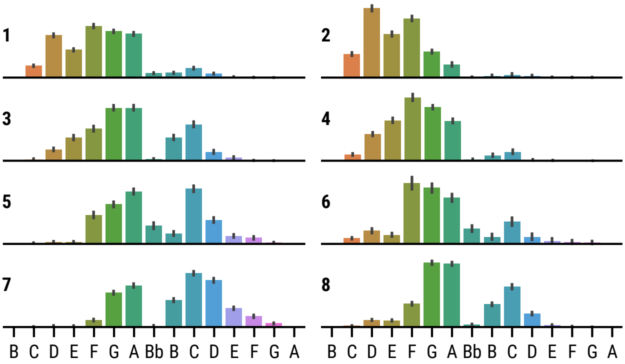
FIGURE 4.2 – Classical features. The classical approach uses the final, range, and initial to determine the mode. The overall distribution for each mode (1–8) is clearly different, although not entirely without ambiguity.

According to the transcription guidelines, flat symbols are transcribed only once, directly before the first flattened note. We replace the first and later flattened notes by the corresponding accidental, a Volpiano character that sits at a specific staff line. In this way, flat notes are also encoded by a single Volpiano character. We discard characters like clefs and pausas and only retain the notes, accidentals, and boundaries (hyphens). The resulting string is used in our three classification experiments, which we now discuss.

CLASSICAL APPROACH: FINAL, RANGE, INITIAL The first approach is motivated by the classical procedure for mode classification. We extract three features from every chant: the final pitch, the range (lowest and highest pitches), and the initial pitch. Theory suggests that the final alone should give an accuracy of roughly 50%, and adding the range should further increase that by roughly 50% if there is no ambiguity. Figure 4.2 shows the feature distributions for all modes. It suggests that there is some ambiguity, and so numbers will be a little lower. For this task, we use random forest classifiers (Breiman, 2001), which aggregate multiple decision trees. Training details of all models are discussed below.

PROFILE APPROACH: PITCH (CLASS) PROFILES The second approach is inspired by Huron and Veltman (2006). Using 97 chants from the Liber Usualis, they compute average *pitch class profiles* (the relative frequency of each pitch class) for each mode and then classify chants to the closest profile. We take a similar approach and use k-nearest neighbor classification, where k is tuned (see paragraph 1). In a commentary, Wiering (2006) argued for using actual pitches rather than pitch classes, as the pitches an octave above the final have a very different role than those an octave below it. We follow that suggestion by also computing *pitch profiles* (Figure 4.3). Finally, we propose a *repetition profile* aiming to describe which notes function like a recitation tone. For every Volpiano pitch q we compute a repetition score $r(q)$, which is the relative frequency of direct repetitions, and collect these to get a repetition profile. Formally, if a chant has pitches

FIGURE 4.3 – Pitch profiles. These show the relative frequency of every pitch in the eight modes. Again, although the distribution of individual modes is clearly distinct, some residual ambiguity remains.



$p_1, \dots, p_N$, then $r(q) = \#\{i : p_i = q \text{ and } p_{i+1} = q\}/(N-1)$ since there are $N-1$ possible repetitions.

DISTRIBUTIONAL APPROACH: TF–IDF VECTORS Our third approach aims to capture the melodic aspect of mode. In short, we use a bag of 'words' model (cf. Gulati et al., 2016) and tweak two parameters: the segmentation (which melodic units to use as 'words') and the representation (pitches, intervals, and contours). The idea is to discard more and more information about the scale and see if we can nevertheless determine the mode.

First, the units. For chant, three natural segmentations suggest themselves: one can segment the melody (1) at neume boundaries, but also wherever we find (2) a syllable or (3) a word boundary in the lyrics. Given the close relationship between text and music in chant, there is some reason to believe that these are meaningful units. Conveniently, all of these boundaries are explicitly encoded in Volpiano by a single, double, and triple dash, respectively. Note that these natural units are nested: neumes never cross syllable boundaries. We compare the natural units to two types of baselines. The first is an n-gram baseline where we slice the melody after every n notes, for $n = 1, \dots, 16$. The second is a random, variable-length baseline. Here the melody is segmented randomly, but in such a way that the segment length is approximately Poisson distributed with a mean length of 3, 5, or 7. We stress that all these units are proper segmentations: units do not overlap. In particular, we choose not to use a higher-order model (using n-grams of units) because we are only interested in comparing different segmentations.

Second, the representation. We represent melodies in three ways: as a sequence of *pitches*, *intervals* (the number of semitones between successive notes), and *contours* (the direction of movement between successive notes: up, down, or level).[1] There is one complication when segmenting sequences of intervals or contours: we introduce dependencies between the units. All units would, for example, start with the interval from the previous unit. We call this a *dependent* segmentation. Alternatively, you could discard the intervals between units to obtain an *independent* version. This effectively makes every unit one interval shorter. We analyze both the independent and dependent versions. Since we use a text-based represen-

[1] In other chapters, 'contour' denotes any description of the general shape of a melody (see chapter 6 and chapter 8). In this chapter, it however denotes one particular representation, also known as Parsons' code (Parsons, 1975).

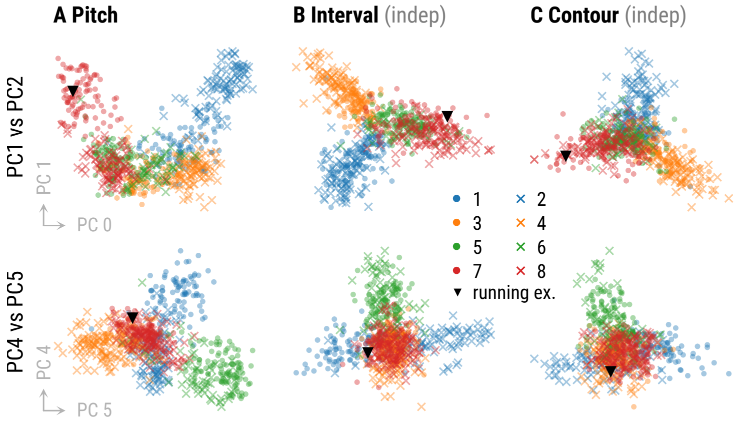


FIGURE 4.4 – PCA of tf–idf features. Principal component projection of the chant vectors, after projecting them on the eight decision axes of the corresponding SVM classifier. This helps separate the modes in the visualization since each decision axis is optimized to separate one mode from the rest (see also section 4.4). The top row plots principal components 1 against 2; the bottom row 4 against 5. Even in the contour representation, the modes are fairly clearly separated.

tation, we found it convenient to start all independent units (including the first) with a dot to keep the segmentation identical across representations. You can think of the dot as marking the omitted interval to the previous unit.

Third, the model. Given a segmentation, we represent every chant by a vector of unit frequencies, but weighted to favor frequent, yet *specific* units: units that do not occur in too many chants. A standard way of doing this in textual information retrieval is using *term-frequency inverse-document-frequency* (tf–idf) scores, which multiply the frequency of a term in a document (tf) by the inverse document frequency (idf). The latter is computed as

$$\mathrm{idf}(t) = \log\left(\frac{1+n}{1+\mathrm{df}(t)}\right) + 1, \tag{4.1}$$

where n is the total number of documents and $\mathrm{df}(t)$ is the *document frequency*: the number of documents containing term t. Intuitively, this factor decreases the scores of common terms that occur in many documents. We use at most 5000 features and found it was important *not* to set a minimum or maximum document frequency. Finally, we determine the mode of a chant by feeding its tf–idf vector to a linear support vector machine. We discuss the classifier in more detail in section 4.4.

In sum, we analyze 22 segmentations (3 natural ones, 16 n-grams, 3 random) and 5 representations (pitch and dependent/independent interval/contour), giving a total of 110 conditions. Figure 4.4 shows a low-dimensional projection of the tf–idf chant vectors, colored by mode, in some of these conditions.

TRAINING We tune every model using a randomized hyperparameter search with 5-fold stratified cross-validation. That is to say that we randomly sample hyperparameters from a suitable grid (determined by extensive manual analyses) and determine their performance using 5-fold cross-validation on the training set, where we ensure the class frequencies

A Classical approach

Feature set	Respons.	Antiphon
final	40	49
range	56	61
initial	37	48
final & range	89	79
final & initital	73	72
range & initital	70	73
final, range & init.	**90**	**86**

B Profile approach

Profile	Respons.	Antiphon
pitch class profile	85	88
pitch profile	**88**	**90**
repetition profile	81	84

0% Weighted F₁–score 100%

C Distributional approach

	Responsory					Antiphon				
	pitch	dep. interval	indep. interval	dep. contour	indep. contour	pitch	dep. interval	indep. interval	dep. contour	indep. contour
neume	92	86	79	63	52	92	80	48	39	30
syllable	**93**	**89**	86	79	76	92	81	52	44	35
word	90	87	**86**	**82**	**81**	**95**	**92**	**90**	**85**	**83**
1-gram	87	53	7	20	7	89	54	12	23	12
2-gram	91	74	38	25	17	92	78	38	29	21
3-gram	92	81	65	37	23	94	87	70	38	27
4-gram	91	83	75	47	34	94	90	83	50	34
5-gram	91	84	81	54	43	94	91	88	63	46
6-gram	88	83	82	60	51	93	90	90	69	58
8-gram	82	77	78	67	60	88	85	85	76	70
10-gram	76	72	74	67	66	81	78	79	73	72
12-gram	71	68	69	66	65	74	71	72	69	69
14-gram	66	63	64	61	62	66	64	64	64	63
16-gram	62	58	59	61	61	60	57	57	57	58
Poisson 3	86	68	59	35	26	89	78	66	38	29
Poisson 5	79	63	60	40	34	85	77	72	48	41
Poisson 7	68	57	55	41	37	79	72	69	53	48

FIGURE 4.5 – **Classification results.** Weighted F_1-score for three approaches to mode classification, using two chant genres: responsories and antiphons. Scores are averages of five independent runs of the experiment. The classical approach **(A)** using the final, range, and initial reaches F_1-scores of 90% and 86%. The profile approach **(B)** works better for antiphons (90% vs. 86%) and somewhat worse for responsories (88% vs. 90%). As Wiering (2006) suspected, pitch profiles outperform pitch *class* profiles by a small margin. The distributional approach **(C)** reaches the highest F_1 scores of 95% on both responsories and antiphons. The choice of segmentation (vertically) is crucial: classification is improved by using 'natural' units, word-based units in particular, rather than n-grams. As the representation (horizontally) becomes cruder, from pitches to intervals and finally to contours, the task becomes much harder. But, when using word-based segmentation, performance remains high.

are similar in all folds. We use the hyperparameters yielding the highest cross-validation test accuracy to train the final model. All models were implemented in Python using scikit-learn (Pedregosa et al., 2011).

4.3 Results

Figure 4.5 gives support-weighted[2] averages of F_1-scores obtained on the full test sets for all three approaches. The scores are averages of five independent experiment runs using different train/test splits. Standard deviations were small and are included in supplement A11. We now com-

[2] The retrieval scores for all classes (modes) are averaged, weighted by the number of instances in each class.

pare the three approaches and then discuss the effect of representation and segmentation on the distributional approach.

APPROACHES: DISTRIBUTIONAL APPROACH WORKS BEST First of all, we report the highest classification scores with our distributional approach using pitch representations: an F_1-score of 93% for responsories and 95% for antiphons. This corresponds to an error reduction of 30–60% compared to the classical approach (90% and 86%). The classical approach confirms the rule of thumb: the range and final are very informative features. Using only these, we obtain F_1-scores of 89% and 79%, which are further increased by also adding the initial. The profile approach outperforms the classical approach for antiphons (90% vs. 86%) but is outperformed for responsories (88% vs. 90%). Our results support Wiering's (2006) intuition that pitch profiles more accurately describe mode than pitch *class* profiles, but the effect is small: it increases F_1 scores by 2–3%. Repetition profiles appear to be less useful for both genres.

Our results in broad strokes validate the classical and profile approach, which both peak around a 90% F_1-score, using simple features. The distributional approach improves this, up to 95% using complex features. Importantly, we now show that the distributional approach maintains high performance when using interval or contour representations.

REPRESENTATIONS: CONTOURS ARE SUFFICIENT We find that the classification task gets harder when the representation gets cruder, from those based on pitch, to intervals and finally to contours (Figure 4.5C, horizontally). This was anticipated: cruder representations are obtained by discarding information from every unit. Shorter units are impacted more by this information loss. For example, the performance with 1-grams drops by over 75% when moving from pitch to independent contour representation. At that point it performs at majority baseline (a 7% F_1-score for responsories and 12% for antiphons).[3] For longer units such as 10-grams, the drop is not as dramatic (around 10%). However, this comes at the cost of a comparatively low performance using the pitch representation, presumably because of increasing sparsity.

Natural units, however, escape this trade-off. Word-based segmentations perform consistently well, dropping only 3% below the classical baseline using the highly impoverished independent contour representation. In contrast to the other representations, the contours do not carry any information about the scale: the same contour can be reproduced in any scale. Apparently, we can discard the scalar aspect of mode and still classify it: contours alone contain sufficient information for mode classification. The success of pitch-based methods might obscure that mode is as much a melodic phenomenon as a scalar one.

Interestingly, the earliest chant notation used *unpitched* neumes that mainly described the contour of the melody—not the exact pitches. Our results reinforce the idea that contour is highly informative—so informa-

[3] Every unit is identical for 1-grams in the independent interval *and* contour representation: a dot representing the omitted contour to the previous note. The majority class for both responsories and antiphons is mode 8, taking up 21% and 28% of the test data respectively (see supplement A4). This is precisely the accuracy of the model in those conditions.

tive that given a mode, text, and contour, an experienced singer could reconstruct the chant melody.

SEGMENTATIONS: NATURAL UNITS WORK BEST. Our most important result is that natural units (neume, syllables, and words) yield the highest classification performance among all the representations we considered. The 4- and 6-gram baselines also reach top F_1-scores in antiphons, but only when we use representations that include information about pitch. Furthermore, the success of natural units cannot be explained solely by their length. In responsories, neumes, syllables, and words are on average 2.3, 3.0, and 7.1 notes long, respectively (see supplement A6). Yet, the performance of these natural units is consistently higher than n-grams of comparable length. The performance of the natural units is also consistently higher than that of the variable-length Poisson baselines, which are intended to mimic the overall distribution of natural lengths but ignore musical and textual semantics.

A few other observations merit discussion. Firstly, although neume and syllable segmentations behave differently for responsories, they behave similarly to each other for antiphons. The reason may be that neumes and syllables more often coincide in antiphons. Antiphons are less *melismatic* than responsories (i.e., they use fewer notes per syllable, 1.5 to be precise). Secondly, both the n-grams and the Poisson baseline perform better on antiphons than on responsories, possibly because the n-grams are more likely to end up being coincidentally aligned with the natural units the less melismatic the genre.

CONTROLLING FOR MELODIC VARIANTS We repeated all experiments on a subset of the data from which we removed melody variants (see supplement A12 for details). In terms of the number of notes, this meant a 75% and 66% reduction in data size for responsories and antiphons respectively. The performance of all models decreased on this subset, and for responsories more than for antiphons. Our main findings that contours are sufficient and that natural units work best across representations stand. We do observe some reorderings: some already high-performing n-grams in antiphons, for example, slightly overtake word segmentations, although only for pitch and dependent interval representations. The distributional approach works best for antiphons regardless of including or excluding chant variants. Still, for responsories, the distributional approach drops slightly below the classical approach on the subset (where the profile approach is worst). These findings might be explained by increased sparsity in the smaller dataset: natural units in responsories are, after all, longer.

4.4 Attribution

The distributional approach to mode may classification work well, but *how* so? This section aims to explain and visualize in detail why a chant

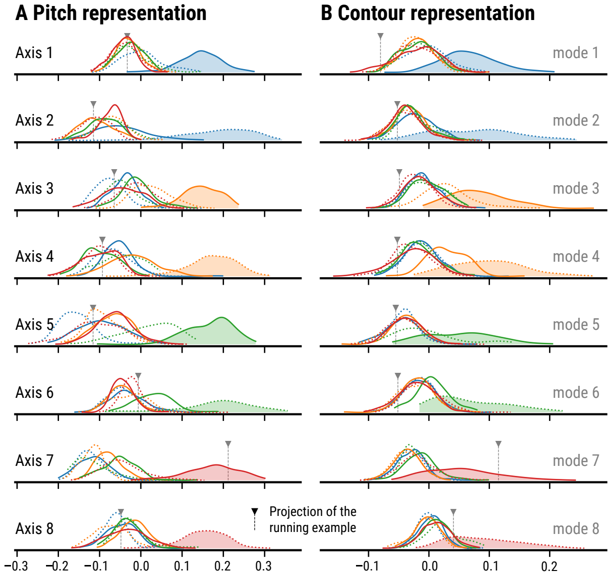


FIGURE 4.6 – **Decision axes.** The eight axes describe the decision planes of the classifier. Each axis discriminates one mode from the rest. The modes are better discriminated in the pitch representation **(A)** than the contour representation **(B)**, consistent with our classification results.

is classified to a particular mode. More precisely, we highlight which motifs contribute to a particular classification and, in that way, *attribute* a classification to specific motifs.

Recall that we represent chants as high-dimensional tf–idf vectors. We then used a linear support vector machine to determine eight decision boundaries: hyperplanes in that high-dimensional space. Each boundary separates chants of a particular mode from chants of the seven other modes in a so-called one-vs-rest classification scheme. A linear decision boundary is represented by a decision vector orthogonal to it, that essentially points out where that boundary lies. But this vector also defines a *decision axis* along which one mode is discriminated from the rest. Figure 4.6 illustrates the decision axes: it shows the distribution of chants after projecting them on each of the eight decision axes. Chants whose mode corresponds to the decision axis tend to get positive projections, while the other chants get small negative values.

Computing the projection of a chant on a decision axis amounts to summing the chants' tf–idf scores, weighted by the coefficients of the decision vector. Since tf–idf scores are positive, if the k-th coefficient or weight has a large positive value, then the corresponding motif can strongly contribute to a large projection. The k-th motif may, in other words, be important for classifying the chant to the axis' mode. Consequently, we use the coefficients of the decision axes as a measure of *class-wise feature importance*. It is a 'class-wise' measure in the sense that it measures the importance for classifying to one particular class—or mode, in this case. We discuss a 'general' variant in supplement A13,[4]. where we also discuss using tf–idf scores as a measure of feature importance.

[4] Counter-evidence, in the form of strongly negative weights, does *not* contribute to the class-wise importance measure discussed here. In the general version, both strong evidence and counter-evidence indicate importance.

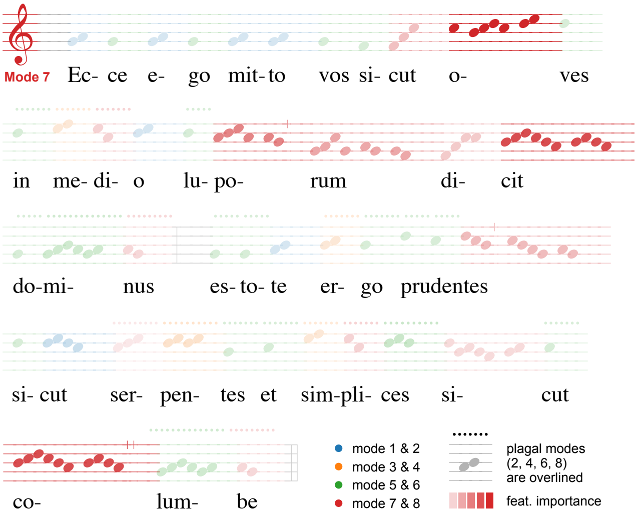

Mode 7
Ec- ce e- go mit- to vos si- cut o- ves
in me- di- o lu- po- rum di- cit
do-mi- nus es- to- te er- go prudentes
si- cut ser- pen- tes et sim-pli- ces si- cut
co- lum- be
● mode 1 & 2
● mode 3 & 4
● mode 5 & 6
● mode 7 & 8
plagal modes (2, 4, 6, 8) are overlined
feat. importance

FIGURE 4.7 – Attribution using witness coloring. Our attribution method visually indicates to which motifs the classification can be attributed. It treats every motif as a *witness* for the one mode from which it gets the highest class-wise feature importance. We color motifs according to the mode they witness: mode pairs share a color, but plagal modes have a dashed line above them. The opacity indicates the feature's importance: darker features are more important.

To visualize the importance of a motif, we look up its importance scores $f_1, \ldots, f_8$ for all eight modes. If the motif corresponds to index k, these scores are the k-th entries of the eight decision vectors. One of the scores, say f_m, will often be markedly larger than the others so that the occurrence of that motif can be seen as evidence for mode m. Differently put, the motif *witnesses* mode m. This motivates our visualization method, *witness coloring*, that colors the motifs according to the mode they witness. The importance scores are visualized by varying the opacity. As distinguishing transparencies is difficult (Cleveland & McGill, 1984), we scale the opacity cubically between 10% and 100%, to make the most important motifs stand out.

We have implemented the attribution method using Chant21. Figure 4.7 shows the result for the mode 7 responsory *Ecce ego mitto vos* (D-KNd 1161, folio 108r), using a syllable segmentation and a pitch representation. The visualization highlights syllable motifs that contribute to a (correct) mode 7 classification, such as *o-* on the first line, *po-*, *rum-*, *di-* and *-cit* on the second line. The last motif also occurs in the final line. In supplement A14, we further illustrate our visualization method using antiphons. Antiphons are sung before and after a psalm and end with so-called *differentiæ* that set the final words of the psalm (*seculorem amen*) and connect it back to the antiphon. As discussed in section 2.5, differentiæ are fairly indicative of mode, and accordingly, they are highlighted by our attribution method in interval and contour representations.

4.5 Discussion and Conclusion

In this paper, we analyzed three approaches to mode classification in a large corpus of plainchant: (1) the classical approach using the final, range, and initial; (2) the profile approach using pitch (class) profiles and (3) the distributional approach using a tf–idf vector model and various segmentations and representations. We found that the distributional approach performs best and that it can maintain high performance on contour representations if using the right segmentation: at word boundaries, in this case. Analyzing the distributional approach in more detail, we proposed an attribution method that visualizes which motifs are important for the classification.

Although our results are specific to one corpus of medieval music and one classification task, we believe our conclusions are of wider relevance. We often fall back on *n*-grams because they are well-understood and easy to use. A more natural segmentation may be harder to obtain, but if finding them can have such a large effect on a relatively simple task like mode classification, their advantages may be even stronger for more complex tasks.

A first next step could be to explore whether lyrics yield equally useful units in other vocal musics. As noted, plainchant's link between text and music is particularly tight. This at least suggests that the text may be useful in other types of chant, like Byzantine chant or Torah trope. For folk melodies designed to standard poetic meters, it is not as obvious whether lyrics would help or hinder the identification of useful units. This is worth investigating, as characteristic motifs and repeated patterns are commonly used in computational folk-song studies, particularly for tune family identification (Janssen et al., 2017; Volk & van Kranenburg, 2012).

Our results raise another question: is chant indeed composed by stringing together certain melodic units, much like a sentence is composed of words? It has been suggested (and disputed) that Gregorian chant is composed in a process of *centonization* and that a chant is a patchwork of existing melodic chunks called *centos*. A recent study used the tf–idf weighting to discover centos in Arab-Andalusian music (Nuttall et al., 2019). This raises the possibility that classification using natural units may have been successful because they indeed are the building blocks, the *centos*.

Computational studies of plainchant are still quite rare, and we hope this study shows that chant is an interesting repertoire that can yield insights of broader relevance. The immense efforts of chant scholars mean that data are abundant. In short, we think chant can aid the development of models that apply beyond Western classical and pop music and embrace the true diversity of musics around the world.

SUPPLEMENTS **A1** Data and code · **A2** Filtering · **A3** Dataset statistics · **A4** Majority baselines · **A5** Chant lengths in two genres · **A6** Mean lengths of natural units · **A7** Lengths of natural units · **A8** Pitch class profiles · **A9** Repetition profiles · **A10** Melody variants in Cantus. · **A11** Results with standard deviation · **A12** Main results on subset · **A13** Feature importance · **A14** Witness coloring antiphons

DATA AND CODE Cantus Corpus v0.2 is available via github.com/bacor/cantuscorpus. All code can be found on github.com/bacor/ISMIR2020.

REFERENCE The majority of this chapter was published as Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2020a). Mode classification and natural units in plainchant. *Proceedings of the 21st International Conference on Music Information Retrieval*, 869–875. DOI 10/jx94. The work on witness coloring has been presented at MedRen 2022: Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2022). Understanding automatic mode classification in Western plainchant. *50th Medieval and Renaissance International Music Conference (MedRen 2022)*.

AUTHOR CONTRIBUTIONS BC designed and conducted the research and drafted the original conference paper. JAB and WZ supervised the research and edited the manuscript.

ACKNOWLEDGEMENTS We would like to thank the anonymous reviewers for their comments. We have tried to address all your concerns.

CHAPTER

5

Interlude

Neural chant

CHANT SCHOLARSHIP is an intimidating field of study for an outsider. The literature breathes an intimate familiarity with practices, repertoires, and manuscripts I do not have. Therefore, my approach to chant has not been a humanistic close reading but a computational distant reading. But our reading in the previous chapter relied on a questionable assumption: we treated chants as unordered bags of motifs and ignored their temporal order. In this interlude, I would like to revisit the same chant—Cantus Corpus v0.2—but using a model that *does* respect temporal order. For this, we will use a *long short-term memory* (LSTM) network (Hochreiter & Schmidhuber, 1997), rather than, say, a state-of-the-art transformer. One might motivate this choice in various ways—inspired by previous work, a study in interpretability, or just a proof of concept—but I hope this short interlude will motivate itself.

5.1 Recurring connections

Neural networks are quite literally graphs of computations. Every node in this graph has a certain activation, computed as a weighted sum of its inputs, which is then transformed in a nonlinear way. *Recurrent* neural networks are designed to model sequential data and to that end, contain nodes with a connection to themselves. Besides ordinary inputs, these nodes also receive their own output from the previous time step as an input. An LSTM is a recurrent network, but its recurrent units are not plain nodes. Each unit contains a so-called *cell state* that can retain information over a long time span and influences the unit's output. The cell state is updated based on the input via multiple gates, whose parameters are learned when training the network.

How does training work? We present the network with a Volpiano character, represented by a numerical index, and 'ask' it to predict the

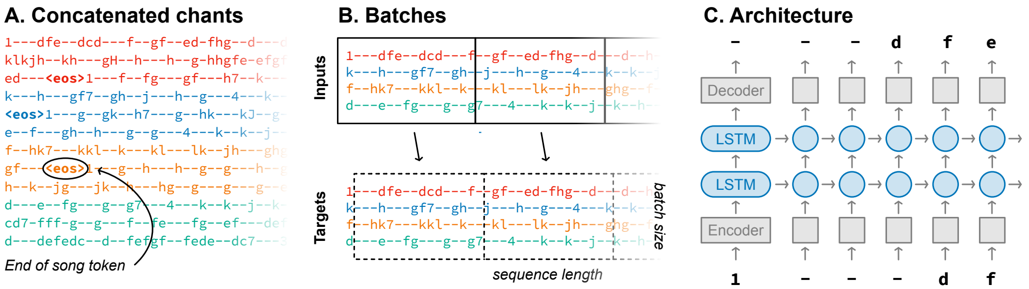


FIGURE 5.1 – The neural chant model. The model consists of two recurrent layers with LSTM units **(C)** and is trained to predict the next character given an input character. To do this efficiently, all chants are concatenated **(A)** and divided up in parallel batches **(B)** of a fixed length.

next character. For that reason, the network's final layer has as many nodes as there are characters. Those nodes can output how probable it is, according to the model, that a character is the next one. When training, we know what the next character *should* be. That allows us to measure the error in the predictions of the model. The crucial trick is to differentiate the error signal and determine in what direction to change the model's parameters so as to make the correct prediction more likely. We then update the parameters of the network by taking a small step (given by the *learning rate*) in that direction.

But how are the predictions computed precisely? We start with a character (see Figure 5.1C). The first step is to *embed* (the index of) a character into a high dimensional space; this is also known as the *encoding* step. While the indices are meaningless, the embedding hopefully organizes them in a useful or even meaningful way. Next, the embedded character is passed to the recurrent layer, together with the previous state of the hidden layer. This is where the network integrates the current input with what it has seen before. We feed the output through a second recurrent layer before *decoding* those outputs to a distribution over the vocabulary that indicates the most probable next character.

Instead of presenting the network one character at a time, it is more efficient to present it with small batches of inputs (see Figure 5.1A and B). First, we concatenate all chants, separated by <eos> tokens that mark the end of a song.[1] Then we split this sequence into B parts (the batch size) of equal length and form a batch by taking S characters (the sequence length) from each of the B parts. You can think of this as sliding over the corpus in B parallel parts. We also truncate the flow of errors back in time to S steps. This is known as *truncated backpropagation through time* and means that there is no explicit error signal for more than S time steps.

Although we train the network to predict the next character, that is not the task we are after. To predict the next character, it turns out that the model needs to build up meaningful representations of the input data (a form of representation learning). Suppose we train it on next-*word* prediction. In that case, the embedding space might, for example, become a model of word meanings, and the network's internal representations may start to show sensitivity to grammatical categories, which may be

[1] We shuffle the order of the chants—but not their characters, of course—at the beginning of each epoch.

	Model			Parsing errors	Mode classif.		Genre clf.
Name	Emb.	Hid.	Params		F_1	SOTA	
VOLPIANO-SMALL	8	64	56k	11,4%	63%	93–95%	85%
VOLPIANO-LARGE	32	256	838k	1,7%	84%		86%
INTERVAL-SMALL	8	64	55k	—	49%	89–92%	66%
INTERVAL-LARGE	32	256	833k	—	56%		71%
CONTOUR-SMALL	8	64	53k	—	37%	81–85%	77%
CONTOUR-LARGE	32	256	825k	—	42%		77%

FIGURE 5.2 – Overview of models and results. We trained a large and small model on next-character prediction using three representations: raw Volpiano, intervals, and contours. The large VOLPIANO model clearly learns the Volpiano syntax: only 1,7% of the samples from the model are not parseable. We also trained a linear SVM on a chant's final hidden representation to predict its mode or genre. This performs better on predicting genre (final column) than mode (columns before that).

detected in the outputs of the recurrent units. Something like this is what we are really after: rich chant representations. Afterward, we can also use the network for something else. Suppose we present the network with a character to start with, like a clef, and then sample the next character from the model's predictions. Append the result to the clef, repeat, and the model is composing a new chant.

TECHNICAL SUMMARY Let me summarize all that in more technical terms. The model architecture and its implementation are inspired by Gulordava et al. (2018). We train a 2-layer LSTM on next character prediction using a cross-entropy loss. We split the training data into mini-batches consisting of 32 sequences of 64 characters. The learning rate is dynamically adjusted using Adam with default parameters. For each of the three representations, we first broadly tuned the embedding size, hidden size, sequence length, learning rate, batch size, initialization range, dropout, and gradient clip with HyperOpt using ASHA scheduling and then more finely tuned the learning rate and sequence length using a population-based training. We then fixed the batch size to 32, the initialization range to $(-0.15, 0.15)$, the dropout to 0.15, and the clipped gradients to 0.5. All models are implemented in PyTorch (Paszke et al., 2019), and tuning is done using Ray Tune (Liaw et al., 2018). We train two classes of models, SMALL ones with an embedding size of 8 and a hidden size of 64, and 'LARGE' ones with an embedding size of 32 and a hidden size of 256. All models were trained to predict the next character, but using three different chant representations from chapter 4: plain VOLPIANO, INTERVALS and CONTOUR (see also Figure 5.2).[2] Let's see how those models learn to chant.

5.2 Learning to chant

STEP 1: VOLPIANO SYNTAX To test whether the VOLPIANO models were training properly, I generated short samples after every few hundred batches, starting from an end-of-song token. In the training data, that token is always followed by the clef of the next chant and some space: something like "<eos>1---." Initially, the predictions are nonsensical, but gradually the model learns to start chants with clefs (see Figure 5.3). A more typical hyphenation pattern also starts to appear: groups of, say, four hyphens no longer occur. It appears, in other words, that the character model learns

[2] I used the dependent interval and contour representation; see chapter 4.

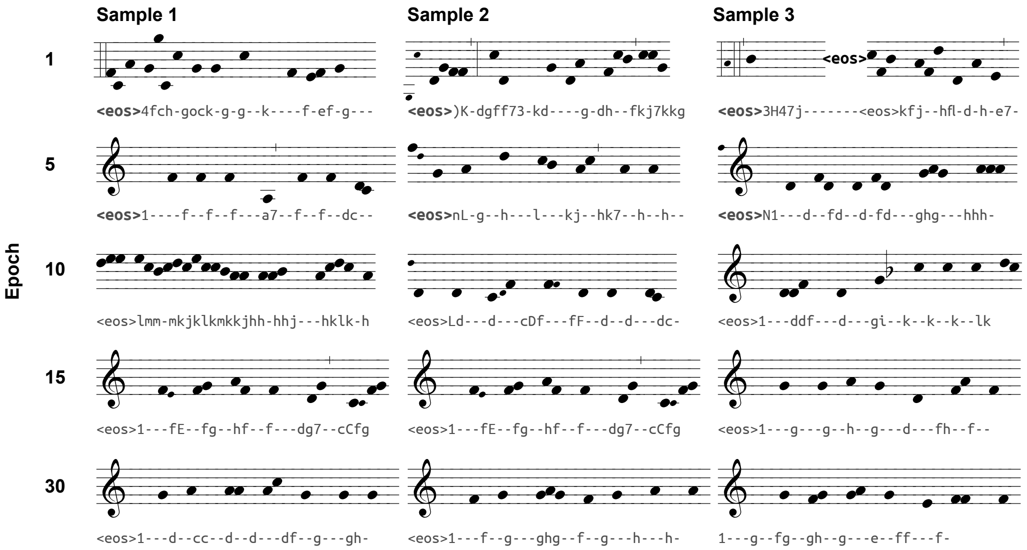

Sample 1
Sample 2
Sample 3
Epoch
1
<eos>4fch-gock-g-g--k----f-ef-g---
<eos>)K-dgff73-kd----g-dh--fkj7kkg
<eos>3H47j-------<eos>kfj--hfl-d-h-e7-
<eos>
5
<eos>1----f--f--f---a7--f--f--dc--
<eos>nL-g--h---l---kj--hk7--h--h--
<eos>N1---d--fd--d-fd---ghg---hhh-
10
<eos>lmm-mkjklkmkkjhh-hhj---hklk-h
<eos>Ld---d---cDf---fF--d--d---dc-
<eos>1---ddf---d---gi--k--k--k--lk
15
<eos>1---fE--fg--hf--f---dg7--cCfg
<eos>1---fE--fg--hf--f---dg7--cCfg
<eos>1---g---g--h--g---d---fh--f--
30
<eos>1---d--cc--d--d---df--g---gh-
<eos>1---f--g---ghg--f--g---h---h-
1---g--fg--gh--g---e--ff---f-

FIGURE 5.3 – **The model learns to start chants with a clef.** Shown are three samples (horizontally) generated by a small model after several training epochs (vertically). The model generated 30 characters, starting from the end-of-song token <eos>. After a few epochs, it has learned to start chants with clefs and use typical hyphenation. Samples are from a very small model (embedding size 2, hidden size 32), but I saw this behavior consistently.

the Volpiano syntax. In fact, I can measure quite precisely how well it does so, using my Volpiano parser from Chant21 (see chapter 2). Of the 1000 samples generated by the *large* Volpiano model, only 1,7% could not be parsed. This closely approaches the $<$ 1% transcriptions in Cantus that could not be parsed. The *small* Volpiano model fares not so well, with over 11% of its samples failing to parse.

STEP 2: PITCH FOR BEGINNERS How does the network represent Volpiano characters? In Figure 5.4, I visualize the character embeddings in two dimensions using UMAP (McInnes et al., 2018). Characters of the same type, such as notes, liquescents (smaller ornamental notes), or bars, tend to cluster. But it appears the embeddings are also ordered according to their pitch: going through the liquescents (orange) in the small model from left to right, we encounter A-B-C-D-F-E-G-K-J-H. Pitches (blue) from top to bottom also appear to have some order. Indeed, we can find embedding dimensions for both models that correlate with the pitch of note characters (see the bottom row of Figure 5.4). The correlation is much stronger for the smaller model, which aligns with my informal impression that small embeddings tend to be more clearly organized. Both models, to some degree, appear to order notes and liquescents by their pitch, and they learn this solely from how characters are distributed in chants.

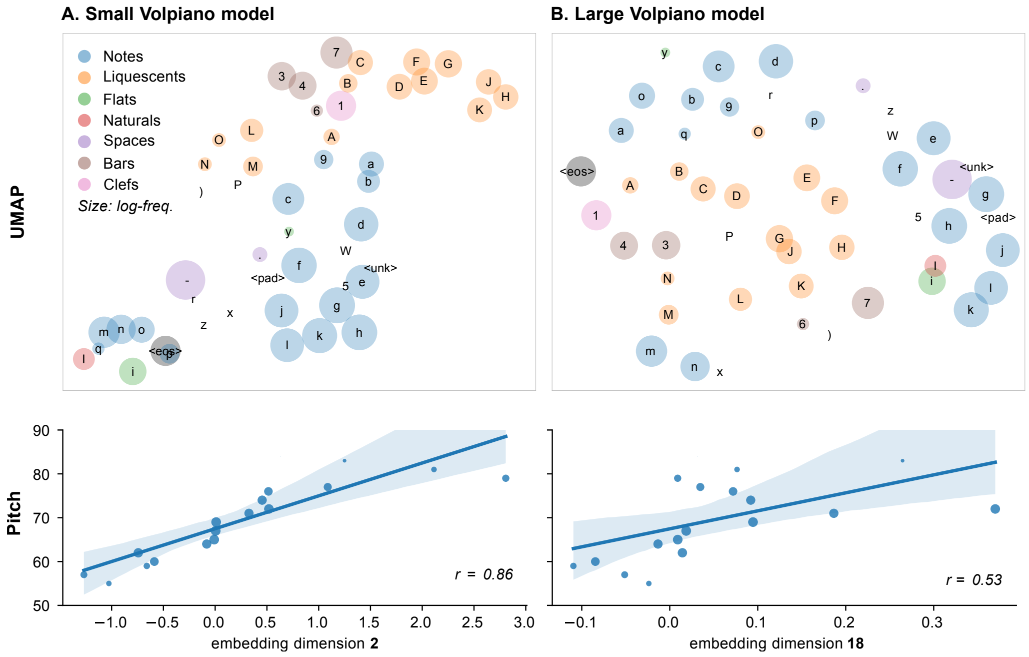


FIGURE 5.4 – Character embeddings of notes correlate with their pitch. The top row shows a UMAP projection of the character embeddings for **(A)** VOLPIANO-SMALL and **(B)** VOLPIANO-LARGE. The points are colored according to their category, and sizes reflect the log frequency of the character. For both models, the bottom row plots the MIDI pitch of note characters against the embedding dimension that best correlates with pitch. This suggests that both models, and the smaller one in particular, learn to represent the pitch of characters.

STEP 3: PITCH FROM INTERVALS It may be unsurprising that pitch is helpful when dealing with melodies, but how about pitch intervals? The INTERVAL-LARGE model is trained to predict sequences of characters like "-3211222" that (we know) encode interval sizes. Does the model also learn this? To find out, I pass an unseen chant through the model and record the hidden state of layer two after every character. This turns a chant into a sequence of 256-dimensional vectors. We now ask two questions. First, can we predict the current *interval* from these vectors (e.g., the current step moves two semitones up)? And second, can we predict the current *pitch* relative to the starting pitch (e.g., we are now five semitones above the starting pitch)? To answer the second question, the model must represent interval sizes *and* compute their cumulative sum.

And indeed, it seems as if the model is doing something like that (Figure 5.5). I trained a linear regressor to predict the interval or the pitch from the hidden representations. This is known as *probing* or *diagnostic 'classification'* (Veldhoen et al., 2016): a way to assess whether a network represents certain information. In this case, intervals can be well predicted ($R^2 = 0.78$), but even pitches are fairly predictable ($R^2 = 0.55$). Figure 5.5 shows two examples of predictions compared to the targets. The contour

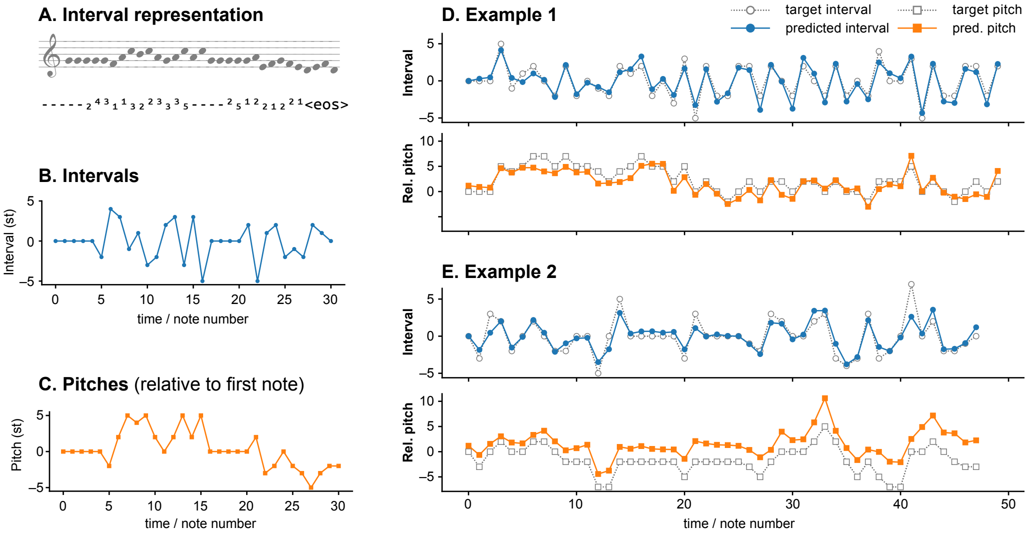


FIGURE 5.5 – An interval model learns to track pitch throughout a melody. The INTERVAL-LARGE model is trained on sequences of characters **(A)** that represent the interval size to the following note **(B)**. By summing up all successive intervals, the actual pitches (relative to the starting pitch) can be obtained **(C)**. It appears that the model learns to represent this information: using linear regression on the hidden representations after every step, we can reasonably well predict the interval (blue) and even the pitch (orange) **(D–E)**.

of the predicted pitches roughly follows the actual contour, even after 50 steps or more. Example 2 illustrates what can go wrong: the predicted pitch contour lies above the target.

STEP 4: MODES AND GENRES Next, we turn to more high-level structures: does the model learn to represent complete chants in a useful way? To find out, I passed unseen test chants through the VOLPIANO-LARGE model and recorded the hidden state of the second layer after seeing the entire chant. This produces a 256-dimensional chant vector. I then trained linear support vector machines on these vectors to predict the mode and the genre. The latter reaches F_1 scores of 71% and 77% for the interval and contour representations, and even 85% for the Volpiano representation (Figure 5.2).[3] But only the latter representation includes hyphenation, which is quite different in syllabic versus melismatic genres. On mode classification, larger models outperform smaller ones, but both perform worse than the tf–idf model in the previous chapter. Then again, we use only the very last representation, the resulting vectors are smaller than the tf–idf vectors, and the LSTM is not explicitly trained to predict mode. And so, this is probably not the performance ceiling.

I also visualized the chant vectors using PCA and UMAP and then colored them according to their mode or genre (Figure 5.6). The first principal component (horizontal) roughly separates chants according to their genre:

[3] I have no good explanation for why intervals here score worse than contours and must leave this to future work.

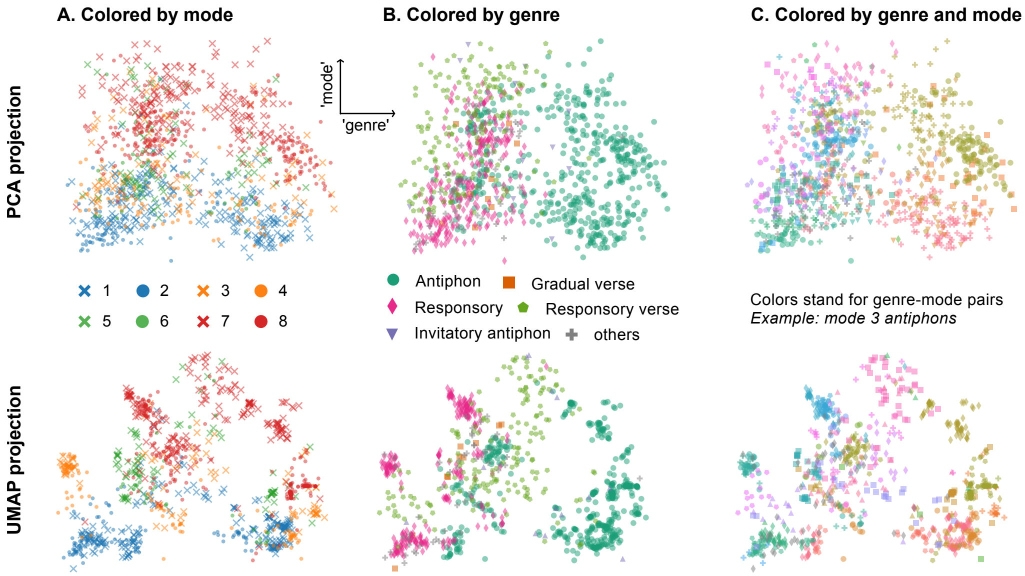


FIGURE 5.6 – Chants are represented along roughly two dimensions: mode and genre. As chant vectors, we use the hidden state of the second layer of the large Volpiano model after processing a complete chant. The top row shows a PCA projection, and the bottom row is a UMAP projection, colored differently in each column. **(A)** The model appears to represent modality, as chants in a cluster appear to have the same mode. Modes appear to correlate with the second principal component, where the first correlates with the genre, as seen in **(B)**: the genres responsory (verse) and antiphon are clearly distinguished along this axis. Overall, clusters appear to be primarily specified by their genre and mode, as shown in **(C)**.

it mainly sets apart antiphons. The second principal component (vertical) appears to capture modality: the modes are ordered in what seems to be the same order as the average pitch height of the modes: 7 > 8 > 5 > 6, and so on. All this is consistent with the idea that genre and mode are central to the organization of the repertoire. The bottom row of Figure 5.6 shows a reasonably similar UMAP visualization but highlights a more local clustering structure. In column C, chants are colored by their genre-mode pair (e.g., mode 4 responsory). The resulting coloring appears to correspond to more local clusters.

Although these results are preliminary and require more work—the pattern is, for example, less evident in an interval model—the implications are provocative. It suggests that chants cluster in groups corresponding to a unique combination of genre and mode. That would mean that for plainchant, the *statistical modes in melody space are not melodic modes*, but something like *genre-mode combinations*. This challenges the hypothesis that melodic modes correspond to statistical modes. But before abandoning the hypothesis, one could wonder whether modes in other traditions, such as raga or maqam, *do* correspond to melody clusters. If so, the notion of *statistical mode* might still be a valuable operationalization of a cross-cultural concept of musical mode. However, it would *not* correspond to chant modes in the traditional sense but to genre-mode combinations. Promising as this line of thought may be, exploring statistical modes in the melody spaces of other traditions is, unfortunately, well beyond the scope of this interlude and left for future work.

STEP 5: CHANTING I cannot end this interlude without letting the model take a final step. After it has learned to produce valid Volpiano, represent pitch, and distinguish modes and genres—can it chant? Figure 5.7 shows the first six chant samples generated by the large Volpiano model, using the clef as the initial seed. All of these are valid Volpiano strings that resemble actual chant in several ways. First, each sample appears to be either *syllabic*, with few notes per syllable (example 1, 2, 3, and 6) or *melismatic* (example 4 and 5), with many more notes per syllable. Of course, the chants have no text, but since the spacing represents boundaries, one can immediately see that samples 4 and 5 appear denser or more melismatic.

Next, the samples appear to be somewhat modal. Figure 5.7 shows the predicted mode of each chant, using a majority vote amongst the tf–idf classifiers from chapter 4 with various segmentations and genres. Except for example 5, the predicted mode is consistent with the modes suggested by the final and range of the chants, shown in the last column of Figure 5.7. Interestingly, examples 2 and 6—both of the syllabic, antiphon-like type—include *differentiæ*: formulae that connect a psalm back to the beginning of an antiphon (see section 2.5). Both differentiæ appear in the Differentiæ Database (I have included their ids) and are usually found in chants with the same modes as the predicted ones.

The generated chant may not yet convince someone well-versed in the repertoire—some future ChantGPT no doubt will—but nevertheless poses excellent puzzles. How can you phrase the material? How can you divide time and stress so that the notes start to make sense and melodies spring to life? The possibilities are endless, but some ways of dividing the melody into phrases seem much more compelling than others—to me, in any case. And that brings me to the next chapter (and its sequel, chapter 8): how are phrases in melodies structured?

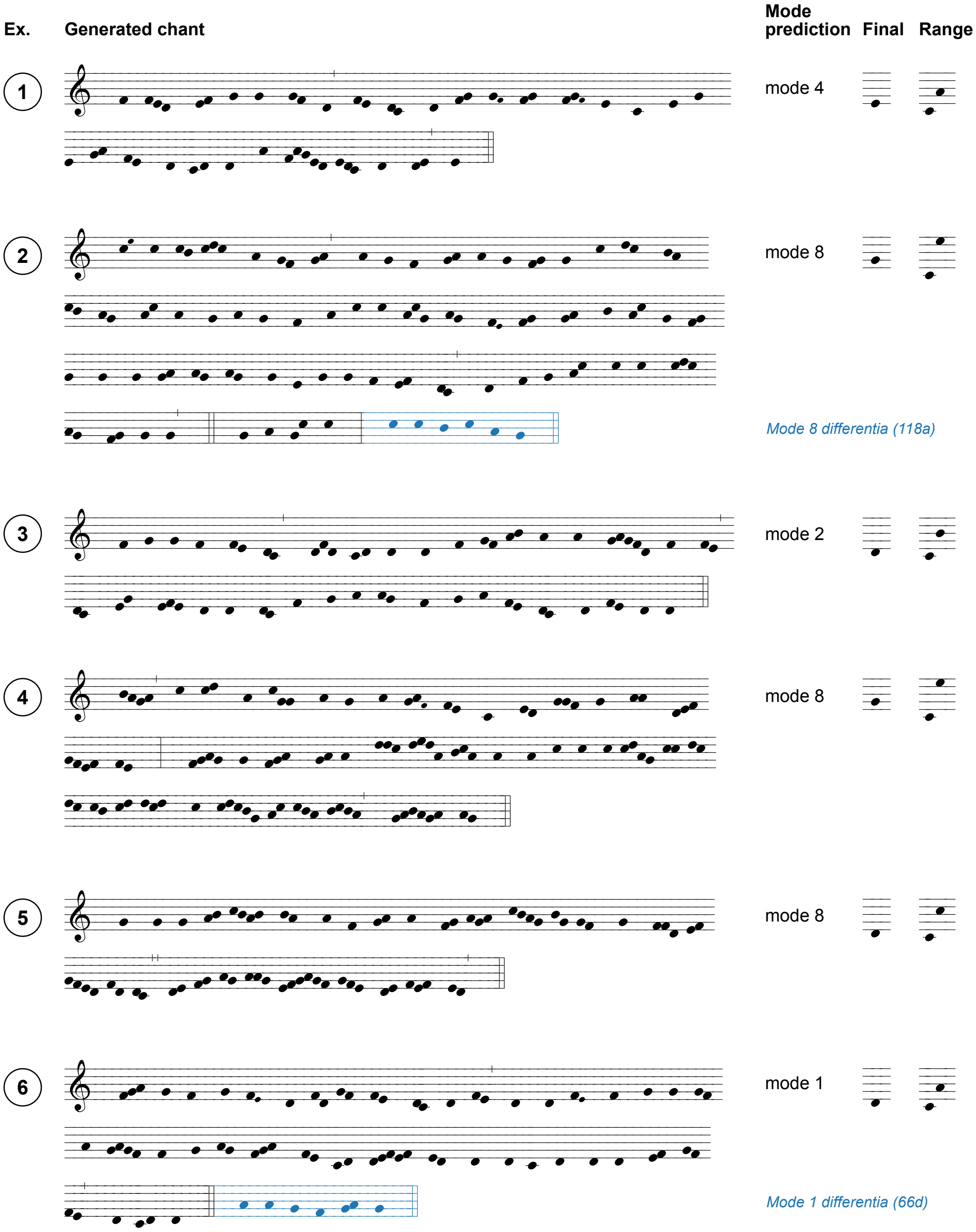


FIGURE 5.7 – Examples of generated chant. Shown are the first six chants generated by the large Volpiano model, all of which are valid Volpiano strings. The mode of these examples was predicted using a majority vote of the tf-idf models with natural units and a pitch representation from chapter 4. Examples 2 and 6 end with so-called differentiæ (blue) that, in both cases, correspond to the predicted mode.

DATA AND CODE Cantus Corpus v0.2 is available via github.com/bacor/cantuscorpus. All code can be found via github.com/bacor/neural-chant.

REFERENCE The work in this chapter has not been published or presented elsewhere.

AUTHOR CONTRIBUTIONS BC and Marianne de Heer Kloots designed the research. BC conducted all experiments, analyzed the results and wrote this chapter.

ACKNOWLEDGEMENTS This interlude sprang from a spontaneous and unreasonably productive brainstorm with Marianne de Heer Kloots—thanks, Marianne. I suggested some of the ideas in this chapter to Axel Bremer and Ioanna Sanida in 2019, and I would also like to thank them for giving them a first try. Finally, I would also like to thank Jaap Jumelet and Oskar van der Wal for their advice on the implementation and Jelle Zuidema for his feedback on an earlier version of this chapter.

CHAPTER

6

Article

Cosine contours

Melodic contour is central to our ability to perceive and produce music. We propose to represent melodic contours as a combination of cosine functions using the discrete cosine transform. The motivation for this approach is twofold: (1) it approximates a maximally informative contour representation (capturing most of the variation in as few dimensions as possible), but (2) it is nevertheless independent of the specifics of the datasets for which it is used. We consider the relationship with principal component analysis, which only meets the first of these requirements. Theoretically, the principal components of a repertoire of random walks are known to be cosines. We find, empirically, that the principal components of melodies also closely approximate cosines in multiple musical traditions. We demonstrate the usefulness of the proposed representation by analyzing contours at three levels (complete songs, melodic phrases, and motifs) across multiple traditions in three small case studies.

6.1 Melodic contour

Humans are born with a remarkable sensitivity to melodic contour. This is dramatically illustrated when newborns cry: the cries of German babies tend to go down in pitch, but those of French babies go up, even if falling contours are physiologically easier to produce (Mampe et al., 2009). By imitating the intonation patterns of their mothers' language, babies take the first steps towards a spoken language, guided by the exaggerated pitch contours of infant-directed speech (Wermke et al., 2021). Contour perception remains central to speech later in life for intonation or even word distinctions—but it is also a key ingredient of human musicality (Honing et al., 2015).

In the musical domain, melodic contour describes the overall shape of a melody while abstracting away from the particular pitches and precise rhythms. Dowling (1978) famously argued that contour plays an important role in musical memory. He suggests that melodies are remembered as two independent parts: a scale and a contour. On this account, a scale functions as a ladder "on which the ups and downs of the contour were hung." Indeed, when listening to novel melodies, contours appear to stand out more than the exact intervals and influence the perceived similarity of melodies (Schmuckler, 2016).

Given the importance of contour, this chapter asks for the optimal way to describe the shape of a melody. How can we capture as much of the variability in melodic contours as efficiently as possible? One approach would use a *principal component analysis* (PCA). We empirically show that the principal components of melodies do not have arbitrary shapes but closely approximate cosines. We relate this observation to theoretical results explaining how the covariance structure of certain random walks yields sinusoidal principal components.

Our findings motivate a new contour representation that describes melodic shape as a combination of cosine functions. The proposed *cosine contour* space closely approximates the optimal solution provided by PCA but offers several benefits, such as being data independent. The central argument for this representation is theoretical, and we leave a systematic comparison of contour representations for future work. Instead, we discuss three case studies demonstrating the usefulness of cosine contours.

6.2 Contour representations

Melodic contour has been characterized in many different ways. First, ethnomusicologists and composers have used *contour typologies* that describe a small set of contour types. Huron (1996), for example, distinguished nine types of contours by comparing the initial and final pitches to the average pitch on the middle part of a melody. We return to such discrete descriptions of melodic contour in chapter 8. Second, there are combinatorial models of contour that rely on the relative ordering of all pairs of notes in a melody, summarized in a matrix. We will not further discuss those models here since these expand rather than reduce the representation, break the linearity of the melody, and are sensitive to local changes (Müllensiefen & Wiggins, 2012).

Instead, we focus on more direct representations, such as—third—representing contour by a simple sequence of pitches or intervals. Melodies extracted from audio are commonly represented this way. Various contour features, such as the range or pitch deviation, can be derived from this and have successfully been used in classification tasks (Bittner et al., 2017; Bittner et al., 2015; Panteli et al., 2017; Salamon et al., 2012). As we have seen in section 2.4, melodic contours in symbolic data can also be represented in this way by using *step curves* that interpolate the notes (Müllensiefen & Wiggins, 2012; Steinbeck, 1982). This has been illustrated by the black line in Figure 6.1B.

Fourth, several contour representations can be directly derived from step curves. *Parsons* code is a drastic simplification that discards interval sizes and note durations and only considers the direction of movement from one note to the next: up, down, or level (Parsons, 1975). We have encountered step curves and Parsons code in chapter 4. Variants between these two extremes have also been used by distinguishing various classes of jump sizes (Müllensiefen & Frieler, 2004). Another related class of representations only considers salient notes, such as maxima and minima (Adams, 1976; Densmore, 1918; Salamon et al., 2012; Steinbeck, 1982). This often requires special handling of ornaments (Müllensiefen & Wiggins, 2012), possibly tailored to the repertoire.

Fifth, one can describe melodic contour by *fitting a function*. For example, Müllensiefen and Wiggins (2012) fit polynomial functions to the step curve and represent the contour using the coefficients. The degree of the polynomial is chosen per phrase, using the Bayesian information criterion (BIC) to avoid overfitting. Polynomial coefficients are quite difficult to interpret, however: they change drastically when the degree changes and can also be sensitive to changes in the data, especially when the polynomials are not orthogonal and introduce correlations between the coefficients (collinearity).

Sixth, instead of fitting a function to the contour, one can also *decompose* the contour and express it as a sum of (orthogonal) basis functions. Velarde et al. (2016) have used *Haar wavelets* as basis functions in musical pattern discovery. The step-like shapes of those wavelets are well suited to describe particular melodic patterns but make them less suited for describing the overall contour. An alternative basis of sinusoidal functions is implicit in Schmuckler's use of a Fourier analysis to represent melodic contour (Schmuckler, 1999).

The contour representation we propose in this chapter is similar in spirit and will decompose the contour using cosines as basis functions. This is motivated by a curious regularity observed in the principal components of melodic phrases while working on the case study on the melodic arch

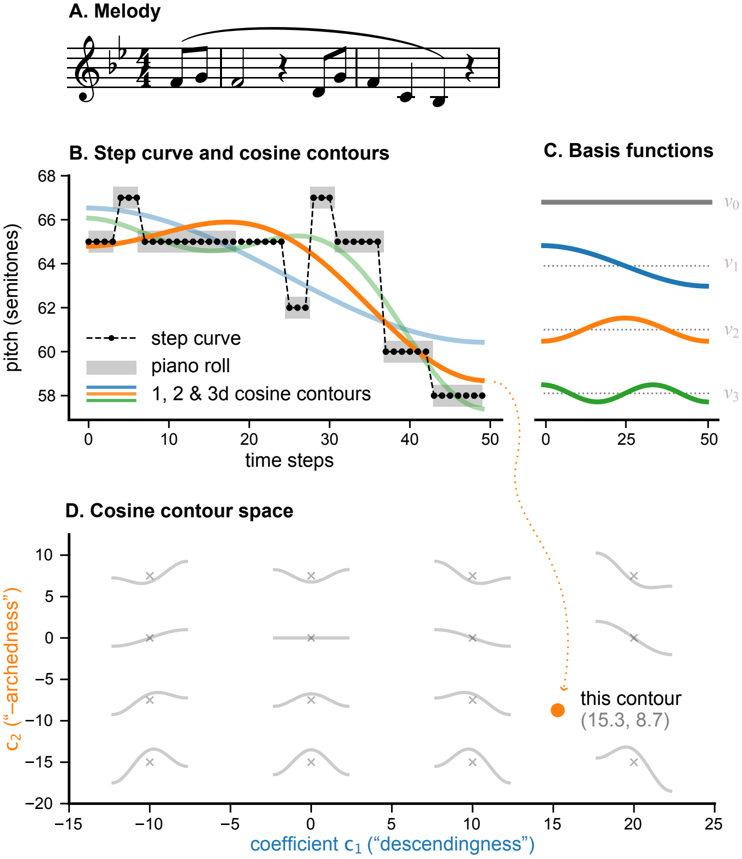


FIGURE 6.1 – Cosine contours represent a melodic contour as a combination of cosine functions. (A) A short melodic phrase illustrates this. **(B)** A piano roll is interpolated to obtain a fixed-length vector of MIDI pitches (black curve). This vector is approximated using a discrete cosine transform (colored curves). Increasing the dimensionality (from, say, the blue to the green line) improves the approximation. **(C)** The basis functions correspond to simple shapes. This makes the cosine contour space interpretable, as illustrated in **(D)** for the first two dimensions. Every point in this space defines a contour shape, varying in what we call the *descendingness* and *archedness*. The orange dot represents the orange contour from **(B)**.

hypothesis in chapter 2. Before we can explain that regularity, we have to introduce the data.

6.3 Data

In this chapter, we analyze contours from musical scores at multiple levels of description, from complete songs to phrases and melodic motifs, as well as two random baselines.

MOTIFS All segmentation levels are readily available in the two plainchant corpora introduced in chapter 2: Cantus Corpus and GregoBase Corpus. The close connection between music and text in chant suggests a natural grouping of the notes into words or syllables, and the notation moreover suggests an even smaller grouping into *neumes*. The motifs corresponding to neumes, syllables, or words are all extracted from Cantus Corpus (v0.2), using only the two most frequent genres: antiphons and responsories.

PHRASES Phrase boundaries are not available in the Cantus Corpus, and we thus extract phrases from GregoBase Corpus (v0.3). As also explained in section 2.4, the chant notation used by GregoBase includes explicit breathing marks known as *pausas*, which we can interpret as phrase boundaries. Phrase markings are also included in the Essen Folksong Collection (Schaffrath, 1995). We additionally analyze phrases from German and Chinese folksongs and focus the discussion on the two largest subsets, which are also included in Catafolk (see chapter 3): *Erk* (9782 contours) and *Han* (7601 contours).

SONGS Finally, at the level of complete songs, we look at music from the Lakota people (also known as the Teton Sioux) made available in the *Densmore Collection* (Densmore, 1918; Shanahan & Shanahan, 2014). Analyses of several other corpora from the *Essen* and *Densmore* collections are only included in supplement B2 to simplify the discussion in the main text.

RANDOM SEGMENTS Finally, we consider two random baselines: random segments of melodies and synthetic phrases generated by a random walk. While the latter are entirely generated, the *random segments* consist of actual melodic material. The segments are obtained by randomly slicing a melody into approximately phrase-length parts so that their boundaries will usually not overlap with actual phrase boundaries (see page 16 for details).

SYNTHETIC PHRASES Next, to generate the *synthetic phrases*, we draw the number of notes K from a (truncated) Poisson distribution to roughly approximate the length distribution of phrases.[1] Then we draw an initial pitch x_0 uniformly between 60 and 85 in MIDI pitch space. In every next step, we draw a step size r_k from a shifted Binomial distribution with mean zero[2] and let the next pitch be $x_k = x_{k-1} + r_k$. This results in small, approximately normally distributed step sizes. This process yields a sequence of pitches $(x_0, \dots, x_{K-1})$: a synthetic phrase.

PITCH SEQUENCES We convert all melodic material—songs, (synthetic) phrases, segments, and motifs—to fixed-length pitch sequences, just as in section 2.4. To do so, we extract the note onsets and pitches (in quarter notes and MIDI semitones respectively) and then interpolate a step function through these points. We sample $N = 100$ equally spaced pitches from the step function and collect those in a *pitch sequence* $\mathbf{x} = (x_0, \dots, x_{N-1})$, as illustrated in Figure 6.1B. These vectors form the primary data analyzed in this chapter. Unlike section 2.4, we do *not* center the contours to have a mean pitch of 0. This is sometimes done to make contours transposition invariant and more directly comparable (Savage et al., 2017; Velarde et al., 2016), but the proposed representation elegantly resolves this problem without requiring centering.

[2] We constrain the step sizes to lie between −12 and +12, meaning that jumps cannot exceed an octave.

[1] We truncate the distribution so that $K \geq 3$ and use $\lambda = 12$. For more details, see supplement B1.

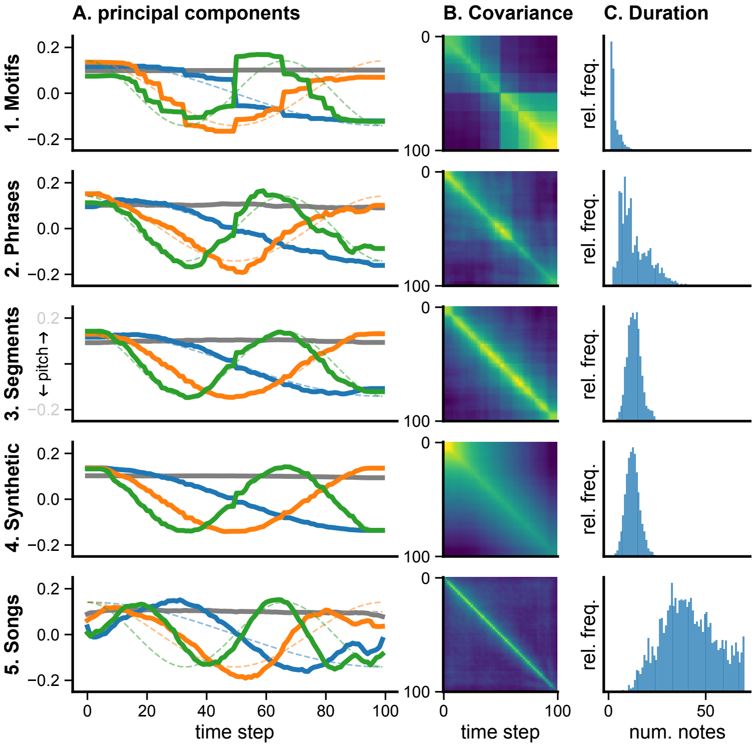


FIGURE 6.2 – Principal components of contours are roughly cosine shaped across different levels. (A) shows the PCs as solid lines and the cosines as dashed ones. This is a result of the particular structure of the covariance matrix **(B)**: matrices of this type have Fourier basis functions as their eigenvectors. This is clearest for phrases **(2)** or random segments from melodies **(3)**. Crucially, we see the same effect for synthetic phrases, generated by random walks **(4)**. For complete songs **(5)**, the effect is less clear, probably due to differences in typical length **(C)** and data size.

The basic representation makes several common assumptions (e.g., Savage et al., 2017; Tierney et al., 2011; Velarde et al., 2016). First, we ignored all rests. Second, we normalize the duration of all contours: both 3-note motifs and 30-note songs are represented by vectors of 100 pitches. Of course, the relative durations within that melody are retained, so we should still find simpler contours in shorter fragments. Third, we assume Euclidean distances between contours. Our analyses require that all contours are embedded in a vector space. Using more sophisticated measures such as dynamic time-warping distance would require us to reconstruct a space (e.g., using multidimensional scaling), making all analyses less transparent.

6.4 Principal components of contours

In the introduction, we asked for the optimal representation that efficiently describes most variability in melodic contour. A principal component analysis (PCA) would be an obvious starting point. The goal of PCA is to find a set of orthogonal axes, the *principal components*, along which one finds most of the variance in the dataset. The axes are described by vectors from the same space as the original data. And so, if we take a dataset of pitch sequences, the principal components will be N-dimensional vectors that can themselves be interpreted as pitch sequences. We use this in Figure 6.2A to visualize the first four principal components of motifs, phrases, random segments, and complete songs, all from the plainchant corpora.[3] Similar results with German and Chinese folksongs can be found in supplement B2.

We find that the principal components are highly similar across most datasets and correspond to well-known contour shapes: descending, convex, and—perhaps—undulating. This can be seen in phrases and random segments. The effect is weaker for complete songs, especially in smaller datasets (see the supplement B2). Besides small data sizes, the fact that songs are much longer also plays a role (see Figure 6.2C). Interestingly, the pattern is even more evident for the synthetic phrases. Since these are generated by a random walk, this suggests that the phenomenon has a mathematical explanation.

To give that explanation, we must first describe PCA more formally. We consider a collection of M contour vectors $\mathbf{x}_m$ of length N. Denote the sample mean by $\bar{\mathbf{x}} = \frac{1}{M}\sum_m \mathbf{x}_m$ and the centered data by $\hat{\mathbf{x}}_m = \mathbf{x}_m - \bar{\mathbf{x}}$. The first principal component of the dataset is then defined as a normalized vector $\mathbf{u}_1 \in \mathbb{R}^D$ for which the projected data $\{\mathbf{u}_1^T\mathbf{x}_m : 1 \leq m \leq M\}$ has maximal variance. It can be shown (e.g., Jolliffe, 2002) that this is the case when $\mathbf{u}_1$ is an eigenvector corresponding to the largest eigenvalue λ_1 of the covariance matrix

$$\mathbf{S} = \frac{1}{M}\sum_{m=1}^{M}(\mathbf{x}_m - \bar{\mathbf{x}})(\mathbf{x}_m - \bar{\mathbf{x}})^T, \tag{6.1}$$

so that $\mathbf{S}\mathbf{u}_1 = \lambda_1\mathbf{u}_1$. It follows that the projected variance is given by λ_1, the largest eigenvalue. The other principal components similarly emerge as the other eigenvectors of the covariance matrix.

The covariance matrices (Figure 6.2B) for both random walks and our empirical data have a particular structure: they *roughly* resemble *Toeplitz matrices*, which have fixed values along each of their diagonals. Such covariance structures are frequently encountered in spatial or temporal data when the covariance decreases with the distance between the points (Antognini & Sohl-Dickstein, 2018; Gray, 2006; Novembre & Stephens, 2008). That appears to be the case for the contours: there is a higher correlation between successive pitches and a lower correlation between distant pitches. As a result, the higher covariances are concentrated along the diagonal. Again, this is clearest for the phrases and random segments. We see some deviations for motifs: two blocks in the covariance matrix and corresponding jumps halfway through the principal components. This is easily explained by the fact that motifs often span only two notes. In that case, all pitches in the first half of the contour are then perfectly correlated, as are pitches in the final half. Crucially, despite such deviations from a perfect Toeplitz structure, the principal components are still well-approximated by cosines.

If you let a Toeplitz matrix grow in size, it asymptotically tends towards a *circulant* matrix, preserving properties such as eigenvalues and eigenvectors along the way (Gray, 2006). Circulant matrices have exactly the same

[3] The *motifs* are responsory syllables from Cantus Corpus, *phrases* are antiphon phrases from the GregoBase Corpus, and the *songs* are song contours from GregoBase Corpus.

values in every row but are rotated one step to the right with respect to the previous row. The surprising result is that all circulant matrices have the same eigenvectors: basis vectors of the discrete Fourier transform. For real and symmetric matrices, like covariance matrices, this results in cosine-shaped eigenvectors of increasing frequency—precisely what we see in Figure 6.2. We discuss all of this in more detail in the supplement B2. In sum, because of a Toeplitz-like covariance structure, the principal components of melodic contours will tend to look like cosine functions.

6.5 Cosine contours

Next, we turn this observation, and its explanation, into a proposal for a new contour representation. The idea is to approximate the principal components by cosine functions and then project the contours on those first few cosines to obtain a low-dimensional representation. This is exactly equivalent to taking a *discrete cosine transform* (DCT) of the contour (Ahmed et al., 1974).

Formally, consider a collection of contours of length N as before. We approximate the k-th principal component $\mathbf{u}_k$ by a vector $\mathbf{v}_k$ of the form $\big(v_k(0), \dots, v_k(N-1)\big)$ whose entries are given by the cosine function[4]

$$v_k(n) = \alpha_k \cdot \cos \frac{\pi(2n+1)k}{2N}. \tag{6.2}$$

Here $\alpha_0 = 1/\sqrt{N}$ and $\alpha_k = \sqrt{2/N}$ for $k \geq 1$ are normalizing constants ensuring that $\mathbf{v}_k$ has unit norm. The projection of a contour $\mathbf{x} = (x_0, \dots, x_{N-1})$ on $\mathbf{v}_k$ is then given by the inner product $c_k = \mathbf{v}_k^T \mathbf{x}$. Expanding this gives the usual definition of the discrete cosine transform (DCT-II):

$$c_k = \sum_{n=0}^{N-1} x_n \alpha_k \cos \frac{\pi(2n+1)k}{2N}. \tag{6.3}$$

Conversely, the contour can be reconstructed from the coefficients c_k using the inverse transform $x_n = \sum_{k=0}^{N-1} c_k v_k(n)$. Using only $D < N$ coefficients, we define our low-dimensional *cosine contour representation* as $C_D(\mathbf{x}) = (c_1, \dots, c_D)$. Note that we deliberately discard c_0. This coefficient corresponds to a flat line and describes the overall pitch height of a contour: precisely what we need to get rid of to make the contour transposition invariant. In this way, we resolve the centering of contours discussed above.

Why use this representation instead of principal components? Indeed, a principal component projection, also known as the *Karhunen-Loève transform* in this context, is optimal in several ways (Ahmed et al., 1974; Rao & Yip, 1990). Not only does it decorrelate the data, but it also packs most variance in the first few transform coefficients (sometimes called *energy compaction*) and minimizes the reconstruction error when using only a few coefficients. However, the transformation depends on the data. Con-

[4] These basis functions correspond to the most popular version of the discrete cosine transform, DCT-II, for which fast implementations are widely available; others would have been possible (Strang, 1999).

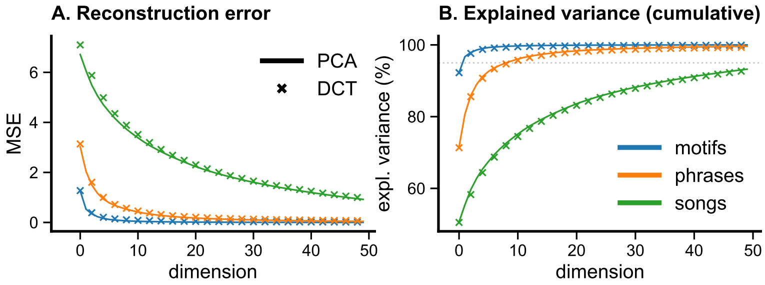


FIGURE 6.3 – DCT approximates PCA. The latter is the optimal transform, in terms of the reconstruction error **(A)** and the explained variance ratio **(B)**. The reconstruction error is the mean squared error between a contour and a lower dimensional reconstruction. Note that data corresponds to Figure 6.2 and that we did *not* discard the first component c_0 of the DCT in this figure.

cretely, the principal components of German phrase contours differ from Chinese ones. Any choice for using one of the two is arbitrary. In contrast, the DCT is a principled, neutral solution—that approximates the optimal transform. In fact, the DCT was initially introduced for similar reasons (Ahmed et al., 1974) and was then found to empirically approximate PCA well in domains ranging from image to audio (Rao & Yip, 1990). The current results suggest that the same applies to melodies.

6.6 Evaluation and case studies

We evaluate the proposed contour representation by comparing it to a principal component transformation to demonstrate that it is close to the optimum. We further designed three case studies to illustrate its usefulness at the levels of (1) song, (2) phrases, and (3) motifs. The case studies show that the representation is musicologically meaningful, as it allows visualization of variation (1), a quantitative evaluation of constraints on variation (2), and accurate classification into traditional categories (3). For simplicity, we only look at two-dimensional representations in these case studies, but higher dimensions may be useful in practice.

OPTIMALITY To empirically verify the claim that the DCT approximates the optimal PCA transform, we compute the reconstruction error and the explained variance ratio using the same data as before. The reconstruction error is measured as the mean square error between a contour and its D-dimensional reconstruction, using either the principal components (PCA) or cosines (DCT) as basis functions (so for $D = N$, the reconstruction is guaranteed to be perfect). Figure 6.3A shows that the reconstruction errors of DCT closely approximate that of PCA. The error rapidly decreases for the shorter contours (motifs and phrases), indicating that low-dimensional representations are already effective. Indeed, to explain 95% of the variance using cosine contours, you need one dimension for motifs, nine for phrases, and 61 for songs—this is sometimes called the *effective dimensionality* (Moore et al., 2018).[5]

CASE STUDY 1: VISUALIZING DIFFERENT TRADITIONS Low-dimensional representations of song contours are not likely to be very informative, yet certain

[5] However, note that Moore et al. (2018) show that high-dimensional random walks can falsely appear to have a low effective dimensionality.

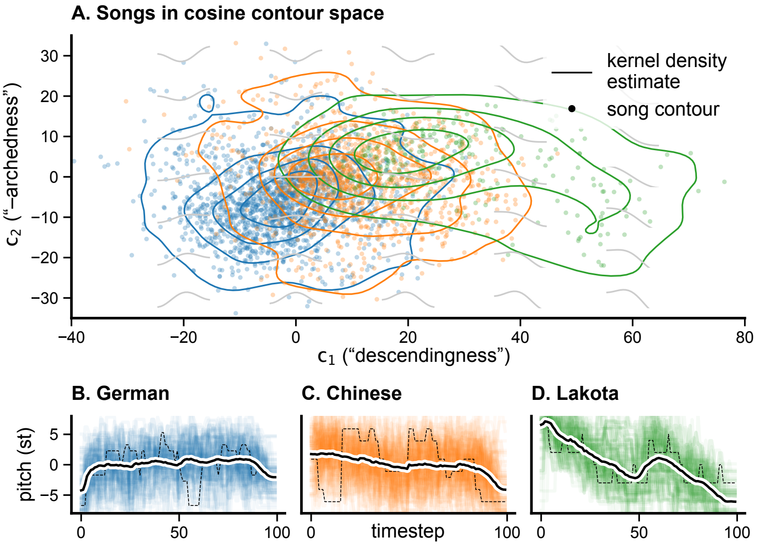


FIGURE 6.4 – Songs of three cultures represented in the cosine contour space. In a 2D cosine contour space **(A)**, every point represents a contour, as illustrated by a grid of gray contours in the background. The first coefficient c_1 measures 'descendingness' (horizontally), and $-c_2$ measures 'archedness'. Three datasets show substantial variability, as best seen from the colored lines that estimate their density: Lakota songs are more strongly descending than German ones. The average of all contours in a tradition **(B–D)** also illustrates this. Thick black lines show that average, while dashed lines highlight a single contour.

traditions can be somewhat distinguished in just two dimensions. Figure 6.4 shows song contours from German, Chinese, and Lakota songs. We observe that the first component c_1 of a cosine representation roughly measures the *descendingness* of the contour, and, similarly, that $-1 \cdot c_2$ measures the *archedness*. Lakota songs often have a strongly descending overall shape (subplot D), which is reflected in the cosine contours having relatively high descendingness. Similarly, German songs appear more arch-like than songs from the other two traditions, translating into lower values of c_2.

CASE STUDY 2: THE MELODIC ARCH HYPOTHESIS In a second case study, we show that cosine contours provide a simple way to test the melodic arch hypothesis (Huron, 1996). Recall that the hypothesis claims that *phrases* tend to be arch-shaped or descending (see also Figure 6.5A and B). This can be reformulated as claiming that c_1 (descendingness) and $-c_2$ (archedness) are larger for phrases than for random segments of the melodies. Comparing Chinese and German phrases, we find that all are significantly ($p \ll 0.001$) more descending and arched than the corresponding random segments (see Figure 6.5C and D). This demonstrates that the coefficients of the cosine contour representation are musicologically meaningful.

CASE STUDY 3: MODE CLASSIFICATION Finally, we revisit the study on mode classification in plainchant from chapter 4. In that chapter, we suggest that the mode of Gregorian chant can be predicted from contours alone, in that case using a Parsons code contour representation. We represented chants with tf–idf vectors of weighted motif frequencies, where motifs were obtained by segmenting chants in various ways. We repeat these experiments using a two-dimensional cosine representation for the motifs. There is one technical problem: whereas cosine contours are continuous,

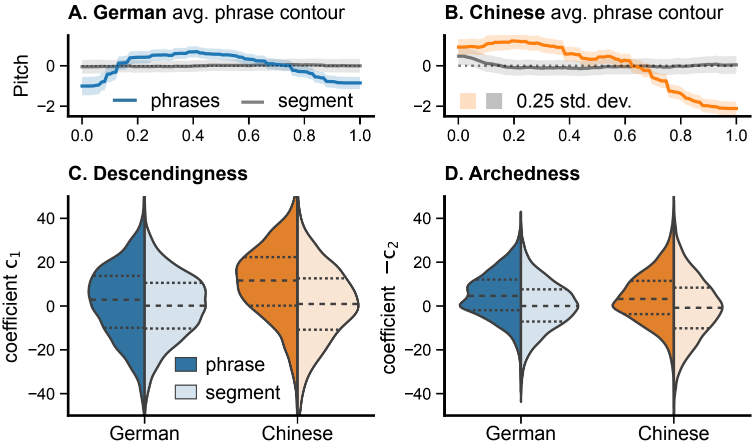
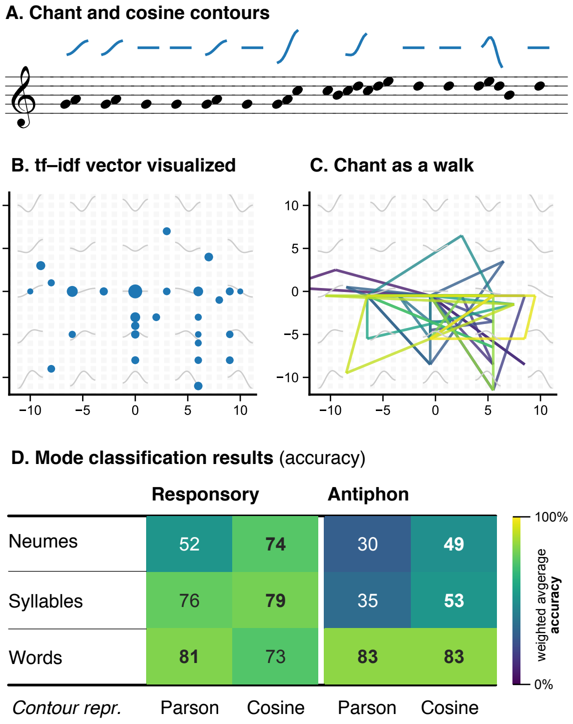


FIGURE 6.5 – Phrases of German and Chinese songs tend to be descending and arched. This becomes clear when comparing the average contours to random segments from the same melodies **(A-B)**. To quantify this tendency, we compare the first **(C)** and second **(D)** coefficients of their cosine representations, which can be used to measure descendingness (c_1) and archedness ($-c_2$) respectively. Consistent with the melodic arch hypothesis, we indeed find that both these quantities are higher in phrases than in random segments.

the tf–idf model requires a discrete vocabulary of motifs. Therefore, we discretize the cosine contour space to a grid and effectively treat every chant as a sequence of grid cells (Figure 6.6c). All in all, this introduces two new parameters to the experiment: the dimensionality of the cosine contour and the resolution of the grid. In this case study, we do not tune these parameters and focus on two-dimensional contours, discretized to a grid between −20 and 20 with a grid size of 1. For ease of reading, the Figure 6.6b shows the grid only from −10 to 10. The results are summarized in Figure 6.6d. We see an interesting pattern: the cosine contours outperform the original results for small motifs such as neumes and syllables but not for words, which form much longer motifs. This makes sense: two-dimensional cosine contours are a relatively crude approximation of those longer contours but may reasonably approximate short motifs.

6.7 Discussion and conclusions

In this chapter, we proposed a novel representation for melodies using the discrete cosine transform: cosine contours. Observing that the principal components of melodies tend to be shaped like cosines, this representation approximates the optimal representation in the sense that it packs most variance in a few dimensions. Cosine contours meet several other desiderata for contour representations. First, the cosine representation is easily interpretable, as it presents contours as a linear combination of cosine functions with intuitive shapes. Second, by changing the dimensionality, the contour's abstraction level can be varied, allowing for an arbitrarily small reconstruction error by including more and more dimensions. Third, this representation allows one to map contours at multiple levels—from motifs to songs—to one shared space. The cosine representation thus creates a common ground for comparing contours across traditions and levels. That is possible as, fourth, the representation is independent of the data and, in that sense, culturally neutral.



	Responsory		Antiphon	
Neumes	52	**74**	30	**49**
Syllables	76	**79**	35	**53**
Words	**81**	73	**83**	**83**
Contour repr.	Parson	Cosine	Parson	Cosine

FIGURE 6.6 – Motifs used for mode classification in Gregorian chant. (A) A chant is segmented into motifs derived from the notation (neumes) or lyrics (syllables, words). The blue curves show the two-dimensional cosine contours for those motifs. **(B)** We discretize the contour space and represent the chant as a vector of tf–idf weighed motif frequencies ('grid cell frequencies'). Dots illustrate the nonzero entries of this vector for the chant shown above. **(C)** The chant is now a walk through contour space, but our bag of motifs ignores the order. **(D)** Using these vectors to classify mode, for the smaller motifs neumes and syllables, we outperform our previous study (chapter 4), which used a string-based contour representation (Parsons code).

The observation that principal components of spatial and temporal data can have sinusoidal shapes is not novel but does not appear widely known. Indeed, the sinusoidal shapes have been interpreted as genuine effects rather than mathematical artifacts. For example, one study interpreted gradients in the principal components of human genetic variation worldwide as evidence for certain migration events in human history (Cavalli-Sforza et al., 1993). Closer inspection revealed that those gradients were sinusoidal artifacts analogous to those reported in the present paper (Novembre & Stephens, 2008). Closer to MIR, it has been observed that the training trajectories of deep neural networks have sinusoidal principal components (Lorch, 2016) for the same reason. Again, a detailed analysis (Antognini & Sohl-Dickstein, 2018) revealed these were artifacts but accurately reflected the behavior of high-dimensional random walks (Antognini & Sohl-Dickstein, 2018; Moore et al., 2018). We hope this paper helps to increase awareness of this phenomenon.

The present work only begins to explore this new contour representation and raises many further questions. One particularly promising possibility is the application to audio data. In this chapter, we only explored symbolic data, but the proposed representation equally applies to acoustic data. One application we hope to explore further is the analysis of speech intonation using the cosine contour representation. Another interesting case would be the analysis of folk song recordings. Folk song researchers have, in various ways, relied on contour to organize repertoires

(Adams, 1976), and one could investigate whether that categorization can be partly automated using cosine contours. Finally, various contour typologies have been used in cross-cultural comparisons (Adams, 1976; Huron, 1996; Kelkar et al., 2018; Savage et al., 2015; Savage et al., 2012) but have not been systematically evaluated. The present chapter is the starting point for such a comparison, which we take up in chapter 8.

SUPPLEMENTS **B1** Random walk baseline · **B2** Analyses of other datasets · **B3** Mathematical background

DATA AND CODE Data and code are available at github.com/bacor/cosine-contours

REFERENCE This chapter is based on Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2021c). Cosine contours: A multipurpose representation for melodies. *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 135–142. DOI 10/gpdr3w

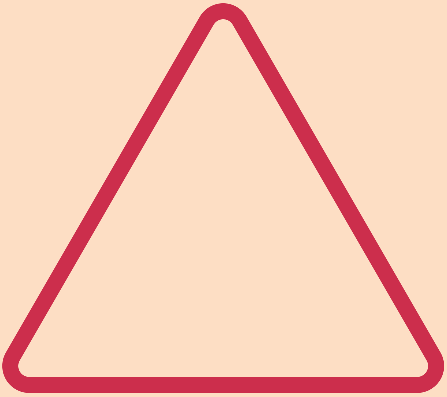
AUTHOR CONTRIBUTIONS BC designed and conducted the research and drafted the original conference paper. JAB and WZ supervised the research and edited the manuscript.

ACKNOWLEDGEMENTS We want to thank Henkjan Honing and Marianne de Heer Kloots for their feedback on the manuscript of the original publication. We would also like to thank the four anonymous reviewers for their careful reviews; we have tried to address all your comments.

CHAPTER

7

Interlude



Rhythm triangles

And now for something completely different: rhythm. Over the last few years, I have often discussed rhythm in a course on the evolution of language and music. Students with little musical training understandably struggle with scales or chords and apparently find rhythm easier to grasp. And if singing is scary, students are happy to clap along, to the point that they once performed something like Steve Reich's *Clapping music.*[1] In one particular lecture on rhythm, I would often play music in different meters and ask them to first clap and then *count* along. Apparently, counting music is not something people normally do. But even if students didn't manage, they do usually recognize when something is wrong, like counting a waltz in four. And that exercise is not only fun, but it explains a musical structure (meter), illustrates how it can vary, and convinces students that they themselves have metrical expectations. But mostly, all of this would be an upbeat for a story about a fascinating musical space: the rhythm triangle.

[1] I learned this simplified version from Gerben Groeneveld. Two groups basically clap the rhythm of the words "ananas, appel, peer, banaan", with a rest after every word, and after every two cycles, one group injects an extra "druif" at the end.

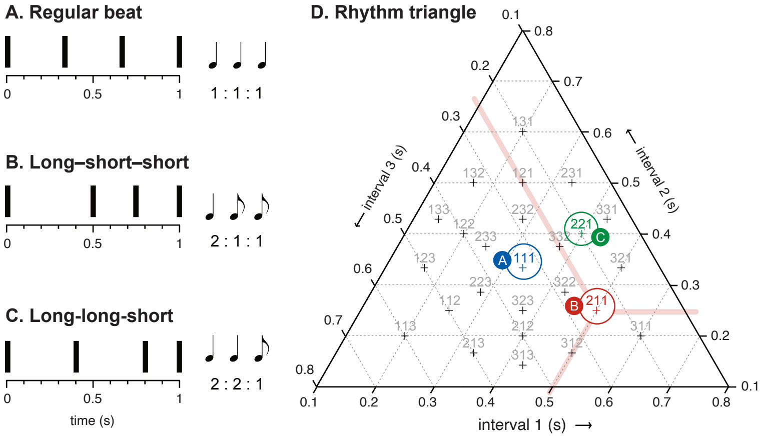


FIGURE 7.1 – **The rhythm triangle.** All rhythm motifs of four onsets and a fixed total duration consist of three intervals and lie in a triangular space **(D)**. The ratio between the intervals determines the motifs position in the triangle. Crosses indicate the positions of small-integer ratio rhythms. All this is illustrated for three examples **(A–C)**. The *isochronous* motif **(A)** with ratios 1:1:1 falls in the very center of the triangle. Red lines indicate how to read the axes for example **(B)**.

7.1 The rhythm triangle

The story starts around 2003, when Peter Desain and Henkjan Honing tried to figure out how listeners perceive rhythms. No drummer, however well trained, will play a rhythm with metronomic precision, *exactly* as a written score suggests. That's probably for the better: the slight deviations from a 'perfect' rendition, the *timing*, often bring a rhythm to life. But to notice the timing one needs a reference, and Desain and Honing reasoned that *categorization* provided such a reference. Categorization occurs when your perception breaks up a continuous phenomenon into a discrete set of chunks or *categories*. But what could be the continuous space of rhythms that we might discretize?

Desain and Honing (2003) decided to look at all the rhythmic motifs that you can make by hitting a drum four times, but in such a way that the time between the first and final stroke is fixed. You can completely specify such a rhythm by giving the time between the first three onsets: the first two *inter-onset intervals*. If the total duration is fixed, the last interval can be computed from the first two. As a result, all such motifs live in a two-dimensional space that happens to be triangular (see Figure 7.1). What determines the rhythm of a motif is not so much the precise duration of the intervals but the *ratios* between the intervals. For example, the motifs with intervals (0.25, 0.5, 0.25) and (2, 4, 2) have a different duration or tempo, but both have the same rhythm: the ratios between the intervals are the same: 1 : 2 : 1. Every point of the *rhythm triangle* corresponds to exactly one such rhythm, and vice versa.

Now, to find out whether listeners categorize this continuous rhythm space, Desain and Honing (2003) played motifs regularly sampled from the space to conservatory students and asked them to write down the rhythms they heard. Their responses were very consistent in some parts of the space and very inconsistent in others. The consistent clumps of the space centered around *small-integer ratio rhythms* like 1 : 1 : 1 or 1 : 1 : 2, and the inconsistent parts were the boundaries in between.

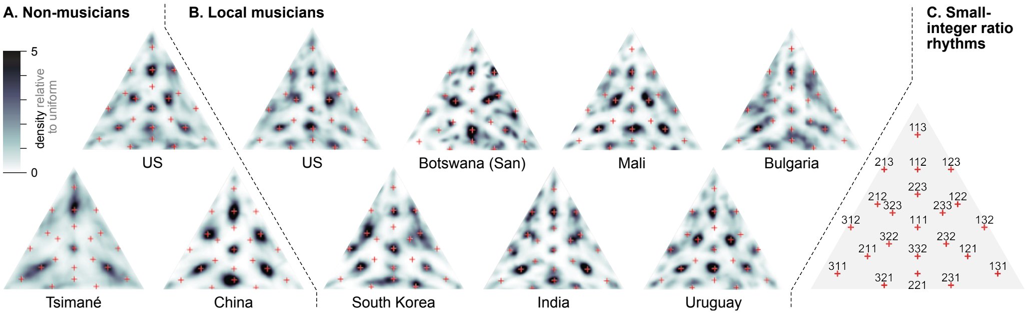


FIGURE 7.2 – Rhythm priors in different subject groups from all over the world (Jacoby et al., 2021). Colors indicate probability density relative to a uniform distribution. **(A)** The prior in three groups of non-musicians compared to seven groups of musicians **(B)**. Small-integer ratio rhythms are highlighted by red crosses and explained in **(C)**. Adapted from figure 2 from Jacoby et al. (2021) (CC-BY 4.0).

All in all, the results suggested that these rhythms were perceived categorically. But just in conservatory students, or also in the general population? Unfortunately, the method could not address such questions: it required highly trained musicians who are used to rhythmical dictations. Around fifteen years later, however, Nori Jacoby and Josh McDermott realized that you don't need participants to notate rhythms. It is enough if they can reproduce a rhythm—again, again, and again.

7.2 Unchaining the triangle

What happens if you pass a sentence around a group of people, each one whispering it into their neighbors' ear? Every six-year-old can tell you: this is the telephone game! Scientists of a particular plumage know this as *iterated learning* or *serial reproduction* and do not consider it a game but an experimental paradigm. Already in the 1930s, Bartlett showed his subjects drawings which they then had to reproduce from memory. Whatever they produced would be presented to the next subject. In this way, an Egyptian hieroglyph of an owl would ten subjects later turn into something like a bin bag before transforming into a cat: a culturally familiar drawing (Xu & Griffiths, 2010).

Serial reproduction appears to change whatever you start with into something that seems highly probable to the subjects, something that reflects their prior expectations.[2] The paradigm has gone through a revival during the last two decades, producing a series of equally fascinating and funny studies. One study had baboons repeat patterns that flashed up on a grid of buttons and eventually found they were passing on Tetris shapes (Claidiere et al., 2014)—a good candidate for an IgNobel price. Other studies have used it as a model of cultural evolution, in particular in language evolution,[3] or, like Jacoby and McDermott (2017), as a tool to measure complex cognitive biases.

To repeat Desain and Honing's study with musically untrained participants, Jacoby and McDermott (2017) played random motifs from the rhythm triangle and asked them to simply tap along. Whatever rhythm the participants produced (averaged over ten reproductions) would be passed on to another participant. In this way, their responses form a walk

[3] This was the topic of my master's thesis (Cornelissen, 2017) and motivated the original proposal for this Ph.D. project.

[2] Such a chain of humans reproducing something can be seen as a so-called Gibbs sampler that estimates the distribution of their prior expectations (Griffiths & Kalish, 2007; Harrison et al., 2020).

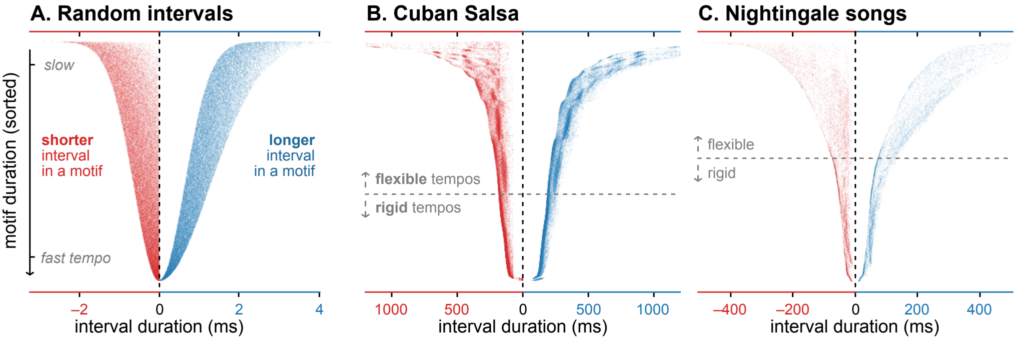


FIGURE 7.3 – **Raster plots.** A raster plot visualizes motifs of two intervals: the shorter interval is plotted on the left, the longer one on the right. The points are then ordered vertically, showing slower motifs at the top and faster ones at the bottom. Roeske et al. (2020) argues that both music **(B)** and nightingale songs **(C)** have categorical rhythms. The flower-like shape of the raster plots is an artifact: it also appears when plotting random intervals **(A)**.

through rhythm space, and this walk tends to gravitate towards expected rhythms, in that way revealing the rhythmical biases of their subjects. The authors repeated this experiment with both North American and Tsimané participants. The Tsimané are a Native American people that live in Bolivia and have had relatively little contact with Western music. The study revealed that their rhythmic prior was strikingly different from those of the North American participants.

By now, Nori Jacoby has gathered a large network of researchers and tested a very diverse group of participants from over fifteen countries (Jacoby et al., 2021). I have reproduced the results from their preprint in Figure 7.2; you can also find the Tsimané and North American participants there. The steady pulse of the *isochronous* motif 1:1:1 is present everywhere, as are 1 : 1 : 2 and its rotations, 1 : 2 : 1, and 2 : 1 : 1. One major source of variation turned out to be the presence of the 3 : 3 : 2 rhythm. This is a very common rhythm in many Sub-Saharan and South American traditions. It is extremely prominent in Malinese dancers but entirely absent in Chinese non-musicians. There you instead see a lot of the 2 : 2 : 1 rhythms (slightly more to the outside of the triangle). In the group of Malineses musicians, you can even find modes that correspond to the complex 7 : 2 : 3 ratio, which the musicians recognized as the rhythm of a popular dance called *Maraka*.

7.3 Flowers

While we told this story in Evolamus, it further unfolded in the Music Cognition Reading Group. In one of our meetings, we were joined by Carel ten Cate, an expert in birdsong. The reading was a paper by Tina Roeske et al. (2020) that analyzed the intervals between syllables in the songs of thrush nightingales and zebra finches, and compared these with inter-

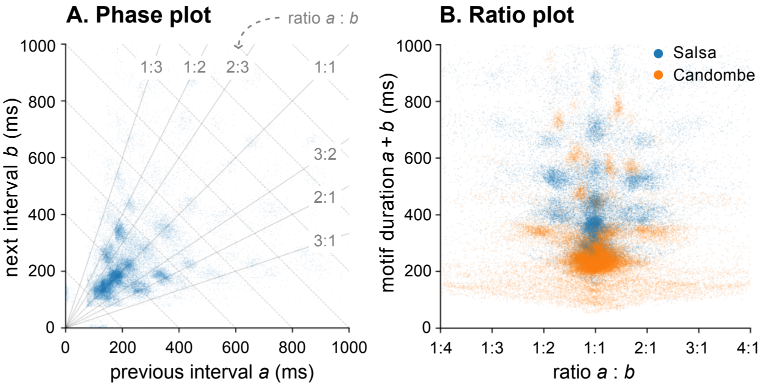


FIGURE 7.4 – Two alternatives for raster plots: phase and ratio plots. In a phase plot **(A)**, the first interval is plotted against the next interval. Blowing up the space near the origin transforms this into a *ratio plot* **(B)**. Both plots are arguably easier to read than raster plots (Figure 7.3B).

vals between notes in recordings from several musical traditions:[4] Indian raga (8 pieces/performances), Cuban salsa (40), Uruguayan candombe (39), Malian Jembe (46), Tunisian stambeli (9), Persian zarb, and Western 'piano' music.[5] The study claimed that *categorical rhythm*—the use of a discrete set of rhythms—is not unique to human music, but also found in the songs of nightingales, although not in zebra finches.

At least as attractive as this claim were the beautiful visualizations. Flower-shaped figures complete with stems and petals somehow visualized rhythmic motifs of two successive intervals. These *raster plots* represented each motif by two points on the same horizontal line: the smaller one of the two intervals is shown on the left, and the larger one on the right. All motifs are then sorted by their total duration, so that slow motifs are on the top, and fast motifs at the bottom of the plot. Beautiful as they may be, their flower-like shape is an artifact. A similar shape appears when you plot random sequences of intervals. What is informative about these plots is the patterning *within* the flowers, which is not only relatively small but also difficult to interpret. It is, for example, hard to identify the lines on which all motifs with the same ratio, say 2 : 1, fall.

The raster plots puzzled the reading group. Wouldn't it be easier, Henkjan Honing wondered, to create a *phase plots*? In such a plot, you show one interval horizontally and the next one vertically (see for example Ravignani et al., 2016). Indeed, phase plots seem to be easier to read (see Figure 7.4), as different diagonal lines now correspond to different ratios, and the further you move from the origin, the longer the duration of the motifs becomes. You could also blow up the space near the origin to transform a phase plot into a *ratio plot*, which shows the duration of a pair vertically against the ratio of its intervals horizontally. The paper also did something like this, but ratio plots are probably still more intuitive.

7.4 Scattered triangles

But instead of two-interval motifs, why not add a third and plot all motifs of three successive intervals in a rhythm triangle? To plot motifs with

[5] The IEMP corpora contain fewer recordings: there are for example only 5 songs in the Cuban salsa and son corpus. They might have counted individual instruments. Their Western piano music consists of performances of Bach's music from the MAESTRO dataset

[4] Most of the music came from the *Interpersonal Entrainment in Music Performance* corpus (IEMP; Clayton et al., 2022), which contains recordings, in various styles, of individual instruments playing in larger ensembles. All of the recordings are publicly available, together with onsets of all instruments.

varying total duration in a rhythm triangle, you would have to normalize the total duration so that you are effectively plotting the ratios between the intervals. The phase plots however showed that duration is clearly a relevant parameter: you find different types of motifs at slow versus high tempos. There is a simple solution: show all motifs in the triangle with a scatter plot, and use a color scale to visualize their duration. On the following pages, I show triangle plots for all of the datasets analyzed by Roeske et al. (2020).

The resulting plots raise all sorts of questions. Do the different clusters in the music corpora correspond to different instruments? Or perhaps to different songs?[6] Or why are some triangles, like Western piano music, asymmetrical? If you rotate it by 60 degrees, you don't get the same pattern. What are the purple clusters in candombe? Why do we see a vertical band in jembe music? And why indeed do we see the same in nightingale song? And what about the zebra finches? Roeske et al. (2020) find no discretization in their songs,[7] while the triangle shows an abundance of discretization, across different timescales. Carel ten Cate suggested that these clusters may correspond to songs of different individuals.[8]

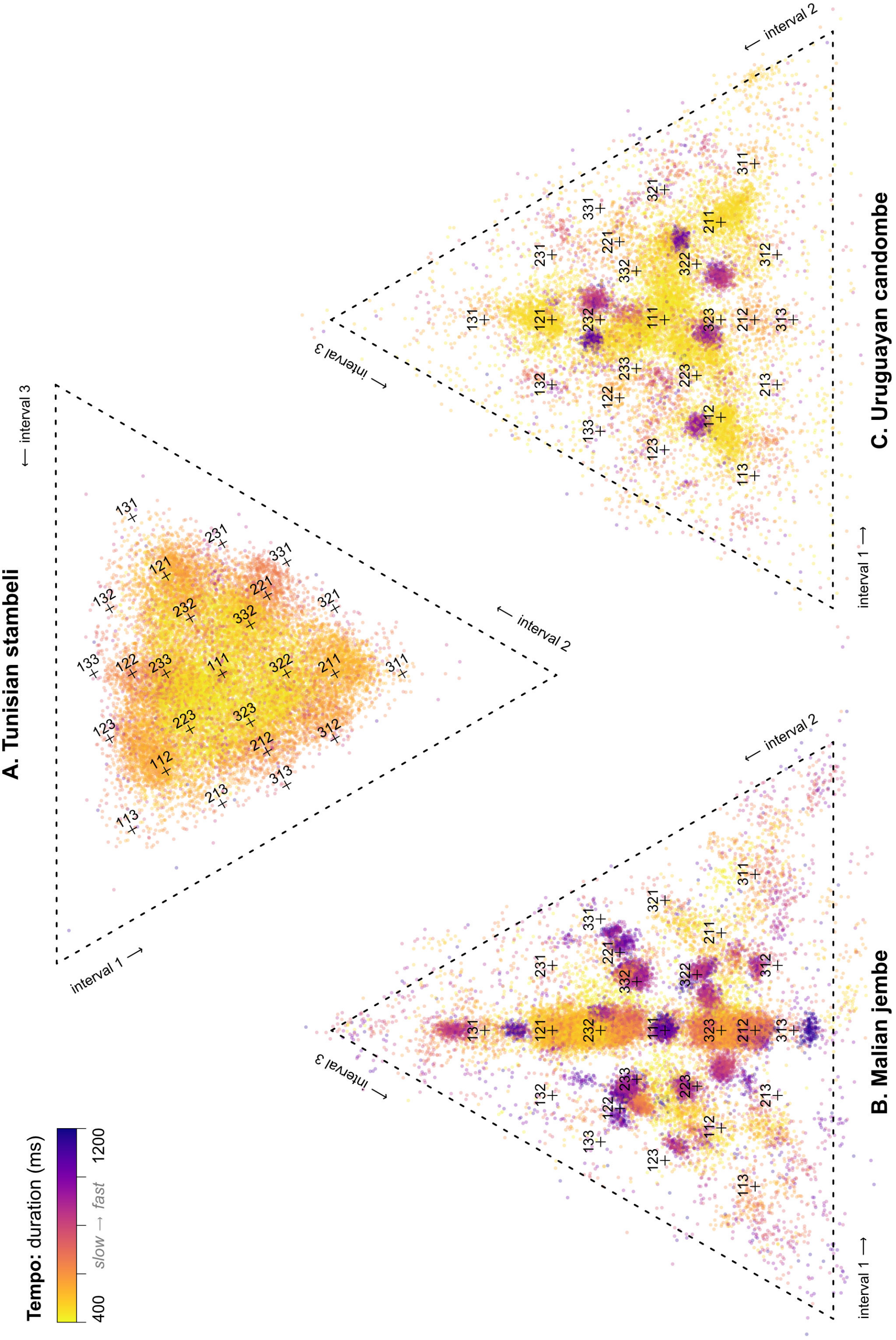
And what about small-integer ratios? This is often cited as a universal tendency, yet Roeske et al. (2020) write that "a statistically significant tendency to produce 1 : 2 ratios was detected only in Western piano and Indian raga performances". And "no significant tendency to produce 1:2 or 1 : 3 ratios was detected in any other music, but in Malian jembe, we found *a significant tendency to avoid* 1 : 3 *rhythms* and favor [non-small-integer ratios] instead" (my emphasis). Indeed, the triangle plots of jembe music contains some clusters that do not correspond to a small-integer ratio, and the same applies to Uruguayan candombe. My aim is not to address all these issues here, but to illustrate how visualization may raise new questions, and hopefully help to address them. And so let's see what else we can plot—surely, humans and nightingales are not the only species with categorical rhythm.

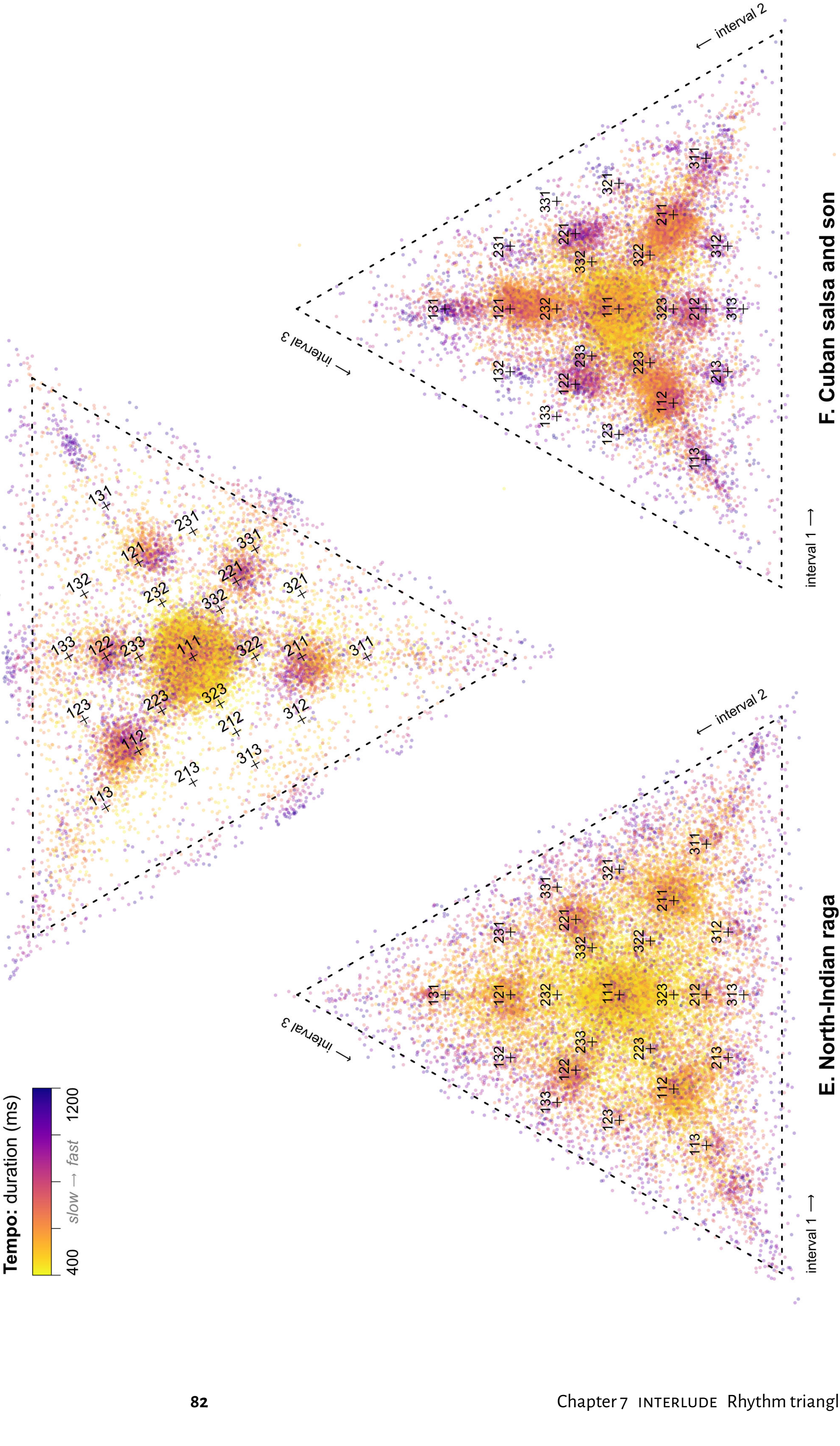
FIGURE 7.5 – **Rhythm triangles show the rhythmic inventories of musical datasets and vocalizations of two bird species (pages 81–84) .** Plots show the data from Roeske et al. (2020). A sequence of intervals is split into overlapping motifs of three intervals each. The ratios between the intervals determine the location in the triangle, and the color indicates the total duration. Darker motifs are slower, lighter ones faster. Small-integer ratio rhythms are indicated by crosses. Most music datasets **(A–G)** show clusters, although these are least pronounced in Persian zarb **(G)**. The zebra finch **(H)** plot shows a very fine yet clear clustering structure, especially when split out in duration ranges of 100ms **(J)**. Nightingale songs **(I)** have less distinct rhythmic clusters, but the motifs are clearly not uniformly distributed either.

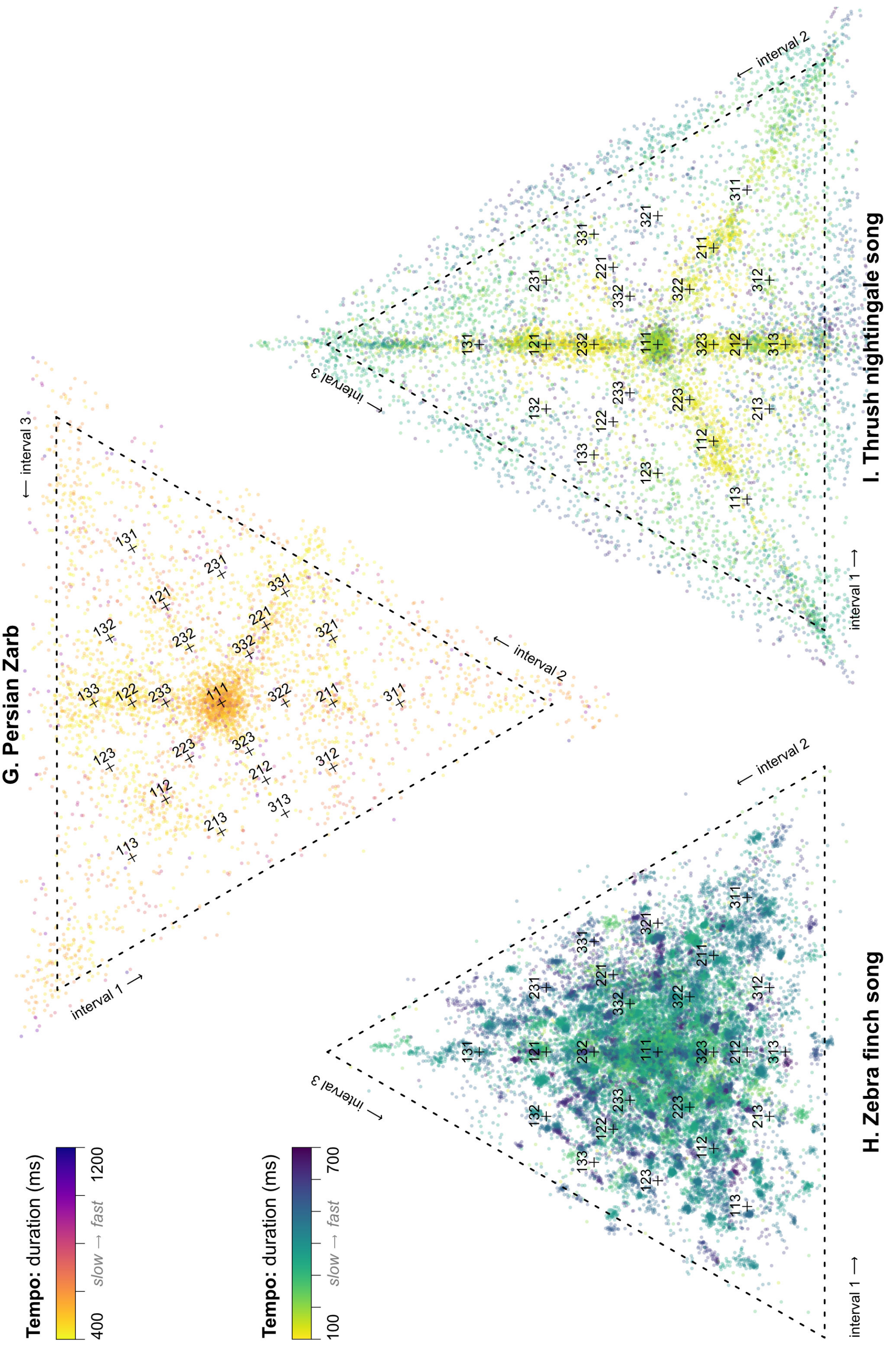
[8] Preliminary analyses of another dataset indeed confirm this.

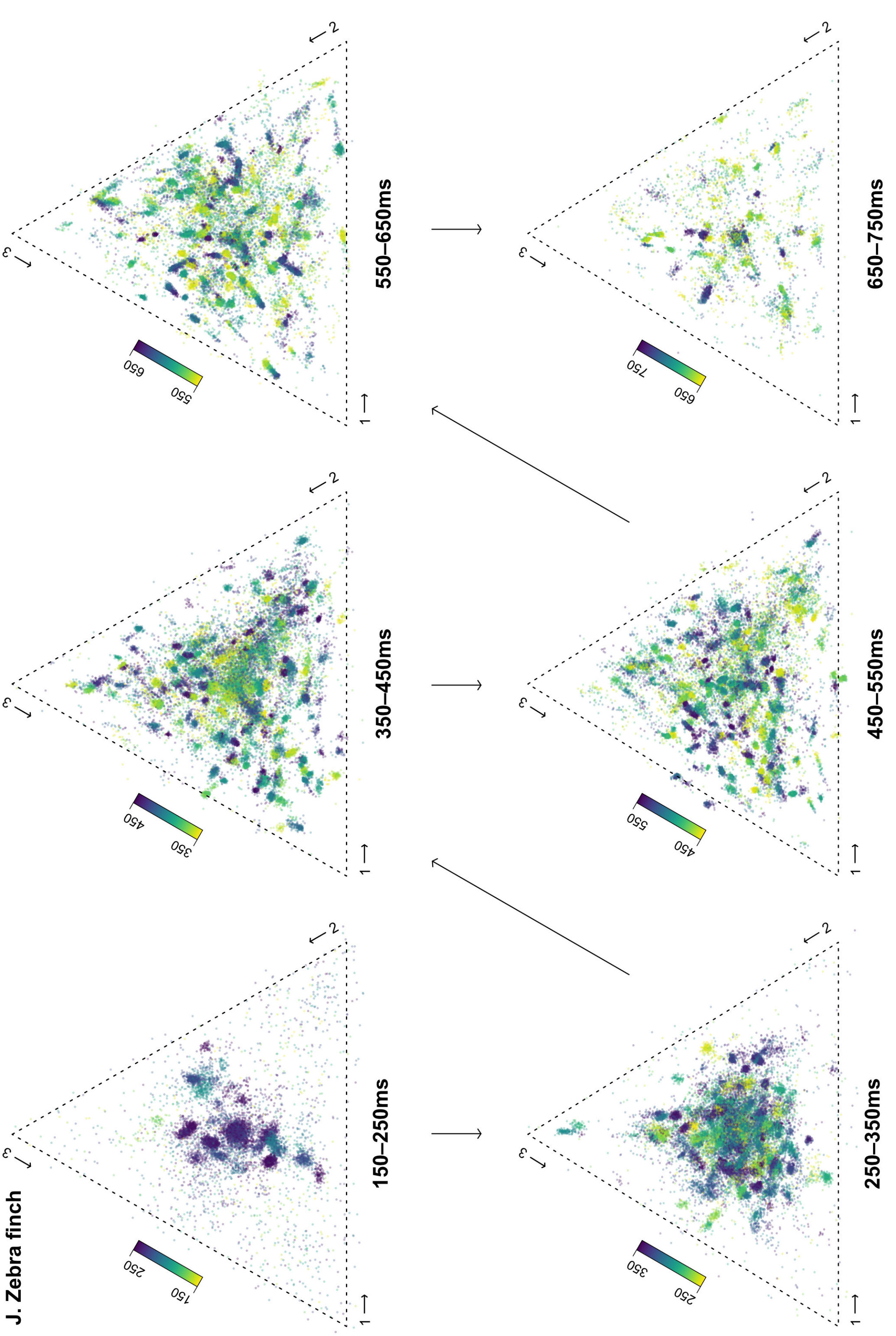
[7] They write that "rhythms were not discretized across zebra finches, even within a colony." They observe a "a roughly unimodal distribution of rhythms, with a prominent mode at 1 : 1 ratio", but looking at the triangle plot, this seems wrong.

[6] It is easy to produce plots that answer those questions from the original IEMP corpora: see supplement C1 for Cuban salsa and son triangles per song or per instrument.


Tempo: duration (ms)
400
slow → fast
1200
A. Tunisian stambeli
interval 1 →
← interval 2
← interval 3
131
231
121
331
132
221
232
321
332
133
122
233
111
322
211
311
123
223
323
312
112
212
313
213
113
B. Malian jembe
interval 1 →
← interval 2
← interval 3
131
121
232
111
323
212
313
231
331
221
332
321
322
211
312
311
132
133
122
233
223
123
112
213
113
C. Uruguayan candombe
interval 1 →
← interval 2
← interval 3
131
121
232
111
323
212
313
231
331
221
332
321
322
211
312
311
132
133
122
233
223
123
112
213
113


Tempo: duration (ms)
400
slow → fast
1200
D. Western piano (Bach)
131
231
121
331
132
221
232
332
321
133
122
233
111
322
211
311
123
223
323
312
112
212
313
213
113
E. North-Indian raga
F. Cuban salsa and son
interval 1 →
← interval 2
interval 3 →
131
121
232
111
323
212
313
231
221
332
322
211
312
331
321
311
132
122
233
223
112
213
133
123
113


G. Persian Zarb
H. Zebra finch song
I. Thrush nightingale song
Tempo: duration (ms)
400
slow → fast
1200
Tempo: duration (ms)
100
slow → fast
700
interval 1 →
← interval 2
← interval 3
111
112
113
121
122
123
131
132
133
211
212
213
221
222
223
231
232
233
311
312
313
321
322
323
331
332
333


J. Zebra finch
150–250ms
150
250
250–350ms
250
350
350–450ms
350
450
450–550ms
450
550
550–650ms
550
650
650–750ms
650
750
1
2
3

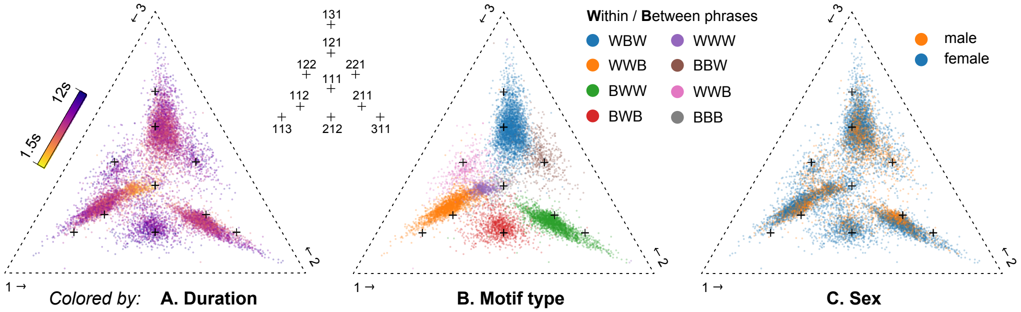


FIGURE 7.6 – Rhythms in the song of the lemur *Indri Indri* appear to be categorical (De Gregorio et al., 2021). Their vocalizations form phrases, and intervals that fall *between* phrases are roughly twice as long as those *within* phrases. For example, a motif of three intervals with types within-within-between has a ratio around 1:1:2 (WWB, orange in **B**). The clusters appear to fall just beside the small-integer ratio rhythms. Finally, there is a slight difference between rhythms in male and female productions **(C)**.

7.5 Singing primates

The work by Tina Roeske and colleagues inspired Chiara De Gregorio et al. (2021) to look at the rhythm of indri vocalizations. The *Indri indri* is a lemur, a primate species native to Madagascar, known for its particularly loud singing duets. The name "lemur", according to Wikipedia, is derived "from the Latin *lemures*, which refers to specters or ghosts that were exorcised during the Lemuria festival of ancient Rome." Ironically, lemurs themselves have almost been exorcised from this planet: they are critically endangered. The 39 individuals that De Gregorio et al. (2021) studied are, in fact, around 1% of all indri left. Analyzing recordings of their duets, the authors found that the inter-onset intervals in their vocalizations are not uniformly distributed but cluster around the ratios 1 : 1 and 1 : 2.

The rhythmic categories can be seen in the rhythm triangles in Figure 7.6. In particular, I colored the motifs by their *type* in subplot B. Indri songs consist of phrases, and De Gregorio et al. (2021) classified each interval as either falling within (W) a phrase or between (B) two phrases (or between two isolated notes). This divides motifs of three intervals into eight possible types: within-within-between (WWB), within-between-within (WBW), and so on. Every cluster in Figure 7.6B clearly corresponds to such a type. The blue WBW cluster lies mostly right of the integer ratios 1 : 2 : 1, which corresponds to a short(ish)-long-short motif. The orange rhythms of type WWB are short-short(ish)-long, and the green ones (BWW) are long-short-short(ish). Indri vocalizations, in short, appear to use two duration values: a long one between phrases and an approximately twice as short one within phrases.

Science journalists jumped on this story: "Singing lemurs have a distinctly human sense of rhythm, study finds", The Guardian wrote. Although I applaud the media attention from a conservationist point of

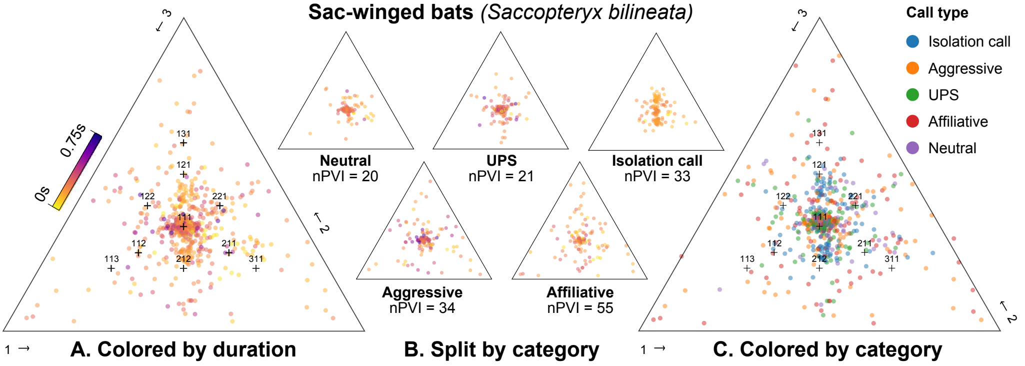


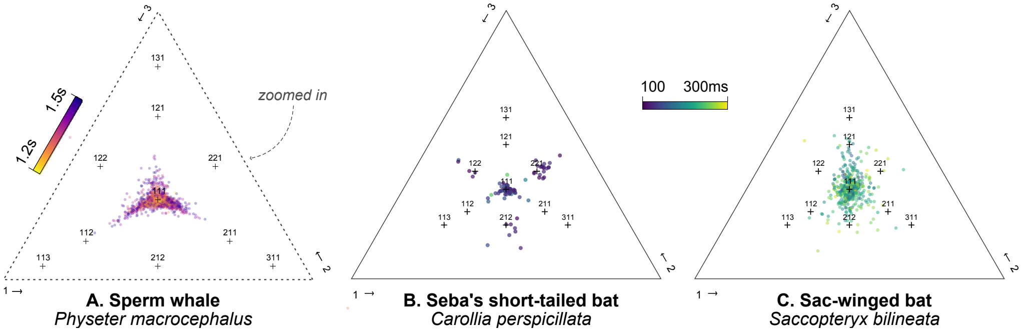
FIGURE 7.7 – Rhythms in the babbling of the sac-winged bats tends to be isochronous (Fernandez et al., 2021). The motifs are assigned to one of five categories, based on the use of corresponding syllables in the adult repertoire (UPS = Undifferentiated proto-syllables). We show the categories separately **(B)** as well as combined **(C)**, which suggests that there are slight differences in the rhythm of different categories. Neutral vocalizations are for example more isochronous (lower nPVI) than affiliative ones (large nPVI). The nPVI scores are averaged over te six syllable trains in each category.

view, it seems overly enthusiastic. A rhythmic repertoire of two duration values—in recordings of multiple individuals spanning more than a decade—seems rather limited. Rhythmic categories in human music, meanwhile, are extremely flexible: they vary across styles, within styles across songs, and within songs across instruments. And so while the finding that indri vocalizations contain rhythmic categories is certainly interesting, labeling it "distinctly human" seem premature.

On a more technical note, the triangle plot in Figure 7.6 also shows that there is less *isochrony* in the data than the paper appears to suggest. An isochronous rhythm is a steady beat where the intervals between all onsets are the same. And while isochronous pairs of intervals (1 : 1) are indeed common in indri vocalizations, three successive intervals of equal duration (1 : 1 : 1) are almost absent. This can be seen by looking at the very center of the triangle, which is relatively empty.

In fact, you can use the distance between a motif and the center of the triangle as a measure of isochrony: the closer to the center, the more isochronous a motif is. In the triangle, we are looking at motifs of length 3, but you can similarly define *(n-gram) isochrony* for other lengths. Interestingly, for motifs of length $n = 2$, the average isochrony is essentially the opposite of the *normalized pairwise variability index* (nPVI), a metric that was originally introduced to measure durational contrasts in speech, but that has also been used to study music.[9] The rhythm triangle thus suggests a novel rationale for the nPVI: it is exactly proportional to how un-isochronous the average rhythmic motif of length 2 is. I explain all of this in more detail in supplement C2.

[9] See Condit-Schultz (2019) for a critical evaluation.



A. Sperm whale
Physeter macrocephalus

B. Seba's short-tailed bat
Carollia perspicillata

C. Sac-winged bat
Saccopteryx bilineata

FIGURE 7.8 – Rhythms in vocalizations of the sperm whale and two bat species (Burchardt & Knörnschild, 2020). The click trains of sperm whales (**A**) are used for echolocation and are extremely regularly timed (note that the plot zooms in on the center). The social vocalizations of both bat species (**B** and **C**) are also strongly isochronous, be it to a lesser extent.

7.6 A bestiary of triangles

The nPVI has recently been used in a number of studies that address how isochronous or beat-like certain animal vocalizations are. Since the data in these studies has been made publicly available, we can use it to test our novel metric of isochrony. But first I briefly discuss four of the studies and visualize the original data in rhythm triangles.

The first study concerned babbling bats. Producing speech sounds requires very fine control over your articulatory muscles, and one idea is that babbling ("da-da!") allows us to gain that kind of control: an articulatory workout. One may expect more species to have these practice periods if they at least modify their vocalizations based on what they hear from others—if they are so-called vocal production learners. And indeed, something like babbling is common among songbirds. Knörnschild et al. (2006) also reported babbling in sac-winged bats (*Saccopteryx bilineata*).[10] These bats have a large vocal repertoire, consisting of 25 different syllables, which combine to form ten types of vocalizations.

Before acquiring the adult repertoire, the bats go through a babbling phase, that according to a recent study by Fernandez et al. (2021) shares many key characteristics of babbling in human infants. The presence of a regular beat appears to be one of those commonalities. Based on nPVI scores, the authors conclude that "four of the five different syllable train categories [...] had a regular beat". Figure 7.7B visualizes those categories, and suggests that the affiliative category, which is most spread out and has highest nPVI, is the category without a regular beat.[11]

The second paper, Burchardt and Knörnschild (2020), concerns isolation calls of *adult* sac-winged bats (*Saccopteryx bilineata*), as well as isolation calls of Seba's short-tailed bat (*Carollia perspicillata*) and click trains of sperm whales (*Physeter macrocephalus*). I have plotted the rhythms in Figure 7.8. The click trains of sperm whales are extremely isochronous,

[11] I cannot exactly replicate the statistics in their table S4: even the 101 statistics deviate slightly, and the nPVI scores I compute are all lower.

[10] The bats are named after sacs in their wings in which males brew their signature smells from "genital and gular secretions." If you wonder how they do so, consult Voigt et al. (2005).

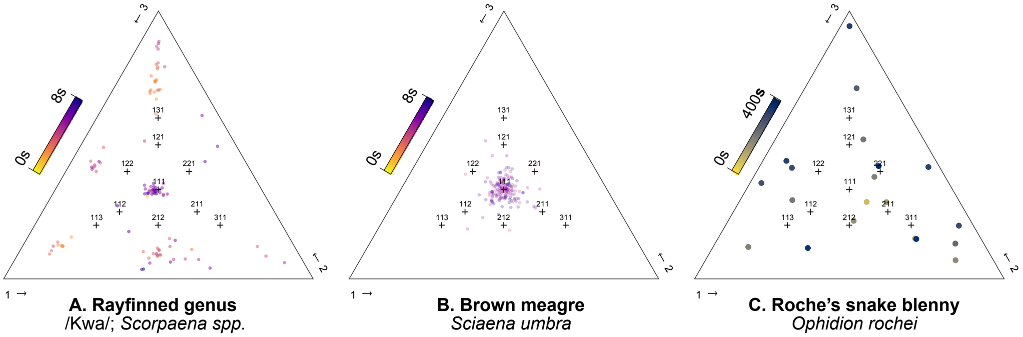


FIGURE 7.9 – **Rhythms in fish sounds (Burchardt et al., 2021).** Shown are three types of sounds produced by fish in the Mediterranean. The /Kwa/ sounds of the rayfinned genus (**A**) and the vocalizations of the brown meagre (**B**) are largely isochronous, while those of Roche's snake blenny **(C)** are very irregular. That was the reason Burchardt et al. (2021) included this species.

which is not surprising since they are used for echolocation. The calls of Seba's short-tailed bat are largely isochronous, although the plot suggests some clustering around the motifs 2 : 1 : 2, 1 : 2 : 2, and 2 : 2 : 1. These calls are about twice as fast as the calls of the sac-winged bat, which also have an isochronous rhythm.

The third paper, Burchardt et al. (2021), analyzed the sounds made by several fish species from the Mediterranean: a particular reproductive vocalization of the brown meagre (*Sciaena umbra*), the so-called /Kwa/ sound that is "most probably produced by species from the rayfinned genus *Scorpaena*", and vocalizations of Roche's snake blenny *Ophidion rochei* with long, irregular gaps. The latter indeed seems completely irregular, while the brown meagre's vocalizations and /Kwa/ sounds tend to be quite isochronous. The /Kwa/ sounds also contain some very high-integer ratio rhythms, or what Roeske et al. (2020) might call ornaments. These occur when otherwise isochronous calls are preceded by a very short call, as can clearly be seen in the waveforms (see Figure 1 of the original paper).

The fourth paper, Filer et al. (2021), compared vocalizations of two Australian frog species: the wallum sedge frog (*Litoria olongburensis*, WSF) and the eastern (common) sedge frog (*Litoria fallax*, ESF). If the two species vocalize at the same time, they are in competition for a place in the acoustic space and the paper suggests that the frogs adapt the rhythm of their vocalizations in the presence of competitors. Figure 7.10 visualizes the rhythms for both species in the presence and absence of competitors.

7.7 Isochrony

With a small bestiary of rhythm triangles and datasets in place, we can evaluate the measure I introduced above: the *n*-gram isochrony, which

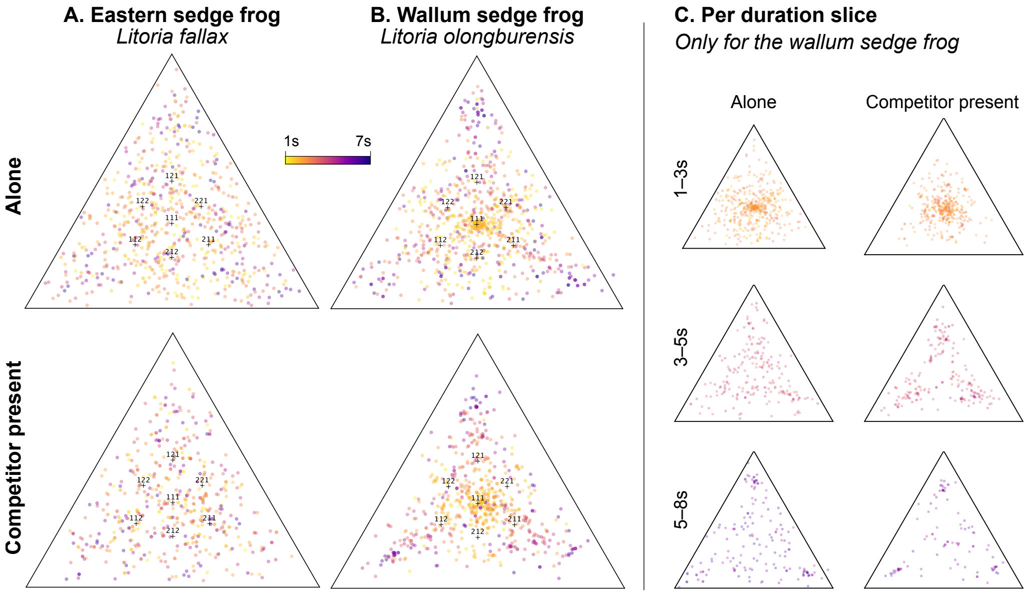


FIGURE 7.10 – Rhythms in vocalizations of two sibling frog species (Filer et al., 2021). The eastern sedge frog (**A**) and wallum sedge frog (**B**) are in acoustic competition and adjust the rhythm of their vocalizations when competitors are present. This is not very clear from the triangle plots and possibly better seen in particular duration ranges (**C**).

generalizes the nPVI. It measures the distance between a given motif of length n and the completely isochronous motif of n identical intervals. Higher values of n intuitively correspond to higher-order notions of isochrony. If a rhythmic dataset has a high average isochrony for $n = 2$, pairs of successive intervals are frequently identical. But for $n = 6$, the dataset has to contain many sequences of six successive, almost identical intervals: a much stronger form of isochrony. Irrespective of n, the score is normalized so that a value of 1 indicates perfect isochrony, while a value of 0 corresponds to the opposite, limit case where all intervals are negligibly short except for one long interval—the corners of the triangle for $n = 3$ (see section C2 for details).

I have plotted the distribution of isochrony scores for motifs of length $n = 2$, 3, and 6 in Figure 7.11, summarizing the entire bestiary. Below the datasets from Roeske et al. (2020), you find data for the indri (De Gregorio et al., 2021). The figure illustrates a point I made earlier: isochrony for pairs of intervals ($n = 2$) may be common since there is a peak close to 1, but longer isochronous sequences appear to be absent ($n = 3$ and $n = 6$). The situation is different for the brown meagre and the sac-winged bat (*S. bilineata*), where even six successive isochronous intervals are pretty common. But the most extreme level of isochrony, unsurprisingly, can be found in the echolocation calls of the sperm whale.

Figure 7.11 also shows isochrony scores for the musical datasets (Roeske et al., 2020). Western piano music stands out by its high isochrony scores,

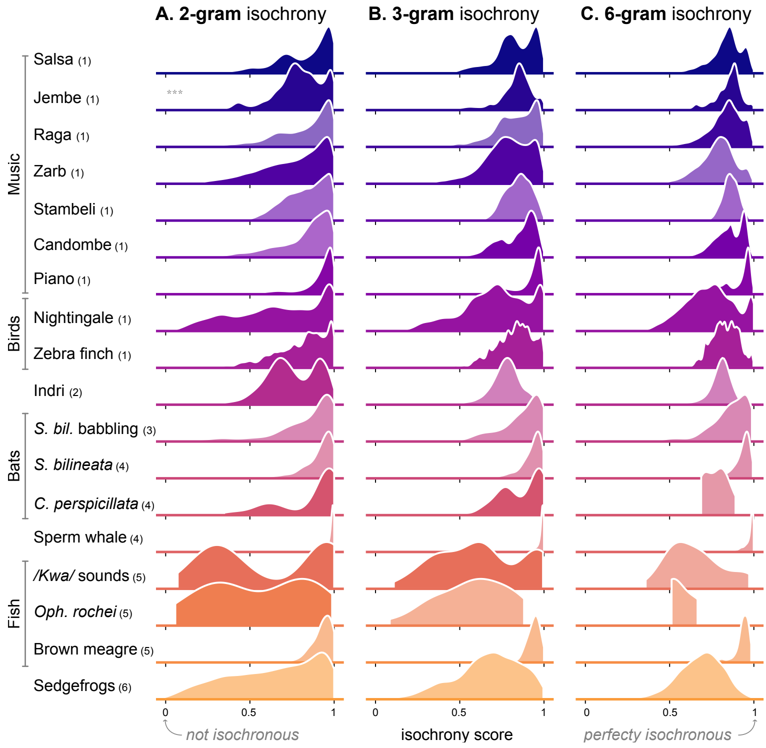


FIGURE 7.11 – Distribution of isochrony scores across several musical traditions and non-human vocalizations. Isochrony scores indicate how much n-gram motifs of $n = 2, 3$ or 6 consecutive intervals deviate from isochronous motifs. A value of 1 indicates perfect isochrony. Scores for $n = 2$ are inversely proportional to the nPVI score **(A)**. At that level we see isochrony in vocalizations of the indri (cf. De Gregorio et al., 2021), but isochronous motifs of length 3 or 6 **(B-C)** are absent. High scores in **(C)** mean that subsequences of six intervals are very regular. This is the case for the extremely regularly timed echo-location vocalizations of sperm whales. Vocalizations of *Ophidion rochei* are completely irregular, resulting in a wide distribution of isochrony scores. In music, we see high levels of isochrony in Western 'piano' music—or harpsichord music, really—and more rhythmic diversity in salsa and jembe music.

Data is from (1) Roeske et al. (2020), (2) De Gregorio et al. (2021), (3), Fernandez et al. (2021), (4) Burchardt and Knörnschild (2020), (5) Burchardt et al. (2021) and (6) Filer et al. (2021).

even for $n = 6$. This dataset consists of recordings of 'piano' music by Bach, but these are treated as single-instrument recordings, and voices are therefore not differentiated. This means that we are effectively looking at a surface rhythm of multiple voices, which is much denser (and presumably full of sixteenth notes, as it was written for harpsichord). The other datasets, especially jembe and salsa music, contain more varied rhythms. All in all, the isochrony score proposed in this interlude seems to be a useful generalization of the nPVI, capable of describing different orders of isochrony.

To conclude, the second half of this interlude can be read as an exercise in musical typology, in which we compare the variability of a musical feature (types of rhythmic motifs) across datasets. To be more precise, it was an exercise in *continuous, cross-species rhythm typology*, since the feature of interest *could* vary continuously, and we studied it in different species. But in every case, it remains an empirical question whether the feature *does* vary continuously, or whether it can be divided into discrete categories. It might, as with the indri, or it might not, as with the sedge frogs.

The question of categoricity applies not only to rhythm but to continuous features generally. We can ask the same about melodic modes in plainchant, like in chapter 5, or about shapes of melodies. Indeed, that will be the topic of the next chapter. To give you a flavor, the core idea is already illustrated in Figure 7.11. If we, for example, look at the isochrony distribution ($n = 2$) of jembe music, we see multiple peaks or *statistical modes*. This can only happen if there are multiple clusters of motifs that have different distances to the isochronous motif. Multimodality, in this case indicates *categorical rhythm*.[12] This suggests that one can look for categoricity by testing for multimodality, and that is what the next chapter will do for melodic contour.[13] Let's dip into it.

SUPPLEMENTS **C1** IEMP Cuban Salsa and Son · **C2** Measuring isochrony

DATA AND CODE The data and code for all results in this interlude, as well as all figures, have been made available publicly online via github.com/bacor/rhythm-triangles. The original data from the cited papers have also been included where possible.

REFERENCE The work in this chapter has not been published or presented elsewhere.

AUTHOR CONTRIBUTIONS BC designed most of the research in this chapter, analyzed all data and wrote this chapter. HH, JAB and CTC suggested rhythm visualizations.

ACKNOWLEDGEMENTS I want to thank Henkjan Honing, Carel ten Cate, Ashley Burgoyne, and Marianne de Heer Kloots for sharing their ideas on visualizing rhythmic datasets. Jelle Zuidema helpfully pointed out the connection to typology in other chapters of this dissertation and provided valuable feedback on an earlier version of this chapter. Thanks to Tina Roeske and Chiara De Gregorio for sharing their data and answering some of my questions. Finally, I would like to thank researchers like Mirjam Knörnschild, Lara Burchardt, Alannah Filer, and Ahana Fernandez for making your research data so easily accessible.

[12] The converse need not be true: symmetrical clusters (categoricity) that are all equally far from the center result in a unimodal distribution of isochrony scores. This seems to be the case in some music datasets (e.g., raga).

[13] I have already applied a Hartigans' dip test for multimodality to the isochrony distributions shown in Figure 7.11: those with a darker shade are significantly multimodal. But because of footnote 12, this is not a good test of categoricity. A better alternative considers the distribution of pairwise distances instead of only the distances to the center. The idea is explained in detail in the next chapter, and applying it to rhythm is left for future work.

CHAPTER 8

Article

Shapes of music

How can one best describe the shapes of melodic phrases in musics from across the globe? Previous studies have often relied on typologies with a discrete set of contour types. We question their adequacy: we find no evidence that phrase contours cluster into discrete types in German and Chinese folksongs or Gregorian chant. The test for clustering we propose applies the dist-dip test of multimodality after a UMAP dimensionality reduction. The test correctly identifies clustering in a synthetic dataset of contours but not in actual phrase contours. These results argue against the use of discrete typologies. Additionally, we identify a hidden parameter in two discrete typologies that can strongly skew the type distributions. Our findings suggest that melodic contour is best seen as a continuous phenomenon. We end by revisiting the melodic arch hypothesis using a continuous approach to contour.

8.1 Introduction

Recent years have seen a renewed interest in the search for musical universals: properties common to most or even all musics across the world (Brown & Jordania, 2011; Mehr et al., 2019; Savage et al., 2015). Musical universals can help to identify the constraints within which most music is made, which may, in turn, point to biological predispositions for music (*musicality*) and inform theories about its evolution (Honing, 2018). The frequent use of isochronous beats is, for example, consistent with a biological, cognitive capacity for beat perception (Winkler et al., 2009). But music might also be shaped by physiological constraints. A frequently cited universal is the prevalence of arch-shaped or descending melodic phrase contours, sometimes known as the *melodic arch hypothesis* (Brown & Jordania, 2011; Huron, 1996; Savage et al., 2015; Savage et al., 2017). It has been suggested that the physiology of our vocal system explains their prevalence, making pitch contours that fall towards the end of a phrase easier to produce (Tierney et al., 2011).

Questions of universality go hand in hand with classification: they usually require typologies that break down music into a set of *characters* or *features* with several possible *values* or *types* (Brown & Jordania, 2011). Examples of features are the type of scale used or the type of rhythmic subdivision. Both of these are *discrete* characters, but there are also *continuous* characters, like tempo when measured in beats per minute. Even though it can vary almost continuously, melodic contour is often treated as a discrete character and described as *ascending*, *descending*, *arch-shaped*, and so on. Mapping the frequency of those contour types across cultures then allows one to assess cross-cultural generalizations like "arch-shaped and descending contours are the most frequent contour types across cultures". Besides *synchronic* questions, typologies also play a role in *diachronic* questions. In the words of Herzog (1937, cited in Adams, 1976), "it is through a discovery of types that we hope to find the stylistic relationships, which are often genetic and historical relationships between different melodies."

The validity of all such comparative questions depends on the validity of the typology used. Consider, for example, a character *modality* taking the values *major*, *minor*, and *irregular* based on the presence of the major third of the scale. While this could make sense for common practice music, it is an awkward description of the modalities in Gregorian chant or the songs of the Lakota (Densmore, 1918).[1] This problem will be familiar to comparative linguistics. If a typologist wants to compare the category 'noun' in different languages, a descriptive linguist could insist that 'noun' has a different, or even incommensurable, meaning in each of those languages (cf. Haspelmath, 2018). But while linguistic typology has flourished despite the problems inherent to comparison, ethnomusicology has largely avoided comparison and questions of typology (Nettl, 2005, ch. 6).

In this chapter, we revisit the question of melodic contour typology: how to describe the shapes of melodic phrases? We first review some of the literature on contour typology. The common assumption seems to

[1] Frances Densmore tabulated the modality of the songs she collected in precisely this way, but was well aware that this notion of modality was alien to the music she was studying.

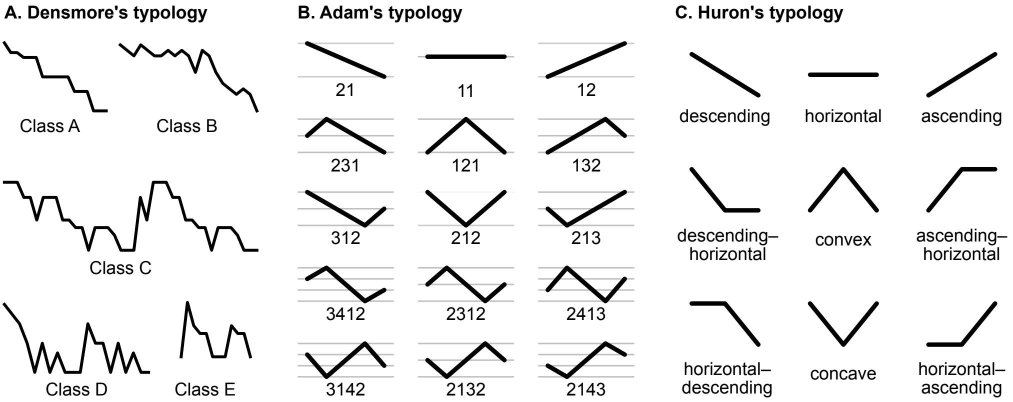


FIGURE 8.1 – Three contour typologies. (A) Densmore (1918) divided Lakota songs into five broad classes based on their contour, each identified by one representative song. Such an inductive approach contrasts with the deductive typologies by **(B)** Adams (1976) and **(C)** Huron (1996). Adams' typology considers all possible orderings of the four boundary pitches. A melody, for example, has type 3412 if its initial (e.g., 3) is above the final (2) and if it in between first reaches a higher (4) and then a lower (1) pitch. Huron's typology considers the ordering of the average pitch on the first, middle, and final parts of a melody.

have been that "that melodic contour types do exist and can be empirically defined" (Adams, 1976, but also e.g., Savage and Brown, 2013). We question that assumption. Contour types cannot be said to exist if contours do not cluster accordingly. But we fail to find any evidence for clustering in phrases from three repertoires, both within each repertoire and when aggregating them. As a result, discrete typologies partition the contour space somewhat arbitrarily. If the partition is not fair, one risks misrepresenting the variability. We show that this is precisely what two typologies turn out to do. Although we also propose a remedy using a maximum entropy criterion, the fact remains that melodic contour appears to be a continuous phenomenon.

8.2 Melodic contour typology

Contour is a key aspect of melody. When still in the womb, humans already appear to be sensitive to the pitch contour of the mother tongue (Mampe et al., 2009), and once born, contours remain a central cue for our first steps in language learning. With such importance in early life, it is not surprising that Dowling (1978) argued that contour and scale underpin our melodic memory. Composers who want to write catchy melodies must also attend to their contours. Indeed, many composition treatises discuss how to shape melodies. Piston (1970) for example opens his *Counterpoint* with a chapter on the "melodic curve", while Perricone (2018) reassures us that "there are only five basic melodic shapes or contours" (p. 179): *ascending*, *descending*, *arch*, *inverse arch*, and *stationary*. Such accounts are primarily meant prescriptively, not as a cross-cultural description of contour shapes—even though we will see some overlap.

Adams (1976) identifies a plethora of melodic contour descriptions in the academic literature. Some narrate how the melody progresses, others settle for word lists, yet others for graphs. Some authors propose ten types, others six, yet others four. Descriptions are often ambiguous—how to distinguish a *bow* from an *arch*?—and sometimes even inconsistent. Alan

Lomax' cantometrics project, for example, coded melodic shape as *arched*, *undulating*, *descending*, or *terraced*. But where the first three apply to the most characteristic phrase in a song, the latter applied to the entire song. Its successor, CantoCore (Savage et al., 2012), only includes phrase-level contour types, but six of them (*horizontal*, *ascending*, *descending*, *U-shaped*, *arched* and *undulating*) and the annotator is given considerable freedom to resolve ambiguities.[2]

An early and more systematic contour analysis is Frances Densmore's 1918 study of the music of the Lakota people (also known as the Teton Sioux). She visualized the contours of complete songs by plotting the accented notes (the downbeats in her transcriptions) while ignoring accidentals. This allowed her to cluster songs into five classes with similar contours and apparently sometimes similar social functions. It is not entirely clear *how* Densmore classified the songs. Sometimes the global shape seems to be the crux (class A usually has only descending intervals), but she also mentions characteristic local features (such as a repetition of the lowest note in class C or the ascending opening in D). Such features are not mutually exclusive, but they suggest the classes are based on more than contour alone. Densmore identified one exemplary song for each class, making her typology entirely culture-specific.

Deductive typologies are not culture-specific since their types are derived from first principles. An example of this is Adams' rather intricate typology (Adams, 1976). It considers all possible orderings of a melody's four *boundary pitches*: the initial note I, the final F, the first occurrence of the lowest pitch L, and the highest H. To simplify matters, assume that there are k distinct boundary pitches, with $L = 1$ the lowest and $H = k$ the highest, so that I and F fall in between. Now a contour type is something like $(I = 2, H = 4, L = 1, F = 3)$ or 2 4 1 3 in short. This means that the initial is below the final, and the melody reaches the highest and lowest pitch in between. There are fewer than four values whenever the final (or initial) is also an extreme value, as in $(I = 2, L = 1, H = F = 3)$ or 2 1 3: starting somewhere in the middle, descend to the lowest and end on the highest pitch. With this representation, one can determine that there are 15 orderings, illustrated in Figure 8.1.

Although Adams' paper is perhaps the most comprehensive study of contour typology, it attracted few followers. The typology best known today was proposed by David Huron and is conceptually much simpler. The idea is to reduce a melodic contour to three pitches: the initial I, final F, and the average pitch M of all notes in between (the middle). The contour types are the nine possible orderings of these three pitches. For example, if $I < M > F$, the contour type is *convex*, if $I = M > F$, it is *horizontal-descending*, and so on. Huron also mentions a variant of the typology that divides the melody into three equal parts and uses the average pitch on the initial, middle, and final third. This should be less sensitive to the initial and final pitch, and like other later studies (e.g., Savage et al., 2017; Tierney et al., 2011), we will consider this variant.

[2] The instructions allow coding of "clear 'hyper-phrase' contours" as a single contour and advice the annotators to ignore "temporary interval changes that do not greatly affect the dominant melodic contour."

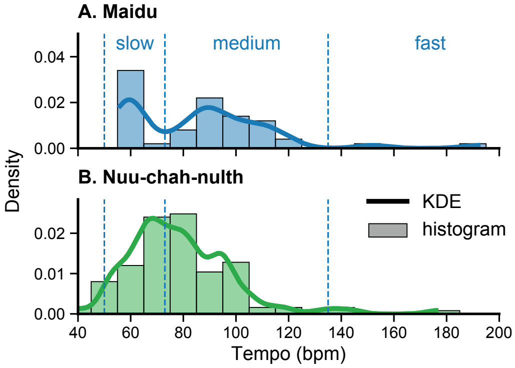


FIGURE 8.2 – Tempo distributions of songs of the Maidu and Nuu-chah-nulth. Shown are histograms and kernel density estimates (KDE). The tempos in Maidu music **(A)** cluster in multiple groups, suggesting a typology with a *slow*, *medium* and possibly *fast* type—which would not be appropriate for music of the Nuu-chah-nulth **(B)**.

Whereas Densmore's typology is derived empirically and therefore culture-specific, Adam's and Huron's typologies are derived from first principles and culture-independent. But which typology should one use? To address that question, we analyze the same phrase contours and random segments as we studied in chapter 6.

8.3 Phrase contours

We use two collections of 'German' folksongs from Catafolk: the *Erk* of 1700 songs (Erk & Böhme, 1893a, 1893b, 1894) and the *Böhme* subset of 704 songs (Böhme, 1895). In addition, we analyze 152 folksongs from Nova Scotia, collected by Creighton (1932), and the three Chinese subsets in *Essen*: *Han*, *Shanxi* and *Natmin*. Finally, we include phrases from Gregorian chants in three liturgical genres: *antiphons*, *alleluias*, and *responsories*. All of these come from the *Liber Usualis* in the GregoBase Corpus, using breathing marks to indicate phrase boundaries (see Figure 2.3).

Just as in chapter 6, all phrases are converted to fixed-length pitch sequences: we interpolate the melody and then sample $N = 50$ pitches equally spaced in time. Using a fixed number of pitches allows us to compare phrase contours irrespective of their length. This means we effectively normalize the phrase duration and usually interpret the temporal axis as the relative position in the phrase. Phrase length, nevertheless, has an obvious effect on contour shape: the more notes, the more shapes you can make. To study such effects, we also record phrases' length (number of notes) and duration (in quarter notes).

The idea that phrases may be shaped according to multiple types raises a question: do these types mostly or perhaps *only* show up when a melody is segmented in phrases or also when sliced up differently? To evaluate this, we also extract random segments of all melodies, that are roughly as long as phrases but unlikely to overlap with them (see section 2.4). Finally, we create two cross-cultural datasets by aggregating phrase contours and random segments sampled from each of the nine datasets. In this chapter, we primarily discuss the aggregate dataset.

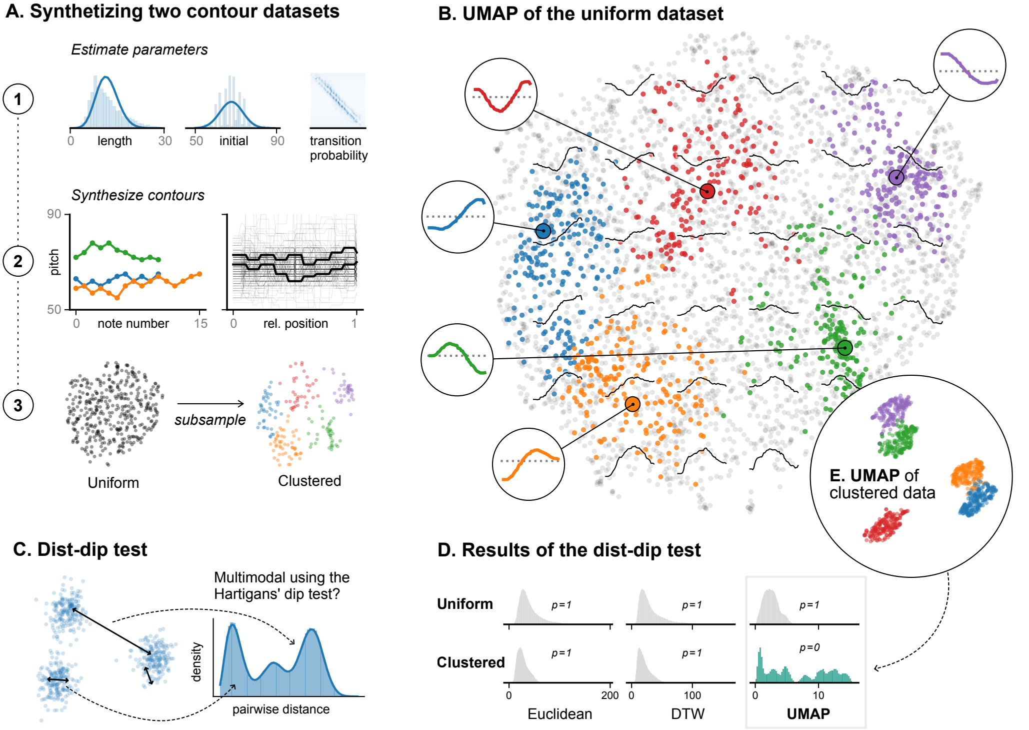


FIGURE 8.3 – The dip-dist test discriminates clustered and unclustered synthetic datasets. **(A)** Synthetic contours are samples from a Markov process whose parameters are estimated from actual phrase contours. We create a *clustered* dataset by subsampling contours close to five suitably chosen cluster centers. Panel **(B)** shows a two-dimensional UMAP visualization of the *uniform* dataset (gray) in which the clustered dataset (colored) is projected. The grid of black contours illustrates that UMAP organizes the space almost exactly like two-dimensional cosine contours ("ascendingness" horizontally, "archedness" vertically). The cluster centers (encircled) correspond to distinct shapes. **(C)** The dip-dist test applies the Hartigans' dip test on pairwise distances: a multimodal dataset should have a multimodal distribution of pairwise distances. **(D)**. The dist-dip test correctly identifies the clustered dataset as multimodal but only the distances in a ten-dimensional UMAP embedding (i.e., using the UMAP-dist test). This lower dimensional manifold appears more informative in that it more clearly separates the clusters **(E)**.

8.4 Clusterability with the dist-dip test

Returning to our central question—which typology should one use to describe melodic contour?—we would argue that a discrete typology should be *appropriate* for the data, in the sense that the types should correspond to clusters in the data (cf. Spike, 2020). Let us illustrate this using a simpler musical feature: tempo. When measured in beats per minute, tempo is a continuous character. A tradition might nevertheless use only a few distinct tempo ranges, such as a *slow*, *medium*, and *fast* tempo. If we plotted the distribution of tempos of many songs, one would expect that distribution to have three peaks or *modes*. Figure 8.2A illustrates that the songs of the Maidu roughly follow that pattern (Densmore, 1958).[3] A typology with three corresponding types (slow, medium, and fast) would therefore be appropriate for Maidu music—but it is inappropriate for the music of the Nuu-chah-nulth (Densmore, 1939).

What we have just discussed is also known as *clusterability*: the question of whether the data show signs of clustering (Adolfsson et al., 2019). One way to formally test this is by looking for multiple statistical modes: peaks in the probability density. The *Hartigans' dip test* (Hartigan & Hartigan, 1985) does precisely that for univariate data like the tempos. It compares the null hypothesis that the data is unimodal with the alternative hypothesis that there are multiple modes. The test revolves around a statistic known as the *dip*: the maximal distance between the empirical cumulative distribution function and its closest unimodal approximation. In the case of the Maidu songs, the test confirms our intuition that the tempo distribution is multimodal ($p < 0.001$), while it cannot reject unimodality for the Nuu-chah-nulth songs ($p \approx 0.08$).

The Hartigans' dip test works for univariate data but not for multivariate data like the standardized contours. A simple trick can, however, reduce the multivariate problem to a univariate one. As illustrated in Figure 8.3C, the *dist-dip test* (Kalogeratos & Likas, 2012) tests whether a (multivariate) distribution is multimodal by checking whether the (univariate) distribution of pairwise distances is multimodal according to Hartigan's dip test. After all, if a distribution is multimodal, you expect to find at least two types of pairwise distances: small within-cluster distances and larger between-cluster distances. This means the distribution of pairwise distances is multimodal, precisely what the Hartigans' dip test can evaluate.

A systematic comparison of clusterability methods recommends the dist-dip test for a wide range of scenarios (Adolfsson et al., 2019). To further ascertain whether this test can reliably detect clusters in contour data, we first evaluate it on a synthetic dataset in which we enforce a cluster structure (see Figure 8.3A and B). The synthetic contours differ from those in chapter 6, as they are generated by a Markov process (see Figure 8.3A). We sample the contour's length and initial pitch from a Poisson and binomial distribution respectively, and then walk through pitch space according to the transition probabilities observed in the actual data. We normalize the duration, center the contour, and sample 50 equally spaced pitches to obtain a pitch sequence as before.

Generating many synthetic contours in this way results in a *uniform* dataset in the sense that it does not exhibit any clustering structure. By appropriately subsampling, one can create a *clustered* dataset from the uniform one. To find good cluster centers, we fit k-means, with $k = 5$, to a dataset of 25,000 synthetic contours and then select the 1000 contours nearest to the centroids found by k-means. To ensure the clusters correspond to shapes and not, say, pitch height, we used a cosine contour

[3] Tempo transcriptions from Catafolk, see chapter 3.

representation (see chapter 6) while selecting neighbors. This resulted in a uniform dataset without clusters and a clustered one with five equally sized clusters. We then computed the dist-dip test on 30k pairwise distances sampled from both datasets[4] and it utterly failed to reject the null hypothesis for the clustered dataset ($p \approx 1$).

The distribution of distances indeed looks unimodal (Figure 8.3D), even though the dataset is designed to contain clusters. And as shown in Figure 8.3E, those clusters are clearly visible in a low-dimensional projection made using UMAP (McInnes et al., 2018). This nonlinear dimensionality reduction technique learns a low-dimensional manifold that aims to preserve the global structure of the original data. This leads us to propose another test of multimodality: the *UMAP-dip test*: the dist-dip test but now applied to the distances on a lower, ten-dimensional manifold learned by UMAP.[5] The UMAP-dip test correctly rejects the null hypothesis for the clustered dataset but not for the uniform one (Figure 8.3D). It appears that the UMAP distances better capture the cluster structure of synthetic contours than Euclidean distance does.

One may wonder whether testing the projected data for multimodality is valid since the result now heavily depends on the projection. This is comparable to how principal component analysis is sometimes used before statistical testing in other clusterability approaches (Adolfsson et al., 2019). Alternatively, one can think of UMAP-dip as a formal test that can replace the visual inspection of low-dimensional visualizations for signs of clustering. But still, dimensionality reduction techniques like UMAP can sometimes suggest clusters that are not present in the data. This behavior would make the multimodality test overly sensitive. Importantly, however, this would strengthen a negative result: if UMAP-dip does *not* find evidence for multimodality, it probably isn't there.

8.5 Phrase contours do not cluster

Returning to the actual phrase contours, Figure 8.4 shows the distribution of pairwise distances for phrase contours and random segments and the two synthetic datasets. The color coding highlights that the dist-dip test only rejects unimodality for the clustered, synthetic dataset. In other words: contours do not appear to cluster.

To rule out that this is an artifact of the representation, we evaluated eight different ones (see supplement D1 for an overview of the experimental setup). Besides the raw *pitch* contour, we transposed the contours to make their shapes comparable irrespective of absolute pitch: we *center* contours to have mean 0 (cf. Savage et al., 2017), or transposed them so that the *tonic* (cf. Tierney et al., 2011) or *final* note of the phrase is 0. Next, in the *normalized* version of a contour, the minimum pitch is 0, and the maximum pitch is 1 (cf. Adams, 1976). Then we add two relative representations. The first measures the *intervals* between consecutive pitches, and the second only does this after smoothing the pitch contour. Finally, we

[4] We used the Python package `diptest`, which is a port of the R package by Martin Maechler.

[5] Note that instead of a two-dimensional manifold, we give UMAP more freedom and measure distances in a 10-dimensional manifold. This is one of the reasons for using UMAP instead of t-SNE: the latter does not scale well to higher-dimensional projections.

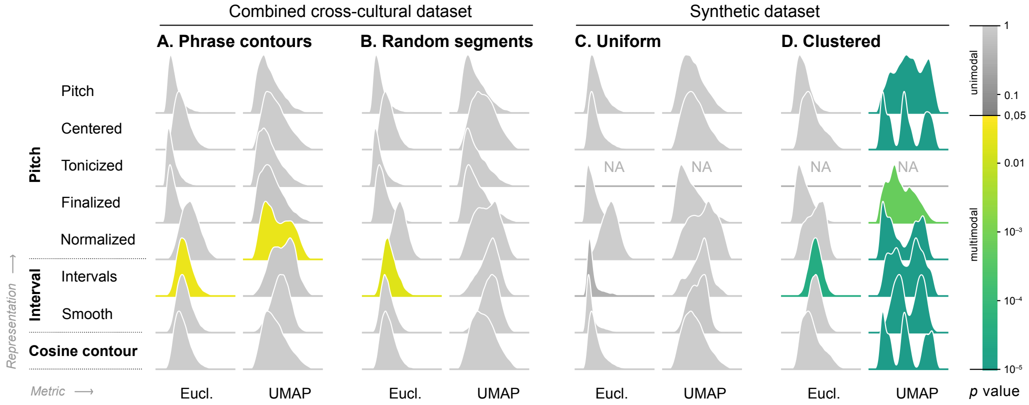


FIGURE 8.4 – **Melodic phrase contours do not cluster.** Shown are the distributions of pairwise distances between contours in various conditions. If contours cluster, we expect multimodal distance distributions. We test this using the Hartigans' dip test and let colors indicate p-values, such that grey distributions are not significantly multimodal ($\alpha = 0.05$). Eight different representations (vertically) and two metrics (horizontally) are analyzed: Euclidean distance and the distance in a lower-dimensional UMAP embedding. The latter successfully discriminates unclustered from clustered synthetic data (**C** vs. **D**; see also Figure 8.3). However, neither in phrases **(A)** nor in random segments from actual melodies **(B)**, the test fails to find clear evidence for clustering.

compute the *cosine contour*, which describes the shape of a contour as a combination of cosine functions (chapter 6). To rule out that our distance metrics prevented us from finding clusters, we also used *dynamic time warping* (DTW) dissimilarity besides Euclidean and UMAP distance. Intuitively, if two sequences are identical except that they have warped time differently—speed up here, slow down there—their DTW dissimilarity is zero.

With none of the eight representations, we find evidence for the clustering of phrase contours or random segments using any of three similarity metrics: Euclidean, DTW, and UMAP distance.[6] The same applies when we only consider unique contours, reduce the dimensionality of the contours from 50 to 10, or analyze individual datasets separately (see supplement D2).

One may expect the length of contours to have an effect: there are simply fewer possible shapes when you have only four notes instead of ten, and so you should see more clusters amongst shorter phrases. If we split out our analysis by length, the UMAP-dip test indeed indicates multiple modes for the smaller phrases up to 5 notes, and sometimes also for the longest ones of around 15 notes or more. But for most contours, with average lengths between 5 and 15, we still find no convincing evidence for clustering of phrase contours. In contrast, we do find such evidence for the synthetically clustered dataset (see supplement D3).

[6] The only possible exception is the interval representation, but that also suggests that the uniform, synthetic contours are clustered, which they are not.

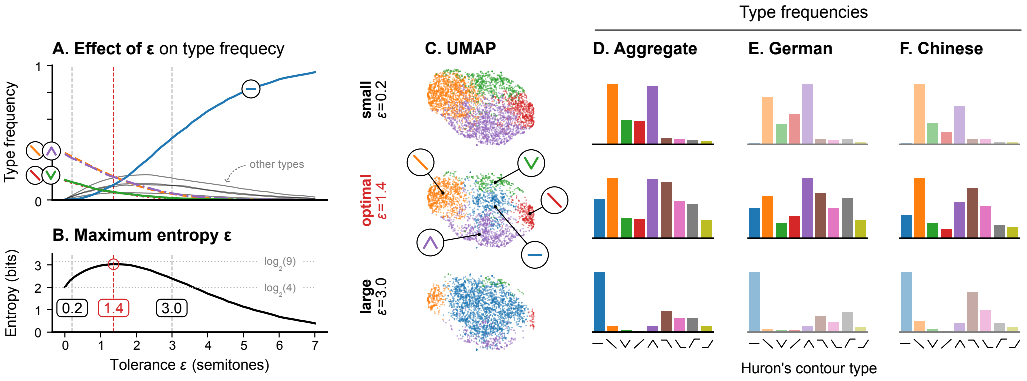


FIGURE 8.5 – The tolerance parameter ϵ in Huron's typology can strongly distort the type distributions. Huron's typology compares the average pitch on a contour's start, middle, and end, where pitches less than ϵ semitones apart are considered equivalent. **(A)** shows how type frequency depend on ϵ. **(B)** We propose to choose ϵ such that contours are as evenly distributed over types as possible: when the type distribution has maximal entropy. While a small value ($\epsilon = 0.2$) obscures partly horizontal types, a large one ($\epsilon = 3.0$) exaggerates it. **(C)** shows this by coloring some types in a UMAP visualization, while **(D)** shows this as a histogram. The maximum entropy criterion makes it harder to find frequency differences. This improves our confidence in observed effects, such as arch-shaped contours being more frequent in German folksongs than Chinese ones (**E** vs. **F**, middle row).

8.6 Rescuing discrete typologies

If contours do not cluster, it is hard to see how Adams' assumption that "contour types do exist and can be empirically defined" can be right. One is indeed free to *define* types, but these definitions will be somewhat arbitrary: the contours suggest no obvious partition. Can discrete typologies then still play a role in comparative questions? Only if they partition contours fairly and do not skew the type frequencies—precisely what Huron's and Adams' typologies appear to do.

FIXING HURON'S AND ADAMS' TYPOLOGY Recall that Huron's typology—the same argument applies to Adams'—compared the average pitch over three segments of a melody. These averages are usually not exactly identical, and so two pitches are treated as equivalent if their absolute difference is below a tolerance parameter ϵ. With zero tolerance, $\epsilon = 0$ semitones, horizontal contours will be extremely unlikely, but with a tolerance of an octave, $\epsilon = 12$ semitones, virtually any contour will be considered horizontal. In short, the choice of ϵ influences how evenly contours are divided over the classes. If not appropriately chosen, the tolerance parameter will strongly distort the type distribution, exaggerating the frequency of some types at the cost of others (see Figure 8.5).

And so, what is a good choice of ϵ? Tierney et al. (2011) use $\epsilon = 0.2$ semitones without motivation, and Huron does not report a choice of ϵ. We propose a more principled alternative: to choose ϵ so that the classes are as small as possible. Firstly, in the absence of clusters, dividing the space as equally as possible seems the best one can do. Secondly, this would ensure our typology contains no redundant, largely empty classes. And thirdly, this effectively imposes a strong prior *against* frequency differences between types. If we nevertheless find frequency differences across traditions, this strengthens the result. One can measure a type distribution's evenness with its *entropy*. A completely deterministic distribution has zero entropy, while a flat or uniform distribution has the highest possible entropy.

Concretely, we propose to choose ϵ so that it maximizes the entropy of the type distribution. Which ϵ yields maximum entropy depends on the dataset, and changing ϵ will change the typology. This means that the typology will be slightly different for different datasets. One way around this is to estimate a value of ϵ on a cross-cultural dataset and then use the resulting typology on each of the individual traditions. Applying this to the aggregated phrase contour datasets, we find that $\epsilon = 1.4$ semitones maximizes the entropy (Figure 8.5B). We discuss the implications for the melodic arch hypothesis in the next section.

LEARNING THE TYPES But even with a maximum entropy criterion, it is conceivable that the types do not divide the space fairly. If one nevertheless prefers to use a discrete typology, one can take inspiration from Densmore's inductive typology and *learn* the types. While her typology was specific to Lakota songs, the method is quite general: identify a number of representative contours and let those represent the types of a typology.

A computational analog could be a *k-means typology*, where one clusters the contours into *k* types by assigning them to the class of the nearest cluster centers. These centers are iteratively updated to minimize the within-cluster variance and come to represent the types in the typology, similar to Densmore's use of exemplars. This results in types that more accurately reflect the contours they represent than types in deductive typologies like Huron's or Adams'. All contours, for example, start and end flat because the melody is stable during the first and final note, which is reflected in the types (see supplement D4). This approach can be extended by using more sophisticated clustering methods and representations.

8.7 Embracing continuous typology

An inductive or learned typology effectively starts from the perspective that melodic contour is a continuous phenomenon—precisely in line with our findings. This is not the end of contour typology, but it does ask for a different approach: one that does not use distinct types but embraces the continuous nature of contour.

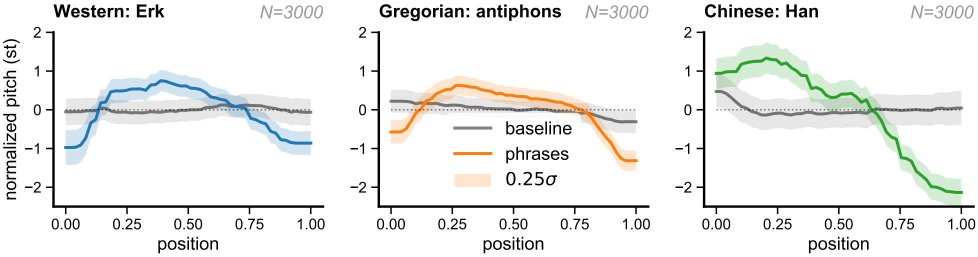


FIGURE 8.6 – Average phrase contours differ across three traditions. The average phrase contours of German folksongs, Gregorian chant, and Chinese folksongs compared to baselines of random melodic segments from the respective corpus (gray). They support the overall tendency for arch- *or* descending average contours but show interesting differences: the Chinese average is *not* arch-shaped and, strictly speaking, a counter-example to the average hypothesis. This illustrates how a continuous approach to contour typology.

Inspiration, again, comes from David Huron. Huron (1996) proposed his typology as a tool to investigate the melodic arch hypothesis. His analysis of over 6000 songs from the Essen folksong collection found that the most frequent phrase contour types were convex and descending contours. But Huron also computed *average phrase contours* by taking all phrases with a certain number of notes and averaging the pitches at every time step. When plotted, the average contours revealed clear arch shapes. Notably, this analysis treats melodic contour as a *continuous* character.

We replicate this result in Figure 8.6 for phrases from three traditions. That figure also shows the average shapes of the random segments in grey, which are almost entirely flat (cf. chapter 2).[7] While the chant phrases tend to be arch-like, the average phrase contour of Chinese folksongs looks quite different: not only is the range much larger, its shape is best described as descending, or perhaps horizontal–descending. We also observed this in chapter 6, but earlier studies (such as Savage et al., 2017; Tierney et al., 2011) seem to have overlooked this. One can, however, also see this using a discrete typology. Figure 8.5E and F show that descending contours are *more* common than convex ones in Chinese folksongs, while they are *less* common in German folksongs.

Strictly speaking, all this argues against two possible formulations of the melodic arch hypothesis: (1) that Huron's convex type is most frequent, and (2) that the average contour is arch-shaped. This underscores the need for precisely formulated, testable hypotheses. In fact, chapter 6 proposed one. If c_1 and c_2 are the first two coefficients of a cosine contour representation, c_1 measures its descendingness and $-c_2$ its archedness, and so we proposed the following:

> HYPOTHESIS: c_1 and $-c_2$ tend to be larger for melodic phrases than for random melodic segments.

This hypothesis was confirmed in German and Chinese folksongs.

[7] As noted before, this implies that the (somewhat tradition-specific) average phrase shape results from the particular placement of the phrase boundaries. This sanity check confirms that phrases are structurally relevant units and that the phrase annotations in *Essen* make sense (like the breathing marks in chant): had they been random, their average should have been flat.

8.8 Conclusions

In this chapter, we revisited the description of melodic contours. Analyzing phrase contours from three musical traditions, we found no evidence that the contours form clusters, which contradicts the assumption that contour *types* exist. We then showed that two discrete typologies, by Huron and Adams, contain a hidden parameter that can lead the typology to favor certain types over others. Although we proposed a remedy using a maximum-entropy criterion, we argue for a continuous approach to contour typology. This directly shows cultural differences and leads to a precise, testable reformulation of the melodic arch hypothesis.

A shortcoming of this work is the limited cross-cultural validity of the data analyzed. Except for Savage et al. (2017), most previous studies have relied on the Essen Folksong Collection, and this study only added Gregorian chant as a third tradition. However, our central finding—that contour shapes do not cluster—is negative. For that, cross-cultural validity is not as much of an issue: even the limited data we analyzed serves as a counter-example. The same is true when rejecting two formulations of the melodic arch hypothesis. But we think that the methods we proposed, and the continuous methodology we argued for, are sufficiently general to be applicable in other traditions—or even different domains.

Phrase contours, after all, are not only studied in music but also in language. The study of intonation in phonology has produced various cross-cultural generalizations, such as the *decline* from the beginning towards the end of a phrase, or the start of a phrase by a sharp rise known as the *reset* (Ladd, 2001). At the same time, models have been proposed to describe the intonation contours found in particular languages, such as the ToBI system in English (Silverman et al., 1992). This revolves around a grammar for combinations of high and low tones and gives rise to a similar set of questions addressed in the present paper. One recent study, for example, used functional data analysis (FDA) to analyze the pitch contours of falling and rising intonation types in English.[8] Although the authors do not explicitly test for this, as we do here, the results suggest that these contours form clusters (Zellers et al., 2010). Analogous to this paper, the authors move from a discrete analysis (ToBI) of intonation contours to a continuous one (FDA). A more recent study Gerazov and Wagner (2021) uses *t*-SNE to visualize intonation contours, and an obvious next step would be to apply the clusterability methods developed in this paper to verify whether those contours indeed cluster. More generally, this convergence calls for an interdisciplinary study of contour in speech and song.

[8] This seems comparable to using a cosine contour representation, assuming that cosines indeed approximate the principal components, as is the case for melodic contours.

SUPPLEMENTS **D1** Experimental setup · **D2** Clusterability of contours · **D3** Length-wise analysis · **D4** Average shapes

DATA AND CODE Data and code have released via github.com/bacor/shapes-of-music

REFERENCE The majority of this chapter has not been presented elsewhere, only the work on the hidden tolerance parameter in Huron's typology been presented at the Locomus meeting in 2021: Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2021a). *Fixing Huron's typology* (poster). Locomus Meeting 2021. http://locomus.net/locomus21/abstracts/.

AUTHOR CONTRIBUTIONS BC designed the research, implemented the experiments, and wrote the paper while being supervised by WZ, JAB, and, at a later stage, HH.

CHAPTER

9

Interlude

Melody squares

Melodic contour is a superficial phenomenon and deliberately so. It abstracts away from individual pitches to describe only the general movement of a melody. In this interlude, I would again like to zoom in on the pitches that underlie a contour using the same lens as in chapter 7. There we broke down rhythms into smaller motifs and visualized these in a rhythm triangle. Now we ask if we can use the same approach to visualize the melodic motifs that are present in a given corpus of melodies.

9.1 Plotting a plot

Rhythmic motifs of four onsets can be plotted in a triangular space only when the total duration of the motif has been normalized, and the last interval is completely determined by the first two intervals. But it is not clear how such a construction would extend to melodies. Whereas time only moves forward, pitch moves both up and down, and so there is no obvious equivalent of the *duration* of a motif. One could try to normalize motifs using the pitch interval between the first and final note, or perhaps between the highest and lowest one, but both seem rather unnatural. Instead, we will consider smaller motifs of three pitches. These form only two pitch intervals and can be visualized naturally in a *phase plot* that shows the first interval horizontally and the next interval vertically.

To visualize a melody in a phase plot, we break it down into a sequence of overlapping motifs and plot each motif in phase space. Figure 9.1 illustrates this for the opening phrase of *Ay mi! dame de valour*, a so-called *virelai* by the French composer Guillaume de Machaut (c. 1300–1377). The song opens with an outcry—*Ay mi!*— whose sharp drop of a major sixth makes for a rather unusual motif. Most motifs later in the melody indeed lie closer to the center of the space. Some motifs even occur multiple times,

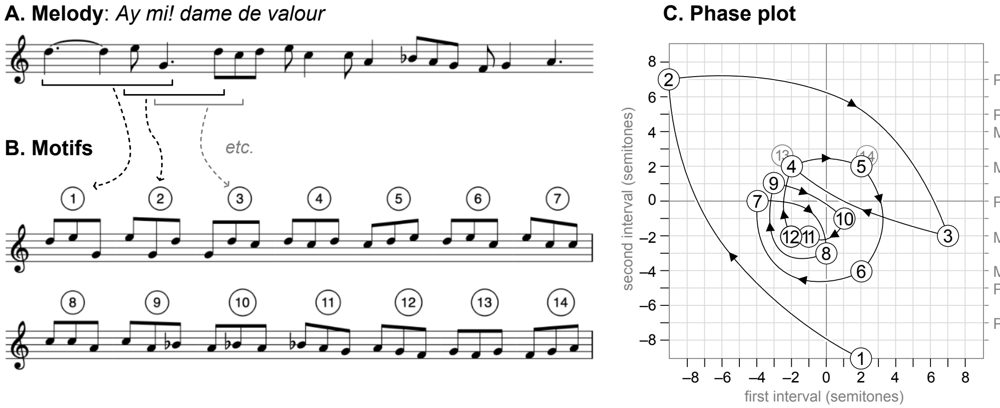


FIGURE 9.1 – Phase plot of a melody. A melody **(A)** is broken down into overlapping motifs of three pitches or two intervals **(B)**. The phase plot **(C)** shows the first interval in a motif horizontally against the second interval vertically. The trajectory formed by all motifs (numbered) is shown in the phase plot. Note that motifs 13 and 14 are the same as 4 and 5.

like motifs 13 and 14, whose intervals are identical to motifs 4 and 5. And that is precisely what I want to look at in this interlude: which parts of the phase space are most frequently visited by a collection of melodies? How frequent are different motifs in a given corpus?

But first, it will be helpful to take a closer look at the space and how it is structured. Figure 9.2 shows that each quadrant contains motifs with a particular contour: moving clockwise from the top left, one finds concave, ascending, convex, and descending motifs. Motifs on the vertical axis *start* with a repetition, while those on the horizontal axis *end* with one. Next, the interval between the first and final pitch—I will call this the *span* of a motif—is identical in all motifs that fall on one diagonal line running from the top left to the bottom right. For example, the motif $(4, -2) = (+\text{M3}, -\text{M2})$ that moves up a major third and then down a major second falls on the same diagonal as the motif $(-3, 5) = (-\text{m3}, +\text{P4})$, and both span a major second.[1] I will call the main diagonal from the bottom left to the top right the *antidiagonal*.

Visualizing motif frequencies in this space using a scatter plot, as Figure 9.1 perhaps suggests, is not an option. I will be looking at musical scores in which the set of possible intervals is discrete, and so most points will overlap. One could *jitter* the points: add some Gaussian noise so that the points form small blobs. Sometimes the sizes of these blobs are clear at first sight but scatter plots often suffer one of two problems—certainly, the triangles in chapter 7 did. Either you plot too many points on top of one another (*overplotting*), or you make the points too small to be visible at all (*underplotting*). Both problems prevent you from seeing the *density* of the data accurately. Fortunately, visualizing the density is easy in this case: we color each grid cell to show how frequent the corresponding motif is.

The color coding does require some attention. Highly frequent motifs will be colored so much darker than the rest that it becomes hard to see differences between infrequent motifs. It might make sense to discard the most frequent items, but a more principled solution scales the colors in a logarithmic fashion. Now one however faces the opposite problem:

[1] I use intervals and their sizes (in semitones) interchangeably:

Sym.	Name	Size
m2	minor second	1
M2	major second	2
m3	minor third	3
M3	major third	4
P4	perfect fourth	5
TT	tritone	6
P5	perfect fifth	7

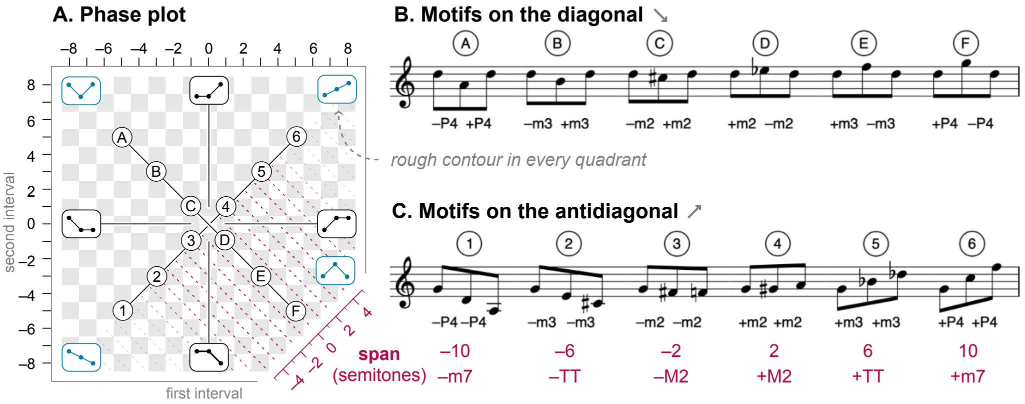


FIGURE 9.2 – A guide to the melody phase plot. (A) Each quadrant contains motifs with a different contour (blue boxes). Motifs on the vertical and horizontal axes either start or end with a repetition (black boxes). On the diagonal **(B)** one finds motifs that start and end on the same pitch (examples A–F). The antidiagonal **(C)** contains motifs that take two identical steps (examples 1–6). Finally, motifs on a diagonal (dashed red lines) have the same *span*: the interval between the first and final note.

low-frequency items can start to dominate the visualization. One unique motif among, say, 10^6 motifs stretches the color scale to 10^{-6} if it has to describe all motif frequencies. As I am interested in the more common motifs, I will cut off the color scale at 0.001: less frequent motifs will all get the same color, while those that are completely absent from the data are masked to remain white. This cut-off point depends on the part of the space that is actually shown: a little more than a perfect fifth up and down in this case.

To summarize, the idea is to visualize the frequencies of three-note melodic motifs in a two-dimensional phase space that I will call the *melody square*. In more technical terms, it simply plots the bigram log-frequencies of pitch intervals. And precisely because of its simplicity, I expect a melody square to be insightful.

9.2 Commonalities and rarities

I produced melody squares for 22 corpora which I had readily available in Catafolk (see chapter 3): Chinese folksongs from the Essen Folksong Collection (Schaffrath, 1995), from which I also took three corpora of German folksongs[2]. Then I included nine Native American corpora[3] from the Densmore collection (Shanahan & Shanahan, 2014), songs from Nova Scotia,[4] and some corpora encoded by Damien Sagrillo with songs from Ireland,[5] Scotland,[6] Germany,[7] and Luxembourg. Finally, I included a corpus of 31 Tsimané songs, recorded by Jürgen Riester (1978). Before we show all melody squares individually, let's look at some averages.

[7] Pinck, 1926, 1928, 1933, 1939.

[6] Haydn, 1792.

[5] O'Boyle, 1976; O'Sullivan, 1981.

[4] Creighton, 1932.

[3] Densmore, 1913, 1918, 1922, 1929b, 1932, 1939, 1943, 1957, 1958.

[2] Böhme, 1877, 1895; Erk and Böhme, 1893a, 1893b, 1894.

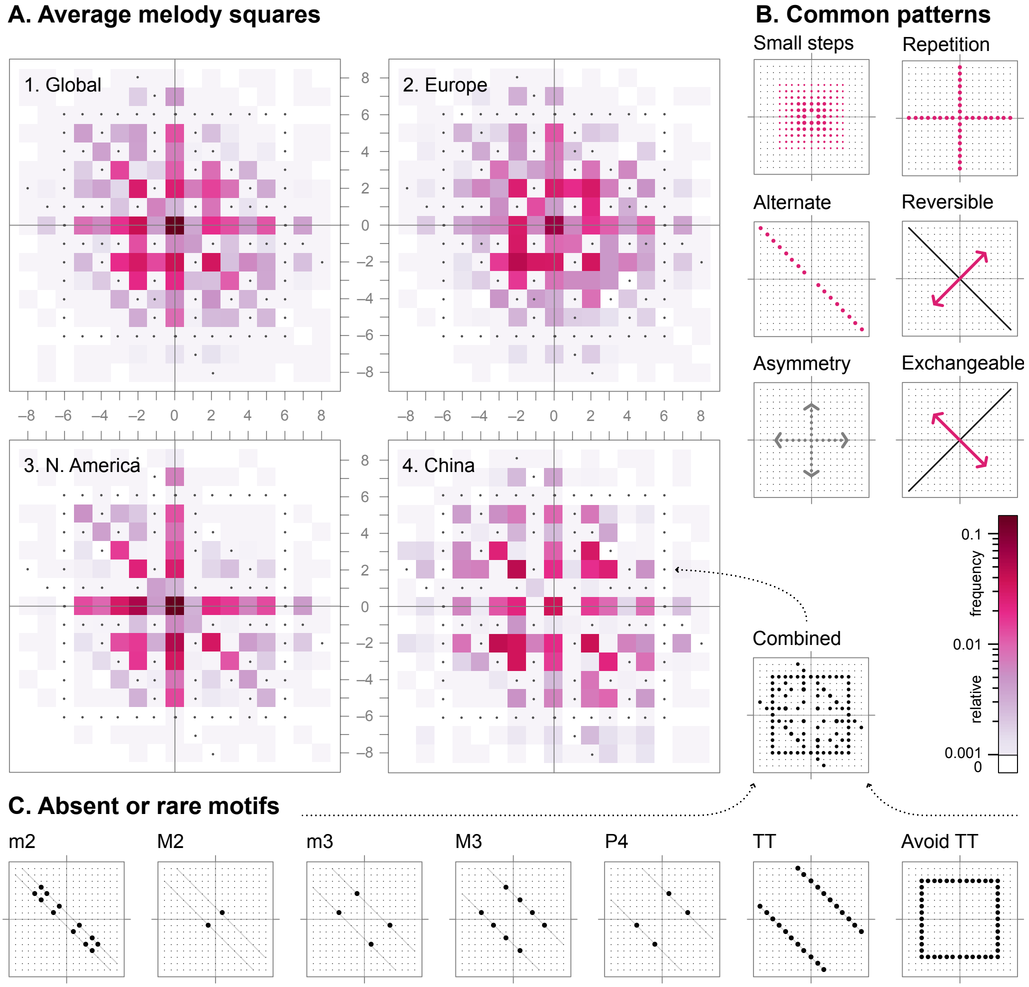


FIGURE 9.3 – Melody squares reveal common and rare melodic motifs in corpora from three geographical areas. The plots in **(A)** show the log-frequency of two-interval motifs in (1) all corpora combined, (2) European, (3) North American, and (4) Chinese corpora only. This reveals several common patterns **(B)**, which are discussed in the main text. It also reveals which motifs are rare (black dots). Panel **(C)** orders these by their span to show that the corpora avoid spanning intervals in particular ways. The second of these plots for example shows that motifs spanning a major second (M2) up or down rarely consist of two successive minor seconds (m2). Tritones (TT), finally, appear to be avoided altogether.

Figure 9.3 shows a global melody square based on all corpora next to melody squares for the European, North American, and Chinese corpora separately. I will sketch some common tendencies and rarities in these melody squares. These observations should not be read as established empirical claims but as hypotheses yet to be rigorously tested. The first thing to notice is that we mostly see *small steps*: the most frequent motifs all lie in the center of the square. This is indeed a commonly cited universal tendency (e.g., Brown & Jordania, 2011; Savage et al., 2015) Second, we see that *repetitions* tend to be common, as motifs on both axes often have a relatively high frequency. Third, so are what one might call *alternations*: motifs on the main diagonal that jump to another pitch and then move back to the first pitch.

Fourth, the squares appear to be mirrored in the main diagonal, which means that the motif (x, y) tends to be as frequent as $(-y, -x)$. Musically, this means that motifs are *reversible*: For a motif $(5, -2)$ that goes *up* a fourth and then *down* a minor second, there is an equally frequent reverse motif $(2, -5)$ that goes *up* a minor second and then *down* a fourth. Fifth, squares also appear to be mirrored in the antidiagonal: (x, y) and (y, x) are roughly equally frequent. For example, the motif $(5, -2)$ spans a minor third via a fourth up and a whole tone down, and the symmetry suggests that there will be equally many motifs spanning a minor third by first moving down a major second: $(-2, 5)$. In that sense, motifs are *exchangeable*. These two are curious, and not always perfect: in the North American square, the top right quadrant does for example not mirror the bottom left one in the main diagonal, thus violating reversibility. But the pattern seems apparent enough to deserve further study. All the more, if one considers that some other symmetries are clearly absent: sixth, the squares are *asymmetrical* in the horizontal or vertical axis.

Seventh, many motifs are commonly absent, which means that the squares are *not convex*. I have organized the rare motifs by their span in Figure 9.3c to highlight that these corpora systematically avoid spanning particular intervals in certain ways. You for example rarely find motifs spanning a minor second, either up or down, with a major second. Similarly, major seconds are usually not spanned by two minor seconds: the use of successive semitones, in short, is rare. In the same spirit, minor thirds are not often spanned using major thirds, nor do these corpora approach major thirds via minor thirds or fourths. The last observation, however, does not hold for the European square where the motif $(5, -1)$ *is* in fact quite common. Motifs that either include or span a tritone, finally, appear to be avoided altogether.

9.3 A tree of squares

Besides commonalities, Figure 9.3 also reveals differences between the corpora. The North American square has an upper quadrant which is relatively empty compared to the other squares. In a previous chapter, we observed that songs of the Lakota on average have a strongly descending contour (see Figure 6.4), and indeed the infrequent quadrant contains precisely the motifs with an *ascending* contour. The European square stands out from the others by its more frequent use of minor seconds, even in motifs with a larger span, while the Chinese square largely avoids minor seconds. These observations suggest that differences in musical

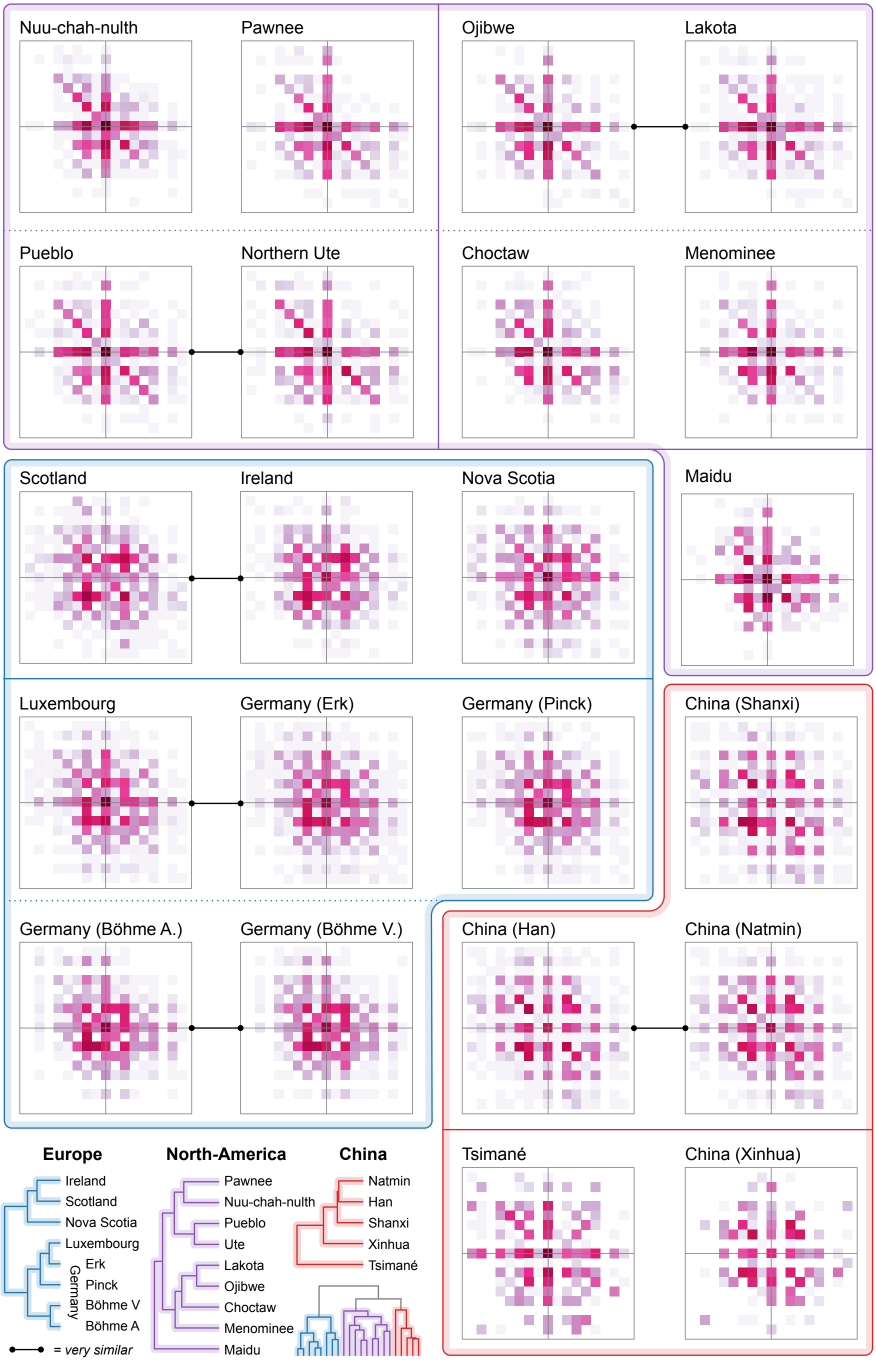


FIGURE 9.4 – **Melody squares allow corpora to be classified to their area of origin.** See the main text for details.

style are reflected in melody squares. And so one wonders: can you turn this around and use melody squares to measure style similarity?

To find out, I looked at slightly larger melody squares ranging from −12 to 12 semitones along both axes,[8] and measured pairwise distances between all those squares.[9] Next, I applied hierarchical clustering to the obtained distance matrix, where I measured the distance between two clusters as the distance between their furthest members. This grouped the 22 corpora in a tree, shown in the corner of Figure 9.4. The tree has three main branches that largely correspond to the European, North American, and Chinese corpora. The squares in Figure 9.4 were manually organized to reflect this clustering structure. The three groups are outlined in different colors, and squares that are neighbors in the tree are connected by black lines.

If we interpret the three branches as broad areas of origin and allow the songs from Nova Scotia to be grouped with the European corpora, only the Tsimané corpus is clearly misclassified. It is however quite distant from the other corpora in its branch, as can be seen from the branch length. Within the European group, German corpora cluster closely together, as do songs from Ireland, Scotland, and, to a lesser extent, Nova Scotia. In the Native American group, the Ute appear close to the Pueblo peoples, with which they have indeed been in cultural contact, The Lakota and Ojibwe are similarly from geographically close areas. But the tree suggests that the Pawnee and Nuu-chah-nulth are also quite similar melodically, even though the former have lived around Nebraska, while the latter live on Vancouver Island.

9.4 Conclusions

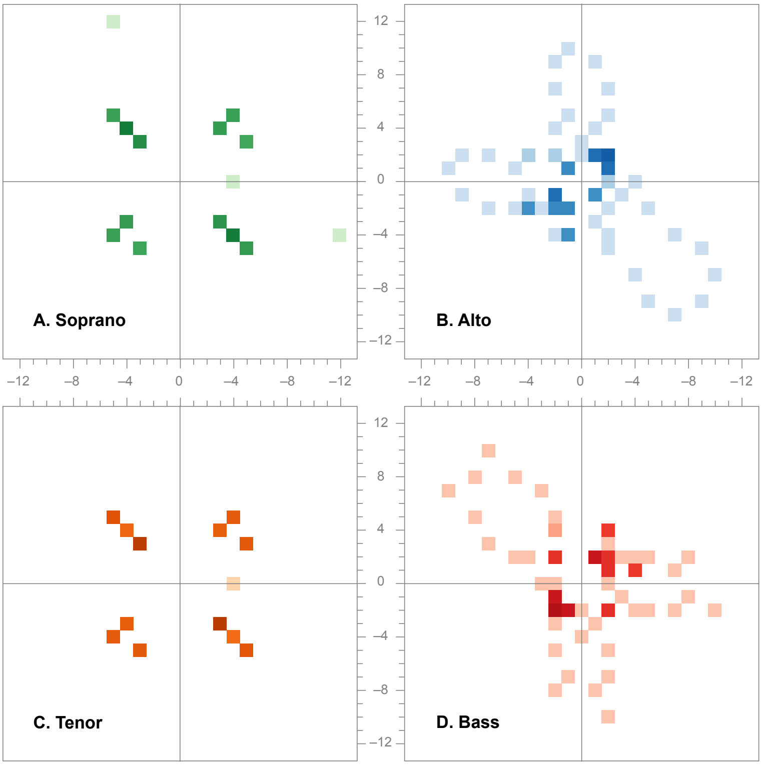
In this interlude, I looked for a melodic equivalent of the rhythm triangle and proposed the melody square. It shows the relative frequency of melodic motifs of three pitches in a two-dimensional phase plot. The squares show which motifs are common and rare across multiple corpora and revealed some interesting generalizations. But what *explains* the patterns we observed, for example in Figure 9.3? This will be left for future research, but perhaps an explanation can be found in the scales that are used. For example, if European songs often use scales with a semitone step, while Chinese corpora prefer pentatonic scales without minor seconds, that could explain why motifs with a minor second are more frequent in the European melody squares.

Where I looked at actually recorded rhythms in chapter 7, this interlude only analyzed musical scores. This was a pragmatic choice: Catafolk made all these different corpora readily available to me. But it should be possible to extend this approach to continuous pitch recordings. One could, for example, average the pitch estimate of each individual note in a recording and visualize the intervals between them as a scatter plot in phase space. This would require both reliable pitch estimates and accurate annota-

[9] I measured the distance between squares A and B by $\sqrt{\sum_{i,j} |a_{ij} - b_{ij}|^2}$.

[8] Using two octaves on both sides did not change the results.

FIGURE 9.5 – Melody squares for Arvo Pärt's *Summa*. Works by Pärt are often constructed according to numerical procedures or inspired by geometric shapes. Some of the constructions underlying *Summa* appear to be reflected in the melody squares of the four voices A–D. Unraveling the regularities in *Summa* is the topic of chapter 10. Note that these squares were produced while ignoring all ornamental notes.



tions of note onsets and offsets. Such datasets are indeed available. The *Erkomaishvili Dataset*, to name just one interesting example, contains transcriptions, pitch contours, and note annotations of polyphonic Georgian vocal music (Rosenzweig et al., 2020).

If continuous melody squares prove fruitful, one could even move on to visualize animal sounds in a similar fashion: complement the zoo or rhythm triangles with a zoo of melody squares. A visual compendium showing how different musics, or even animal sounds, from around the globe organize their melodic movements. And then one might find one type of music to stand out: the music of Arvo Pärt. The strange symmetries in the melody squares of his piece *Summa* (Figure 9.5) are the product of strict regularities hidden beneath the surface of his music. What is going on here—how does Pärt's music work? Time to move on to the next chapter.

DATA AND CODE The data and code for all results in this interlude, together with all figures, have been made available publicly online via github.com/bacor/melody-squares. My digital transcriptions of the Tsimané songs have not been made available for copyright reasons.

REFERENCE The work in this chapter has not been published or presented elsewhere.

AUTHOR CONTRIBUTIONS BC designed and conducted the research and wrote this chapter.

ACKNOWLEDGEMENTS I would like to thank Jelle Zuidema and Ashley Burgoyne for their comments on the first version of this chapter and Henkjan Honing for making me realize that version needed two more figures.

CHAPTER

10

Article

Algo Pärt

Arvo Pärt is one of the most popular contemporary composers, known for his highly original *tintinnabuli* style. Works in this style are typically composed according to precise procedures and have even been described as algorithmic compositions. To understand how algorithmic Pärt's music exactly is, this paper presents an *analysis by synthesis*: it proposes an algorithm that almost completely reconstructs the score of *Summa*, his "most strictly constructed and most encrypted work," according to Pärt himself in 1994. The piece is analyzed and then formalized using so-called tintinnabuli processes. An implementation of the resulting algorithm generates a musical score matching Summa in over 93% of the notes. Due to interdependencies between the voices, only half of the mistakes (3,5%) need to be corrected to reproduce the original score faithfully. This study shows that *Summa* is a largely algorithmic composition and offers new perspectives on the music of Arvo Pärt.

10.1 Introduction

Music and algorithms share a long history, but rarely has their marriage been as fruitful as it has been in the hands of the Estonian composer Arvo Pärt. According to one study, Pärt was the most frequently performed contemporary composer from 2011 until 2019.[1] Not only is his music popular, but it is also highly original. In the 1970s, Pärt developed a unique compositional technique, known as *tintinnabuli*, that is deeply algorithmical due to its use of numerical procedures. The main melody may, for example, walk down a scale, moving one step further with every measure. Alternatively, it may be determined by the text: in his *Missa Sillabica*, the number of syllables in a word determines the melody for that word. Examples such as these raise the question *how* algorithmic Pärt's music precisely is. Can all notes in a score be explained by formal procedures? And when does the composer deviate from those, if at all?

To address such questions, I propose a type of computational music analysis (cf. Anagnostopoulou & Buteau, 2010) that one could call *analysis by synthesis*. Motivated by the idea that one cannot understand what one cannot create, the aim is to implement an algorithm that reconstructs as much of a score as possible. By measuring the *reconstruction error*, the number of errors in the reconstructed score, one can evaluate the algorithm. In practice, such an analysis is an iterative process in which one successively refines the rules to further reduce the reconstruction error. As the error decreases, the explanatory power of the algorithm increases, until adding new rules no longer seems to be theoretically productive. Adding a rule that explains only a single note, for example, is not very productive and similar to "overfitting" a mathematical model. But up to that point, the algorithm provides an answer to a central question of musical analysis: how does the piece work?

The idea of using algorithms to analyze Pärt's music is not new.[2] Shvets (2014) describes multiple constructions commonly found in the work of Pärt using concepts borrowed from programming languages, such as loops. Shvets and De Paiva Santana (2014) then went on to implement several models of Pärt's compositions. In a more formal analysis, Roeder (2011) proposes to understand Pärt's compositional procedures as musical transformations (cf. Lewin, 1987). His analysis effectively results in several (functional) programs that model certain aspects of Pärt's music. This paper takes these ideas one step further by first formalizing a piece, then implementing an algorithm to reconstruct the full score, and finally quantitatively evaluating that against the original: a complete analysis by synthesis.

Notes can be found at the end of this chapter.

Our case study looks at *Summa*. This piece was written in 1977, one year after Pärt wrote his first piece in tintinnabuli style (*Für Alina*). *Summa* is best known as a composition for mixed choir or solo voices but was originally written for two voices (tenor and bass) and six instruments (Hillier, 1997). It has since been adapted for many instrumental combinations, from string quartet to trombone quartet. The composition is intricately

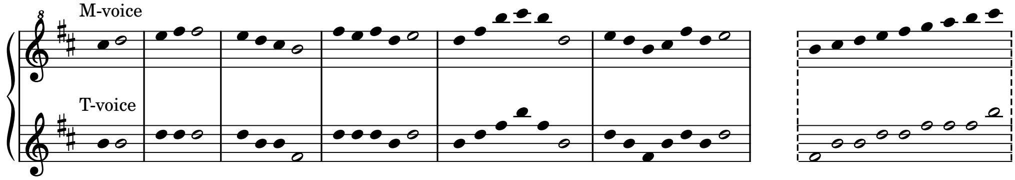


FIGURE 10.1 – **Excerpt of *Für Alina* (mm. 2–7).** This piano piece was Arvo Pärt's first work in his tintinnabuli style. The right hand plays a melodic voice (M-voice) that mostly moves stepwise, and which is freely composed. The left hand has the tintinnabuli voice (T-voice), which is restricted to notes from the B-minor triad. The relation between the two voices is shown on the right: the T-voice plays the highest triad note below the M-voice, but one octave lower.

structured, but many of the underlying regularities will escape notice when listening to a performance, or even when studying the score. Indeed, some of the procedures identified in this paper seem to have escaped previous analyses of *Summa* (Hillier, 1997; de la Motte-Haber, 1996; Patrick, 2011). Arvo Pärt may have anticipated this when he wrote:

> I have developed a highly formal compositional system in which I have been writing my music for 20 years. In this series, *Summa* is the most strictly constructed and most encrypted work. The encryptions are found in many layers of the score.[3] (Pärt, 1996)

I read this as an invitation to decrypt *Summa*. But first, let me introduce the tintinnabuli style and the terminology that Hillier (1989, 1997) developed to describe it–some of which Pärt himself has adopted.

10.2 Tintinnabuli

At the heart of the tintinnabuli style lie two voices: a melodic voice or *M-voice* and an accompanying tintinnabuli voice or *T-voice*. The M-voice is usually diatonic and tends to move in steps around a pitch center. It is sometimes freely composed, but more often constructed according to numerical procedures. The accompanying T-voice is even more constrained. It can only use notes from a central *tintinnabuli triad*, and is determined by its M-voice according to a strict procedure, such as always using the first note in the triad above the M-voice. The resulting texture is often homophonic, further emphasizing the unity of the two voices.

For Pärt, the M- and T-voice are indeed much more than a compositional technique. The M-voice "signifies the subjective world, the daily egoistic life of sin and suffering; the T-voice, meanwhile, is the objective realm

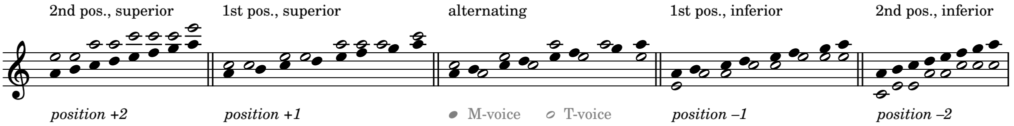


FIGURE 10.2 – Tintinnabuli positions for an A minor triad. Solid notes show the A minor scale as a melodic voice, and open notes show tintinnabuli voices in the five different positions introduced by Hillier (1997). His terminology is shown above the staff, the numbering used in section 10.4 below it.

of forgiveness" (Hillier, 1997, p. 96). Their duality is only appearance: they really are a "twofold single entity," as has been summarized in the equation $1 + 1 = 1$.

To clarify the terminology, let's consider two famous examples (see e.g., Hillier, 1997 for more extensive analyses). Figure 10.1 shows an excerpt of the piano piece *Für Alina* (1976). The piece is composed around the B minor tintinabulli triad, and the tonal center of B is reinforced by a low pedal note not shown in this excerpt. The right hand plays the M-voice, and the left hand plays the T-voice in the same rhythm, using only notes from the tintinnabuli triad. The relation between the two voices is simple: the left hand plays the highest triad note below the melody, but one octave lower. Pärt deviates from this only once, when the T-voice plays a C♯ (in bar 11; not shown). This special event is marked with a flower in the original score. In other pieces, the T-voice consistently picks the second triad note above the melody or alternates the one above and below it. Hillier (1997) called such configurations *tintinnabuli positions*. As illustrated in Figure 10.2, he distinguishes two *superior* and two *inferior* positions, which use triad notes *above* and *below* the M-voice respectively. Tintinnabuli positions do not change when transposing them octaves up or down, and *Für Alina* therefore uses a T-voice in first position inferior.

The melody of *Für Alina* seems to be more freely composed than the melodies in many of his other works. Still, it follows a numerical regularity: every measure adds another quarter note until the pattern flips midway and measures become shorter and shorter again. We find an even more systematic melody in *Fratres* (1977).[4] The piece is built around an A minor tintinnabuli triad and has two M-voices moving in parallel tenths. It is unusually dissonant, as the melodies move along a D harmonic minor scale, which includes a C♯ instead of the C♮ from the triad. Figure 10.3 illustrates the backbone of *Fratres*: nine variations on a six-measure theme, with each variation lowering the pitch center by another third. In the first half of the theme, the melody moves down from the pitch center, and then approaches it from above, moving one step further every bar. The second half of the theme repeats the first half retrogradely. This results in four types of melodic movement that Hillier (1997) also frequently encountered in other works of Pärt. Figure 10.4 summarizes these four melodic *modes*:

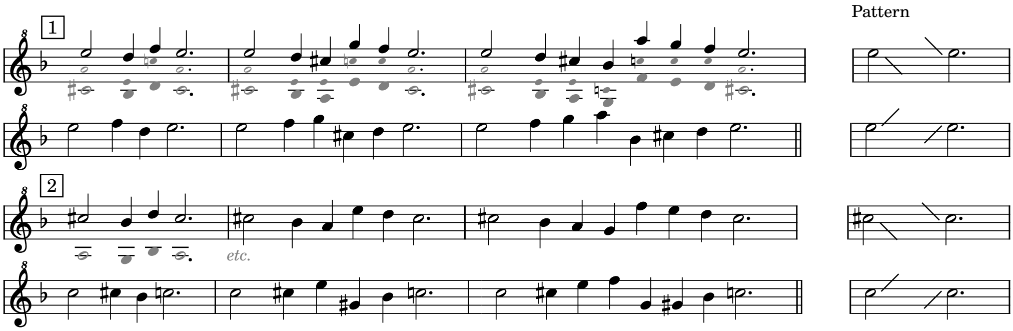


FIGURE 10.3 – The melodic structure of *Fratres* is a series of variations of a six-bar theme. The first two variations are shown. The theme leaves and then approaches a pitch center, moving one step further for three consecutive bars. The next three bars repeat the first three, but are played backward. This results in four types of melodic movement, or *modes*, that are often used by Pärt to compose M-voices. The first staff also shows the parallel M-voice and the T-voice as small notes.

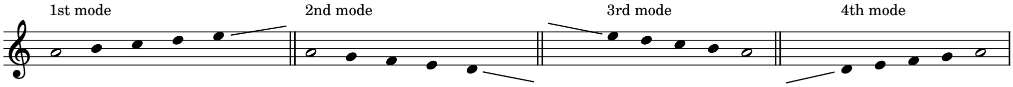


FIGURE 10.4 – Four melodic modes commonly used by Arvo Pärt to construct M-voices. Two modes move away from a central pitch, and two approach it (Hillier, 1997). All of these are found in *Fratres*, as shown in Figure 10.3.

moving (1) up or (2) down *from* a central pitch, or moving (3) down or (4) up *towards* it.

The context in which Arvo Pärt developed his tintinnabuli style, and its broader interpretation, has been discussed extensively in the scholarly literature (see e.g., Bouteneff et al., 2021; Hillier, 1989; Shenton, 2012). It emerged during a period of seven years in which he studied early music, from plainchant to Palestrina, after his earlier serialist style had come to a creative halt. Musically, as Hillier (1997) also explains, tintinnabuli contains elements from early polyphony, functional harmony, and serialism. The stepwise motion in the M-voice is for example reminiscent of plainchant, and the homophonic texture it forms with the T-voice can be compared to early polyphonic chant settings. The tintinnabuli style also returns to a form of tonality, but not a functional one. While the hallmark of function harmony, the triad, is omnipresent in tintinnabuli, it has been stripped of its functional role. It is remarkable how Pärt managed to fuse all these different ideas into a musical style that appeals to audiences around the world.

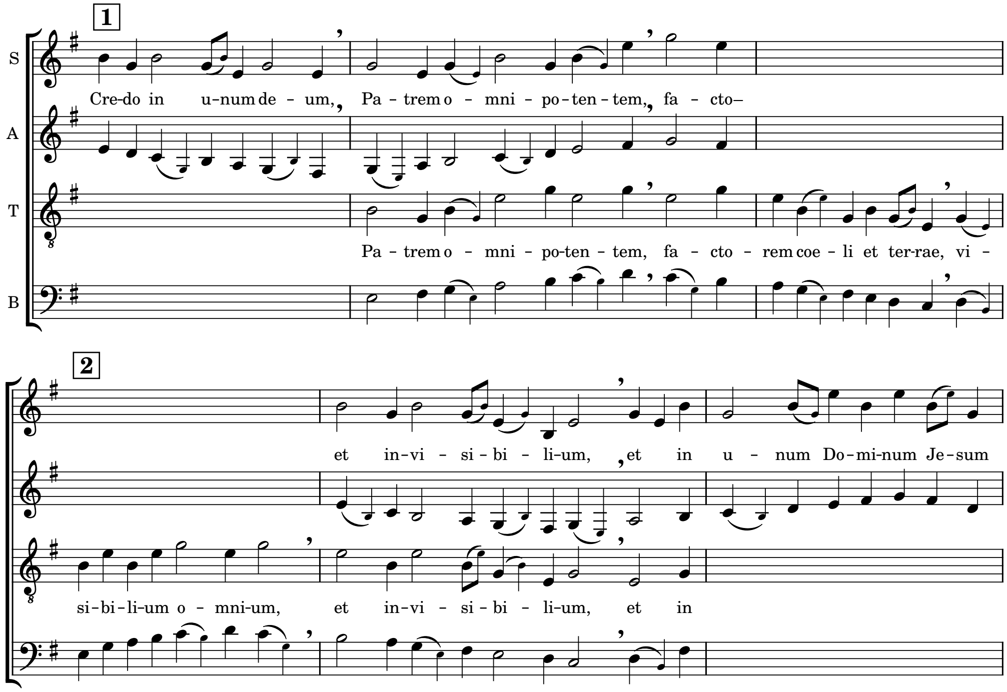

1
S
A
T
B
Cre-do in u-num de - um, Pa - trem o - mni - po-ten - tem, fa - cto–
Pa - trem o - mni - po-ten - tem, fa - cto - rem coe - li et ter-rae, vi -
2
et in-vi - si - bi - li-um, et in u - num Do-mi-num Je-sum
si-bi-li-um o - mni-um, et in-vi - si - bi - li-um, et in

FIGURE 10.5 – The opening measures of *Summa*. The piece is a setting of the *Credo*, consisting of 16 three-bar sections with 7, 9, and 7 syllables. The voice distribution is mirrored in every section: SA–SATB–TB. The alto and bass have the M-voices, the soprano and tenor the corresponding T-voices. If one skips the small, slurred ornaments, the M-voices walk up and down an E natural minor scale, while the T-voices are constrained to the E minor triad.

10.3 Analysis

We now turn to *Summa*, of which several analyses have been published before. The first,[5] de la Motte-Haber (1996), preceded Hillier's monograph on Pärt and misses some key points: it focuses on T-voices rather than the M-voices and discusses the version for string quartet, in which one cannot see how *Summa* is structured around the text of the Credo, the Christian statement of belief. Hillier (1997) points out that syllables in fact form the 'units' of the piece and goes on to reveal the structure of the M-voices. But their relation to the T-voices remains unclear: although their overall contours correspond, he writes that the "note-to-note logic of the T-voice is, exceptionally, self-contained" (p. 112). In an even more extensive analysis, Patrick (2011) does not resolve this issue either. The explanation I will propose below indeed moves beyond a note-to-note logic and describes the T-voices as tintinnabuli *processes* that also depend on previous notes in the M- and T-voices.

But let's start at the beginning. Hoping to make the text more accessible, I first present an analysis and then a formalization, even though the two developed in tandem and often overlap.

TEXT AND STRUCTURE Figure 10.5 shows the opening bars of *Summa* in the version for mixed choir, which I analyze here.[6] The first thing that stands out is the overall organization. *Summa* is divided in 16 sections spanning three measures each. The first and final bars of a section are sung by the highest (SA) or lowest (TB) two voices, the middle bar is tutti, and this pattern is mirrored in the next section. The organization becomes transparent when observing that syllables are the unit of time. The Credo consists of 366 syllables, which Pärt evenly distributed over the 16 sections. Each section contains 23 syllables, divided over three measures of 7, 9 and 7 syllables respectively (see supplement E1). That amounts to a total of 368 syllables, two more than found in the Credo. The final two bars are composed slightly more freely to compensate for this, but as a result break some of the regularities seen in the rest of the score (see supplement E6).

The text setting is homophonic and largely syllabic: most syllables are sung on a single note, some on two notes. Pärt always slurred those two notes, and we can think of the second one as an *ornament* or passing note (cf. Hillier, 1997). This distinction between ordinary *notes* and *ornaments* will be important. The text setting is "fortuitous" (Hillier, 1997) insofar that it is dictated by the numerical patterns that Pärt laid out, not by the text itself. This can be seen in the second bar, where the alto and soprano end without ever finishing the word "factorem". The phrasal structure of the text is maintained in the music and indicated by commas, but these do usually not overlap with bar lines. A notable exception, as Hillier (1997) points out, is the very first phrase: "Credo in unum Deum." Its seven syllables may well have inspired the larger structure.

MELODIC VOICES *Summa* has two melodic voices: the alto and the bass. The opening bar makes clear that the soprano is the T-voice for the alto, and that the tenor forms a pair with the bass. Both T-voices only sing notes from an E minor triad, and the M-voices only use the E natural minor scale, making *Summa* completely diatonic. The E natural minor scale also forms the backbone of the M-voices. To visualize this, Figure 10.6 plots only the notes of the alto and bass, and ignores ornaments and note durations.[7] The blue line highlights that the alto is basically walking up and down the E natural minor scale. The bass exactly mirrors the alto, but has rests in different places. Closer inspection shows that the alto is repeating a fifteen-note pattern, which is interrupted by bars of silence and a return to the tonic whenever it enters, or when a new section starts. As a section contains sixteen syllables, the fifteen-note pattern starts at a different point in every section: it is shifted one step to the left. And so one can alternatively describe the alto as follows: every section starts with the tonic, followed by the pattern, but rotated one more step to the left (cf. Hillier, 1997; Patrick, 2011). Both accounts have the same result, and explain the feeling that the piece could continue forever, were it not for the final two bars.

TINTINNABULI VOICES The tintinnabuli voices in *Summa* have probably puzzled scholars most, even when they have ignored ornaments. Their relation to the M-voices is not as direct as in *Für Alina* or *Fratres*, where the T-voice consistently takes a fixed tintinnabuli position. For example, while the first C_4 of the alto is paired with a B_4 in the soprano, the third time

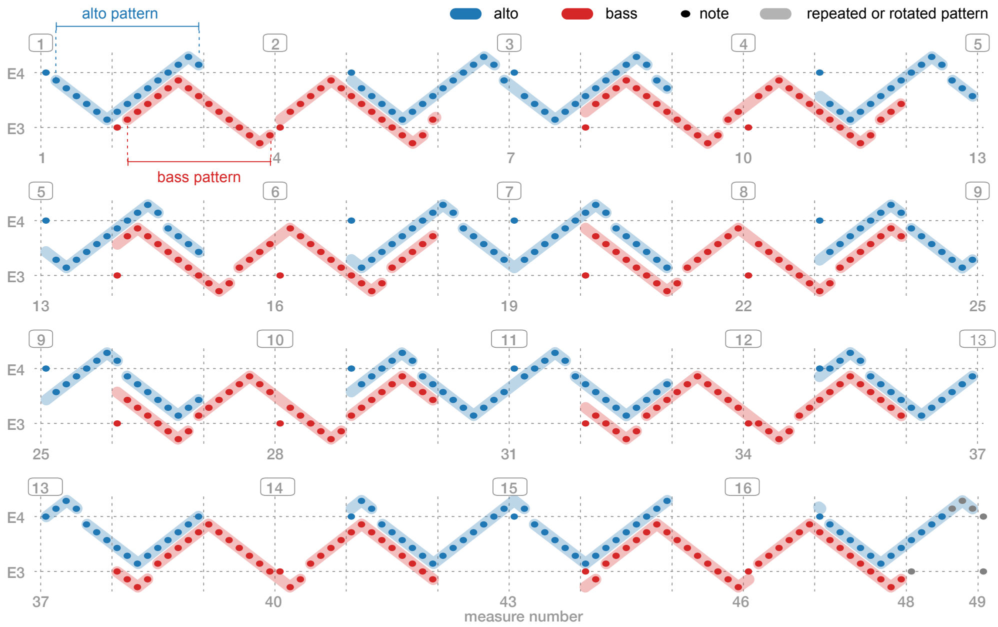


FIGURE 10.6 – The melodic voices each repeat a 15-note pattern that walks up and down a scale. Diatonic pitch is shown vertically and time horizontally, measured in syllables. The notes are shown as dots and ornaments have been omitted. The repetitions of the underlying 15-note pattern are shown in the background. In every section (marked by rehearsal numbers), a voice sings the tonic (E) followed by this pattern, but rotated one step to the left.

we encounter the C_4 in bar 5, the soprano sings a G_4. Both Hillier (1997) and Patrick (2011) conclude that the T-voices only resemble the shapes of the M-voices, but are not predictably related to it. The T-voices indeed cycle through a 30-note pattern that is similarly shaped as the M-voices but with slight variations in each repetition. Still, there appears to be an underlying logic. To identify it, I overlaid all repetitions of this 30-note pattern and worked out an *approximate pattern* that best approximates all of the repetitions (see supplement E2). Except for a few notes, the approximation will thus be the same as each individual repetition in the score.

Figure 10.7 shows the approximate patterns and reveals the constraints that determine the T-voices. First, the soprano is at least two triad notes above the alto and the tenor at least one triad note above the bass. Second, the T-voices only move step-wise to neighboring triad notes (repetitions are not allowed). It turns out that one obtains the T-voices by picking the lowest note satisfying these constraints at every time step. This also explains some of the slight variations mentioned above: these are caused by the melody voice jumping back to the tonic. And so the T-voices are not in a fixed position but appear to be determined by a *process* that depends on the current melody note and the previous tintinabulli note.

ORNAMENTS Pärt has added ornamental notes to both the M- and T-voices. They are always triad notes, which suggests that we can think of the ornaments as tintinnabuli voices themselves. The approximate patterns in Figure 10.7 show that ornaments are not randomly inserted, but the underlying pattern is hard to pin down. For the soprano and tenor we see

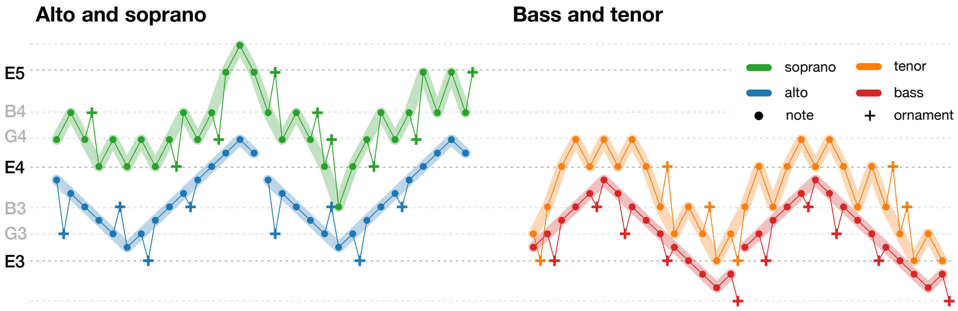


FIGURE 10.7 – Approximate patterns for all voices. These patterns of notes (·) and ornaments (+), when repeated throughout the piece, approximate the melodies of each of the voices. The approximate patterns were constructed manually by comparing all repetitions, so as to make the approximation as good as possible (see supplement E2). For the T-voices, they however remain approximations: they are better understood as functions of the M-voices (Figure 10.8), not as repetitions of the patterns shown here.

that ornaments only occur when the melody moves in the same direction for more than two steps, and the ornament reverses the direction. I see no obvious correspondence between ornaments in the alto and bass pattern. In particular, they are not mirrored, but they do use the same ornaments (E and B) when passing the G and C on the way up. That means that for the alto and bass ornaments, the approximate patterns are the best description we currently have.

RHYTHM Finally, the rhythm in *Summa* is determined by two constraints: first, that syllables start together in all voices (homophony), and second, that melody notes have the duration of at least a quarter note. This implies that if the alto has an ornament where the bass does not, the bass note needs to be twice as long, and vice versa. If a T-voice has two notes where the melody has one quarter, the T-voice has to half both of its notes. These rules are consistently applied throughout the piece, except the penultimate bar, where the bass and tenor start the “Amen” before the alto and soprano.

10.4 Synthesis

We now formalize the construction of *Summa* laid out above, so that we can implement an algorithm to reconstruct the score. Our formalism takes inspiration from Roeder (2011) by distinguishing an M- and a T-space in which the M- and T-voices live. The framework of Roeder (2011) is so general that it even allows for the possibility that the spaces contain objects other than pitch classes. That does not help us here: we need to generate specific pitches, and even pitch *classes* would be too general.

Our formalization therefore starts in a larger pitch space $\mathcal{N}$ that contains all semitones between, say, C_0 and C_8, which are naturally ordered (e.g., $G_2 < A_3$). If we call a subset S that spans no more than an octave a *scale*, we can generate a *scalar pitch space* $\langle S \rangle$: the pitches (in $\mathcal{N}$) with the same pitch *class* as elements in the scale. In this way, we let the E-natural minor scale generate the M-space $\mathcal{M}$, in which the M-voices can move around. The T-voices live in the T-space $\mathcal{T}$, which is generated by the E minor triad–also a scale under this definition. In short,

$$\mathscr{M} = \langle \mathrm{E}_3, \mathrm{F}_3^{\#}, \mathrm{G}_3, \mathrm{A}_3, \mathrm{B}_3, \mathrm{C}_4, \mathrm{D}_4 \rangle \tag{10.1}$$

$$\mathscr{T} = \langle \mathrm{E}_3, \mathrm{G}_3, \mathrm{B}_3 \rangle. \tag{10.2}$$

Both spaces are subsets of $\mathscr{N}$, and $\mathscr{T}$ is moreover a subset of $\mathscr{M}$. But the latter need not be the case: in *Fratres* the triad falls outside of M-space because of the C♮ (Figure 10.3).

MELODIC VOICES We can construct the basic pattern sung by the alto by concatenating fragments of the four melodic modes, or by simply listing its pitches:

$$\alpha = (\mathrm{E}_4, \mathrm{D}_4, \mathrm{C}_4, \mathrm{B}_3, \mathrm{A}_3, \mathrm{G}_3, \mathrm{F}_3^{\#}, \mathrm{G}_3, \mathrm{A}_3, \mathrm{B}_3, \mathrm{C}_4, \mathrm{D}_4, \mathrm{E}_4, \mathrm{F}_4^{\#}, \mathrm{G}_4, \mathrm{F}_4^{\#}). \tag{10.3}$$

The alto sings this 16-note pattern 16 times, but every time rotates the *tail* of the pattern one step to the left: everything after the first note. For a sequence $x = (x_1, \dots, x_N)$, let $\mathrm{Rotate}(x, d) = (x_{i-d \bmod N} : i = 1, \dots, N)$ be its rotation by distance d. Then the tail rotation by distance d can be defined as $\mathrm{TailRotation}(x, d) = (x_1) \smile \mathrm{Rotate}((x_2, \dots, x_N), d)$, where the cup "$\smile$" indicates concatenation. And so

$$\mathrm{alto} = \mathrm{TailRotation}(\alpha, 0) \smile \dots \smile \mathrm{TailRotation}(\alpha, 15) \tag{10.4}$$

gives all notes of the alto. The bass mirrors this. Let $\mathrm{mirror}_{\mathscr{M}}(n, c)$ be the mirror image of n with respect to c: the pitch which is equally many steps (in $\mathscr{M}$) apart from c as n is, but in the other direction. Then write $\mathrm{transpose}_{\mathscr{M}}(n, d)$ for the transposition of note n by d steps. If both these operations work entry-wise on sequences,

$$\mathrm{bass} = \mathrm{transpose}_{\mathscr{M}}(\mathrm{mirror}_{\mathscr{M}}(\mathrm{alto}, \mathrm{E}_4), -6). \tag{10.5}$$

TINTINNABULI PROCESSES The central concept in tintinnabuli music is arguably the tintinnabuli position. Different from Hillier (1997), it will be convenient not to treat positions in octaves as equivalent. Instead, we denote the tintinnabuli note in p-th position above a given note n by $T_p(n)$, and the one below it by $T_{-p}(n)$, and allow p to be any integer. For example, in our case $T_2(\mathrm{A}_3) = \mathrm{E}_4$ since this is the second triad note above A_3, and $T_{-1}(\mathrm{A}_3) = \mathrm{E}_3$. One way to define the function T_p, is as follows:

$$T_0(n) = n \tag{10.6}$$

$$T_p(n) = \min\{t \in \mathscr{T} : \; t > T_{p-1}(n)\} \tag{10.7}$$

$$T_{-p}(n) = \max\{t \in \mathscr{T} : \; t < T_{-(p-1)}(n)\}. \tag{10.8}$$

This definition is recursive: we think of $T_2(n)$ as the first tintinnabuli note above $T_1(n)$, that is, as $T_1(T_1(n))$. The function is defined on all of $\mathscr{N}$, but

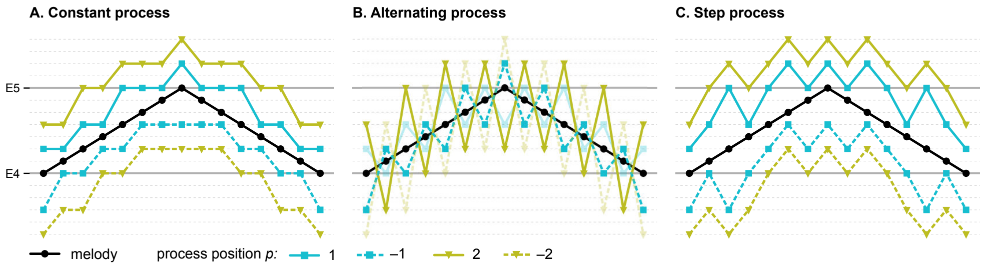


FIGURE 10.8 – Three tintinnabuli processes for a melody that walks up and down a scale. The constant process **(A)** remains in the same tintinnabuli position p, while the alternating process **(B)** flips the sign of the starting position every step. The step process **(C)** only moves to neighboring triad notes, while keeping a distance of at least p triad notes from the melody. This is the process used in *Summa*.

we are most interested in the case when the position p is nonzero, and T_p maps $\mathscr{M}$ to $\mathscr{T}$.

As we have seen, the tintinnabuli voices in *Summa* are not solely determined by the current melody note, but also by previous notes. The same is true for Hillier's alternating position (Figure 10.2). Because of the sequential dependency, I would propose to speak of a "process" instead of a "position" in such cases. To define this formally, consider a sequence of melody pitches $m_1, \dots, m_K$ in $\mathscr{M}$. A *tintinnabuli process* X determines a corresponding sequence of tintinnabuli notes $t_1, \dots, t_K$ in $\mathscr{T}$ via

$$t_i = X\big((t_1, \dots, t_{i-1}),\ (m_1, \dots, m_K)\big), \qquad i = 2, \dots, K. \tag{10.9}$$

Such a process can thus depend on all notes in the melody, past and future, but only on previous notes in the tintinnabuli voice. Of course, we do need to specify the starting point t_1, or else the process cannot start.

The simplest example of a tintinabulli process is one that always returns the same tintinabulli position (Figure 10.8A),

$$\text{Constant}_p(m_i) = T_p(m_i). \tag{10.10}$$

This shows that a position is a special case of a process. A second example would be Hillier's alternating position. Let $P_m(t)$ denote the position of t with respect to note m, that is, the position p such that the p-th tintinnabuli note of m is t, or $T_p(m) = t$. Then the alternating process is

$$\text{Alternate}(m_i, m_{i-1}, t_{i-1}) = T_{-p_i}(m_i), \qquad \text{where } p_i = P_{m_{i-1}}(t), \tag{10.11}$$

which basically flips the sign of the starting position (see Figure 10.8B).

The tintinnabuli voices in *Summa* are determined by a more intricate process that ensures the voices always satisfy two constraints. At every point, t_i has to be (1) at least in position p above the melody note m_i, and (2) one step in the triad apart from the previous note t_{i-1}. And so the process moves stepwise through T-space while staying at least in position p. This *stepwise tintinnabuli process in position p* can be defined as

$$\text{Step}_p(m_i, t_{i-1}) = \begin{cases} T_{-1}(t_{i-1}) & \text{if this is} \geq T_p(m_i) \\ T_{+1}(t_{i-1}) & \text{otherwise} \end{cases}, \qquad \text{for } p \geq 0. \tag{10.12}$$

This process will satisfy both constraints for a stepwise melody as long as the starting point t_1 is at least in position p. Although this would also be defined for $p < 0$, it seems more appropriate to flip the definition for $p < 0$:

$$\mathrm{Step}_p(m_i, t_{i-1}) = \begin{cases} T_{+1}(t_{i-1}) & \text{if this is} \leq T_p(m_{i+1}) \\ T_{-1}(t_{i-1}) & \text{otherwise} \end{cases}, \qquad \text{for } p < 0. \tag{10.13}$$

This resulting process is illustrated in Figure 10.8c.

ORNAMENTATION As the ornaments are always triad notes, we can think of the ornaments as T-voices, but ones that can also be silent (no ornament), besides singing (an ornament). This translates into a hierarchy of T-voices: the tenor ornaments form a T-voice for the tenor, which is a T-voice for the bass. We define a tintinnabuli process that generates the ornaments for both the soprano and the tenor (see supplement E3). The process returns the previous melody note if it does not equal the next melody note (the M- and T-spaces are identical), and while it remains within certain bounds:

$$\mathrm{RepeatPrevious}_{b,B,c,C}(m_{i-1}, m_{i+1}) = \begin{cases} & \text{when } m_{i+1} \neq m_{i-1}, \\ m_{i-1} & \text{and } b \leq m_{i-1} \leq B \\ & \text{and } c \leq m_{i+1} \leq C \\ \text{silent} & \text{otherwise.} \end{cases} \tag{10.14}$$

Without the bounds, the process cannot avoid ornaments at the extremes of the range (e.g., G_5 or B_3 for the soprano), and one can imagine why Pärt might have wanted to avoid those. Although it remains a question whether Pärt actually thought of the ornamentation in this way, the reuse of formal machinery seems appealing.

Finally, the alto and bass have ornaments at fairly regular positions along the 16-note melodic pattern. We therefore define a process that repeats a fixed sequence of ornamental pitches x, which can also contain silences. Since the melodic pattern is repeated with a tail rotation, we also need to rotate x to keep it aligned:

$$\mathrm{TailRotatedPattern}_x(m_i) = r_{i \mod |x|}, \tag{10.15}$$

where $r = \mathrm{TailRotation}(x, \mathrm{floor}(i/|x|))$. The pattern of ornaments we use is illustrated in Figure 10.7 (and in supplement E4).

IMPLEMENTATION To summarize, our formalism describes the notes of the alto as a tail-rotated pattern and the bass as its mirror image. The soprano and tenor are stepwise tintinnabuli processes in second and first position respectively. Ornaments are also described as tintinnabuli processes. We can then insert the notes and ornaments into the measure structure dis-

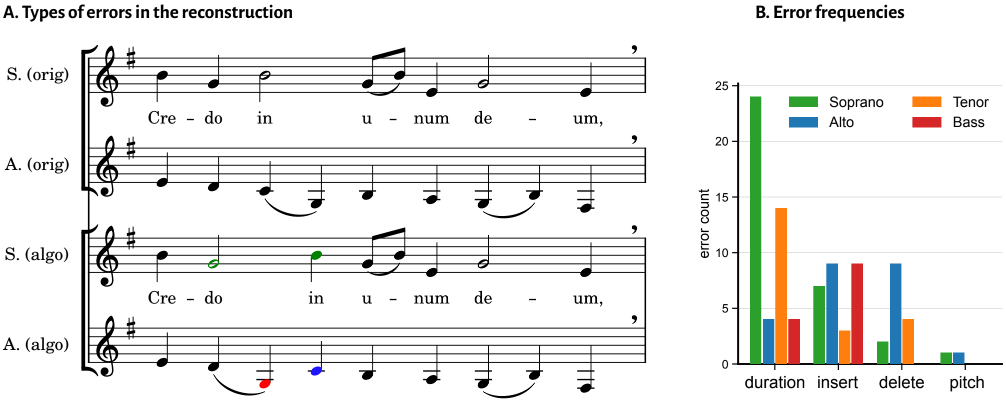


FIGURE 10.9 – Evaluation of the algorithmic reconstruction. This is illustrated in **(A)** by comparing the first bar (bottom staves) with the original (top staves). We encounter four types of reconstruction errors: ornament insertions (red), ornament deletions (blue), duration errors (green) and pitch errors (not shown). In total we find 86 errors (6.7%) after adjusting ornamented exits **(B)**. Over half of these are duration errors, resulting from ornament insertions or deletions. And so only 43 ornaments and two pitches need to be corrected (3.5%) to reproduce the original score.

cussed in the previous section, and determine the note duration. For the latter, we first assign every syllable a duration: 2 if either the alto or the bass has an ornament, and 1 otherwise. Then we evenly distribute the available time over the notes of a voice. I implemented all this in Python using the computational musicology package music21 (Cuthbert & Ariza, 2010). The codebase, named *tintinnabulipy*, provides a convenient interface for plotting and working with T- and M-spaces. It implements all of the tintinnabuli processes described here but is also general enough to be useful for analyses of other compositions by Pärt. Most importantly, it allowed me to generate almost all melodic material of *Summa* in just a few lines of code (see supplement E5).

10.5 Evaluation

How much of the original composition is reproduced by our algorithm? Figure 10.9A compares the first bar of the original score with the algorithmic reconstruction. The reconstruction contains four mismatches–I will call these *errors* for simplicity–in the second and third syllable, which have been colored according to their type. First, we see an *ornament insertion* in the third note of the reconstructed alto part: the reconstruction has an ornament, but the original does not. Conversely, an *ornament deletion* occurs when the original is ornamented, but the reconstruction is not, as with the fourth note in the alto. We also see several *duration errors*: the second note of the soprano for example has double the duration of the original. Finally, *pitch errors* occur when a note has the wrong pitch, but these do not appear in this excerpt. I automated this evaluation to systematically compare the reconstructed score with the original score, part by part and syllable by syllable.[8]

The reconstructed score contains 1288 notes, of which 106 (8%) have one or more errors. Most errors only concern the note duration (60 notes or 56%), but we also find 2 pitch errors, 15 ornament deletions, and 34 insertions, eight of which are in the final two bars. These results show that our algorithmic reconstruction is fairly successful: it correctly reproduces well over 90% of the notes in *Summa*. And this statistic arguably underestimates the performance, since all duration errors are explained by ornament insertions or deletions. If the alto for example misses an ornament, this causes the corresponding soprano note to be too short. And so fixing insertion and deletion errors will automatically resolve all duration errors. That means that only 51 notes (4%) in the reconstruction really need to be corrected in order to reproduce the original score faithfully.

The remaining errors however reveal another plausible regularity. In the reconstruction, one finds several ornaments right before a voice exits to be silent for some measures, whereas ornamented exits are not found in the original score. Removing all ornamented exits resolves six insertions and consequently also reduces the number of duration errors, leaving a total of 86 errors (7%). Of these, 45 (3.5%) are not duration errors and need to be corrected. The alto needs the most correction (19 notes) and is around twice as inaccurate as the soprano, tenor, and bass (10, 7, and 9). This is also summarised in Figure 10.9B. Taking into account that eight errors occur in the final bars, and many other errors remain in the ornamentation, the reconstruction seems very accurate and underscores just how meticulously Pärt constructed *Summa*.

10.6 Discussion and conclusion

Arvo Pärt is known for his unique compositional style, *tintinnabuli*, which has often been described as algorithmic. To assess *how* algorithmic Pärt's tintinnabular music is, this study has attempted to reconstruct one piece, *Summa*, algorithmically. After analyzing and formalizing the piece, I arrived at an implementation that reconstructed most of the original score, showing that at least 93% of the notes in *Summa* can be plausibly explained by an algorithm. Most of the errors, moreover, are faulty note durations caused by insertions or deletions of ornaments in other voices. Correcting these ornamental errors would also resolve the duration errors. This means that only 3,5% of the notes have to be corrected to retrieve the original score, and demonstrates that Arvo Pärt approached the composition of *Summa* extremely systematically.

One might wonder whether the algorithm that I proposed also describes the compositional process: were these the procedures Pärt followed? That may seem plausible, but only the composer can answer that question and Pärt is unlikely to do so.[9] If my analysis is mistaken, the mistakes are probably in the description of the ornamentation, where we found the most errors. However, we should also consider that the composer may have decided to adjust some of the ornaments and that there are no further regularities to be found. After all, multiple corrections of the score have been published. Although I have not been able to compare all editions, some differences in ornamentation can also be heard in recordings.[10] These corrections also leave open the curious possibility that the composer has made 'mistakes' when applying his set of rules. Doing so by hand, rather than by computer, is far from straightforward and would be comparable to a composer from earlier days making an occasional mistake in voice leading.

While analyzing *Summa*, I developed some novel formal machinery. Most notably, I proposed *tintinnabuli processes* to describe how a T-voice can be produced from an M-voice while relying on parts of the melody other than the current melody note. This turned out to be a fruitful generalization of Hillier's tintinnabuli positions. I expect other analyses will also benefit from this concept–as they will from formalization more generally: the intricacies of works like *Summa* are arguably best described in a formal language. This study demonstrates that it can be useful to also implement that formalism, and I hope the resulting codebase will contribute to further formal and computational analyses of Pärt's work.

The methodology this study proposed for that, *analysis by synthesis*, is best suited for understanding algorithmic music: it essentially tries to recover the rules that generated a piece. But it could have wider applicability. Strictly speaking, any piece can be algorithmically reconstructed by simply enumerating all notes in the score. The more rules a piece satisfies, the more concise the description can be. Algorithmic music is an extreme case, but other types of music also follow rules. It may well be possible to for example recover fragments of the middle voices in a Bach chorale from the melody, a figured bass, and voice-leading rules.

That is not to say that algorithmic reconstruction should replace other forms of scholarship. This study has deliberately disregarded all matters of interpretation, which are of course central to understanding the music of Pärt in a broader sense. For that, a methodology like analysis by synthesis seems less useful. But when it comes to understanding how Arvo Pärt's tintinnabular compositions work, this study may provide a fruitful starting point.

Notes

[1] This is based on data released by Bachtrack, a classical music website that tracks many thousands of concerts every year. The website annually releases statistics about concert performances, including the most performed classical composer. In the year 2018 (bachtrack.com/classical-music-statistics-2018), these statistics were based on almost 20.000 concerts, in which Pärt was the top contemporary composers, as he had been since 2011 (see bachtrack.com/classical-music-statistics-2017). In 2019, John Williams came out first, with Pärt second.

[2] I found two conference papers that use small fragments of Pärt's compositions as examples in a live coding setting (Bertram, 2014; Ruthmann et al., 2010). Krämer (2015) cites a script by Christopher Ariza and Michael Scott Cuthbert that generates a score for Pärt's *Pari Intervallo*, which can indeed be found in an old release of music21: github.com/changtailiang/music21/blob/master/music21/composition/phasing.py. David Cope appears to have discussed *Cantus in Memoriam Benjamin Britten* in a course on computer-assisted composition

in 2008. De Paiva Santana and Bresson (2012) presented a poster that modelled *Spiegel im Spiegel* in OpenMusic (see also Shvets & De Paiva Santana, 2014). Outside the academic literature, Guy Birkin in 2015 released the album *Tintinnabuli Mathematica vol. I* with music generated in Mathematica using tintinnabuli rules and number sequences. He explains the process in a blog post available at aestheticcomplexity.wordpress.com/2011/11/11/programming-arvo-part.

3 My translation. It is instructive to read his comment in full:

> Ich habe große Schwierigkeiten, wenn ich meine Werke kommentieren soll.
>
> Ich bin dafür, daß zwischen Wort und Musik ein besonders behutsames Verhältnis sein muß. Wir müssen der Musik eine Chance geben, sich allein auszudrücken. Wörter treiben die Musik in die Enge. Und auch die Musik neigt dazu, sich von Wörtern abhängig zu machen. Ich sehe in dieser »überkommunizierten« Gesellschaft Gefahr für die Existenz der Musik.
>
> Ich muß in mir Raum frei lassen für Musik, und wenn dieser Raum mit Worten besetzt wird, bleibt mir kein Bedürfnis, mich mit Musik auszudrücken — und umgekehrt: wenn ich ein Musikwerk geschrieben habe, bleibt nichts mehr mit Worten zu sagen übrig.
>
> ★ ★ ★
>
> Ich habe ein hochformalisiertes Kompositionssystem entwickelt, in dem ich seit 20 Jahren meine Musik schreibe. In dieser Reihe ist *Summa* das strengstgebaute und verschlüsselste Werk. Die Verschlüsselungen finden sich in vielen Schichten der Partitur.
>
> (Berlin, den 15.6.1994)

4 This analysis is based on the 1980 version for violin and piano.

5 Shenton (2012) also cites the masters thesis by Kosak (1994), which I have however not been able to find.

6 UE 33 686, Korr. III/2012, to be precise.

7 As the piece is diatonic we only have to represent pitches in the E natural minor scale. That means we let E_2 correspond to 0, $F_2^{\#}$ to 1, G_2 to 2, and so on.

8 To obtain a digital version of the original score, I transcribed my physical copy in MuseScore. I manually compared all errors identified in reconstruction against the physical score, and this allowed me to resolve some transcription mistakes.

9 In the comments that are reproduced in footnote 3, Pärt expresses his “great difficulty” in commenting on his own works as he wants to give the music a chance to express itself.

10 To give just one example, in measure 12 the score used in this study ornaments ’ve-’ in ’verum’ with an E_4 in the alto, while the recording by the Hilliard Ensemble (1987) sings a G_4, as does the more recent recording by Vox Clamantis (2016).

SUPPLEMENTARY MATERIALS **E1** Textual structure of *Summa* · **E2** Approximate patterns · **E3** Tenor and soprano ornaments · **E4** Alto and bass ornaments · **E5** Implementation: code sample · **E6** Ending of *Summa*

CODE All code will be made publicly available via github.com/bacor/algo-part,

REFERENCE This chapter has not yet been published.

AUTHOR CONTRIBUTIONS BC designed the study, conducted the research, and wrote the chapter.

ACKNOWLEDGEMENTS I would like to thank Dorien Salet for introducing me to *Summa* and the members of ensemble Bes Klein for giving me a reason to analyze the piece in detail. This would not have happened if Euwe de Jong had not first introduced me to the tintinnabuli style. Finally, I would like to thank Ashley Burgoyne and Jelle Zuidema for their comments on the manuscript, and Iris Bouman for helping me with some German translations.

CHAPTER

11

Postlude

Postlude

Music exemplifies the richness and diversity of human life, I wrote in the opening of this dissertation. In the chapters that followed, I have looked for ways to formally understand and measure some of that richness. By slicing chants into natural units, I tried to capture melodic modes in plainchant. Like the sounds that carry music, the shapes of melodies turn out to be best described as a combination of waves, in this case, cosine functions. But those shapes, when it concerns phrases, can vary almost smoothly and do come in clear categories. This search for categories, or statistical modes, became a recurrent theme. I looked for modes in contours, in plainchant, but also in the rhythm of Malian jembe music or the vocalizations of lemurs. And just as small motifs reveal regularities in those rhythms, small motifs reveal regularities in the movement of the melodies. But rarely is that movement as regular, are the musical shapes as clear, as in the hands of Arvo Pärt, whose music is a treasure trove of formal musical structures.

11.1 Contributions

Let me unpack all this and list the contributions in this dissertation.

CORPORA AND SOFTWARE The first line of research laid the technical groundwork: corpora, parsers, and other software. I released two chant corpora, **Cantus Corpus** and **GregoBase Corpus**. Both repackage existing corpora in a way that makes them more suitable for computational research. With the same goal in mind, I developed the Python library **chant21** that improves support for two chant formats in music21: Volpiano and gabc. I proposed parsing expression grammars for both gabc and Volpiano, and used these to build a hierarchical representation of the chant, segmenting it into neumes, syllables, words, and sections. Besides chant corpora, I proposed a way to index folk music corpora in a proof-of-concept project named **Catafolk**.

CHANT AND MODES The second line of research investigates plainchant and its modes in particular. I proposed a **distributional approach to mode classification** in Medieval plainchant using tf–idf vectors, which outperformed two other approaches. The distributional approach still performed reasonably well using a contour representation that was stripped of almost all pitch information, demonstrating that **mode is more than scale**. Crucially, this worked only when segmenting the chant in its **natural units**: groups of notes corresponding to syllables or words in the text. This is consistent with the idea that chant is composed by centonization, a process in which existing chunks of music are recombined to form new chants.

To better understand the classifier, I introduced a simple attribution method, **witness coloring**, that highlights which motifs contribute to the classification. For antiphons, this method consistently highlights differentiae, the psalm endings sung before repeating the antiphon that frames it. Using an entropy measure, I furthermore showed that **differentiae-antiphon connections** are more predictable in some modes than others and that differentiae are more predictable than antiphons. Finally, I trained a recurrent neural language model on plainchant, capable of generating chant. The **neural chant model** learns rich musical representations. It for example appears to represent pitch information without being explicitly trained to do so, and even when trained on interval representations. I also suggest that the statistical modes in the learned chant space may correspond to mode-genre combinations.

MELODIC CONTOUR The third line of research focused on the analysis of melodic contour. In a first case study, I confirmed the **melodic arch hypothesis** in plainchant, by comparing phrases to a novel baseline of **random melodic segments**. This eventually led to the observation that **principal components of melodies approximate cosines**. Explaining this by a particular covariance structure observed in this data, motivated **cosine contours**, a novel contour representation that uses the discrete cosine transform. Turning to the typology of melodic contour, the UMAP-dip test could discriminate clustered from unclustered synthetic contours but failed to find any evidence for clustering in actual phrases. In other words, **phrase contours do not appear to cluster**. Further identifying a **hidden tolerance parameter** in Huron's typology, lead me to argue for a continuous view of contour.

MOTIFS A fourth line of research concerned musical motifs: small fragments of melodies or rhythms. When classifying modes in plainchant, using the right segmentation of the chant turned out to be key, and suggested that the **units of plainchant** are motifs based on the text. Fixed-length motifs nevertheless prove useful, for example when visualizing rhythm. After identifying some problems in raster plots, I propose ways to **visualize inter-onset interval data** using rhythm triangles. I use these to reanalyze rhythms in music and in vocalizations of a range of species. Thinking in terms of motifs also led to a measure of **isochrony** that extends the nPVI. Inspired by rhythmic motifs, I visualized the occurrence of short melodic motifs in **melody squares**, which motivated several cross-cultural generalizations.

ALGORITHMIC MUSIC Finally, I presented a **algorithmic reconstruction** of Arvo Pärt's *Summa* that almost completely reproduces the original score. It is a case study in analysis-by-synthesis, that resulted in new formal machinery, **tintinnabuli processes**, and software, that can help to formally understand the music of Pärt. Although this music may deserve the label *algorithmic*, it is of a very different nature than the **plainchant generated algorithmically** using the neural chant model.

11.2 Discussion and future directions

Returning to the very first chapter, I motivated this dissertation against an evolutionary backdrop: why did humans evolve to become musical animals? What abilities allow humans to produce and perceive music, and what is their evolutionary history? To pinpoint those abilities, we need to understand what forms musics can take. Analogous to a multi-component perspective on musicality, this requires a typological perspective on musics. When it concerns music in a narrow sense—formal aspects of musical behavior, such as the information contained in musical scores—it makes sense to approach typology computationally: to *measure musics*.

Compared to the breadth of that original agenda, this dissertation takes only some very small steps, as it only studied a few musical phenomena (modes, contours, and motifs). The cross-cultural generality of the studies is moreover rather limited: I have primarily analyzed folk music from the Essen Folksong collection, plainchant (Cantus and GregoBase), and the Densmore collection. As I have noted in chapter 2, the ideal would be a representative, global sample of music corpora. Collecting such a corpus, if at all possible, would require collaboration at a much larger scale, in which music researchers, as a community, would have to bring together their resources—precisely what motivated the Catafolk project. Nevertheless, this dissertation already suggests several interesting directions for future work and I want to highlight a few of those.

This dissertation studied modality in Western plainchant only, and so perhaps the most promising future work would be to extend this work to other traditions. As mentioned in chapter 4, other studies have also characterized modes motif-based models were also used to classify raga or makam. This convergence of models in different musics, could be seen as some sort of computational resolution to the problem of incommensurability. If the modes in plainchant, raga, and makam can be described using the same computational model, the concepts become formally comparable, even if their musical meanings are not comparable. This deserves further investigation.

A closely related question that deserves further study concerns the relationship between statistical modes in melody space and musical modes. Chant representations produced by a recurrent neural network (chapter 5) appeared to suggest that statistical modes in plainchant may correspond to mode-genre pairs. This raises the question of whether modalities in other traditions, such as ragas or makams, correspond to modes in some melody space. If so, it could suggest an explanation for why multiple musical traditions organize their repertoires around modes (Powers et al., 2001). Perhaps a large enough repertoire will tend to be organized in classes, corresponding to statistical modes in the melody space. It is conceivable that this is a by-product of learning melodic expectations statistically. One might also wonder if there are common tendencies in how those modes will be theoretically characterized, such as along the lines of scales and contours.

Another promising line of research that this dissertation unfortunately not touched upon, is the historical development of plainchant. With manuscripts spanning several centuries, the Cantus database is a very promising test bed for studying models of cultural evolution. What is particularly promising is the variety of 'phylogenetic signals' that one can find in the corpus. From the perspective of this dissertation, one might think of deriving such a signal from the melodies themselves, by measuring their melodic similarity. But even just the contents of the manuscripts already provide an informative signal: which feasts do they contain, and what chants are used for which particular feast? This can be indicative for chant traditions, without even requiring any melodic transcriptions.

Besides mode, contour was the second protagonist of this dissertation. It seems likely that studies of intonation contours in language may profit from our approach to melodic contour. Concretely, the *melody development model*, suggests that the vocalizations of infants (not only their cries) also gradually increase in complexity during the first six months of development (as evidenced by an increasing number of arches in the f_0 contour) and that this is an important step toward acquiring a language. In a vast dataset of almost 70,000 vocalizations, Wermke et al. (2021) find evidence for a gradual increase in complexity. The study relies on a classification of vocalization contours into either a simple or complex category. Using cosine contours seems like a promising alternative: as it describes a contour as a combination of waves, vocalization complexity should quite naturally translate into more energy in the higher-frequency components.

In that same spirit, the methods I developed to study the clusterability of melodic contour, seem promising for studying prosody. One interesting case concerns the ToBI system that uses a used to describe English intonation contours (Silverman et al., 1992). Do the types of intonation contours indeed correspond to clusters of contours? This is the exact same question I asked about musical phrase contours and may be addressed using the dist-dip test.

Motifs were the third antagonist, be it in rather different ways. Our work on mode classification highlighted the need to use variable-length motifs, while fixed-length motifs proved informative in a more explorative visualizations. In the case of rhythmic motifs, an interesting extension would visualize the rhythms in vocalizations of various bird species using rhythm triangles. An obvious starting point would be Xeno-canto, a vast collection of recordings of bird sounds from around the world. It is also where Roeske et al. (2020) got their nightingale recordings that were visualized in chapter 7. A challenge, and in itself an important problem, would be automatic onset detection, for which Roeske et al. (2020) may offer a starting point.

11.3 Reflections

I would like to end on a personal note. Like most dissertations, this one did not come together easily and also did not follow the path the first research proposal laid out. This dissertation was originally intended to be about the cultural evolution of language, not about music. Accordingly, I spent the first year of my Ph.D. drafting a syllabus on the evolution of language and music: a fascinating but perhaps too ambitious project. Nine months and several chapters later my motivation had evaporated, and I decided to quit.

In hindsight, my master's, combined with too many other activities must have left me nearly overworked before I even started my Ph.D. Choirs offered a welcome escape, singing lessons soon became a form of therapy, and by the end of the first year, I had taken up a study in classical voice at Utrecht Conservatory. When I told my main supervisor, Jelle, that I had decided to exchange the university for the conservatory, he suggested to also exchange language, my original research area, for music and *combine* singing and science instead. This worked out surprisingly well. The conservatory gave me energy, direction, and inspiration and formed a productive counterweight to the intellectual work at university. When a pandemic forced academics to work from home, conservatories would soon open their doors again and offered another welcome escape.

But having 'lost' a year on a syllabus that never materialized and being occupied by a conservatory study, the pressure to produce some sort of academic output made me opportunistic. Not driven by a deeply felt fascination and limited by practical constraints, I jumped on whatever came along. Modes? Sure. Contours? Why not. Given the circumstances, this was fine, but it fed my insecurities: my academic work felt rushed, unguided, and shallow—even though others evidently disagreed.

Perhaps this is why I could not forge a compelling overarching argument out of my research when the time came to write things up. I instead decided to focus on the interludes. Taking more liberty than scientific articles allow for, I enjoyed writing them, and finished a first version of this dissertation in September 2022. The manuscript was somewhat unpolished and unconventional, but finished. Only some five months later, did I find the energy to start with the final corrections, to find that I now agreed with most of my supervisors' earlier reservations. I have tried to polish the dissertation, but I am well aware of the many imperfections that remain. So be it—*hora est*.

11.4 Acknowlegdements

Let me end by thanking you, dear reader, for reading this. You would, however, not be reading this, had it not been for the many people and institutions that have supported me throughout my life, and for which I am extremely grateful. I want to use this opportunity to thank some of them, starting with the Institute for Logic, Language and Computation. Thanks for trusting me with the rare Ph.D. position that gave me almost complete freedom to pursue whatever research I considered worthwhile while also allowing me to take up singing. I have done what I could to make the best of the privilege and opportunity.

Next, I want to thank my supervisors. Jelle, thanks for supervising me for more than six years and for working together on so many different topics, from cognitive modeling and drawing games to language evolution and, indeed, all the topics in this dissertation. I am very grateful for finding a supervisor with such broad interests and an open mind. In particular, I don't know how to thank you enough for supporting my music studies, for tolerating the chaos and the ever-shifting priorities that followed, and for bearing with me. All this applies equally to Ashley, who I moreover want to thank for guiding me in the field of computational musicology. Henkjan, although you only formally became a supervisor at the very last moment, you have been one since I first studied with you in 2015. Thank you for your encouragement and enthusiasm, which has proven extremely contagious.

Many thanks to all my colleagues at ILLC and the members of the CLC lab in particular: Dieuwke, Jaap, Lisa, Oskar, Samira, and Tom. From the MCG, Atser, Bastiaan, Berit, Carlos, David, Fleur, Jiaxin, Makiko, Xuan, and all those who joined our reading groups. Those that made the ILLC run: Jenny, Peter, Tanja, Roos, Alex. My office mates: Bryan, Jasmijn, Sophie, and, at the very end, Marianna, also for your advice on, well, this. Finally, Mariann*e*, I regret not working together more: I somehow always considered you a kind of academic soulmate.

Without singing, I doubt I would have finished this dissertation. And so, thanks, Eva, for pushing me to start singing in the VU-kamerkoor, where music could again infiltrate my life. I am grateful the Conservatory of Utrecht has allowed me to combine my studies with my Ph.D., which I largely owe to Karin: thanks for your guidance and support. Euwe, thanks for encouraging me and introducing me to plainchant and Pärt. But I am especially indebted to all my fellow students, but Lisa, Sara, and Joana in particular, for the fun, support, and warmth. Back at Science Park, I want to thank Saskia for her advice on genes and music and the other members of Bes Klein for the fun (and *Summa*): Lieke, Dorien, Gerben, Janusz, Vincent, and Anna (who I also want to thank for last-minute corrections to my rusty Dutch).

It has been well over a decade since I first moved to Amsterdam for my bachelor's in Bèta-gamma, which surrounded me with the nicest and brightest people, many of whom became dear friends. Thanks to all those who made me feel at home in Amsterdam and/or kept an academic fire burning: Thomas, Marie Beth, Roland, Jesse, Sjang, Tania, Gerrit, Danna, Micha, Thor.

Finally, Johan and Anna, thanks for your never-ending support and trust in whatever I end up doing: in all those moments that mattered most, you have been there for me. Boris, I consider myself very lucky to have had you as an example. Thank you, also for bearing with me while I type these very words. The last person I want to thank is you, Iris, for who you are, for your love, for your support, and for tolerating me—and this dissertation.

Supplements

SUPPLEMENT A

Supplements for chapter 4: Modes

A1 Data and code

All data and code used in this study have been made available online (see the end of page 44). All randomness in the code has been fixed, so it should in theory be possible to reproduce our results exactly. The evaluation metrics of all experiments are already included in the repository, as is the data used in the first run of the experiment; this should be sufficient for reproducing most figures. We have included model predictions and tuning results only for the first run of the experiment. Detailed logs of everything from data generation to visualization can also be in the repository, together with many more figures besides those included in the paper and the supplements. In particular, the repository contains heatmaps with multiple evaluation metrics (accuracy, precision, recall, and F_1) for all models and all experimental conditions.

A2 Filtering

As described in the main text, we filtered the total dataset of 497,071 chants to obtain a clean subset of responsories and antiphons. The effects of all of the filters are logged and will be made available online. As an example, below we show the output of the series of filters applied to obtain the full set of antiphons used in this study.

```
Exclude all chants with an empty volpiano field
 > 87.20% removed (433443 out of 497071; 63628 remain)
Exclude all chants without notes
 > 2.87% removed (1825 out of 63628; 61803 remain)
Include only chants with simple modes: 1-8, not transposed
 > 23.02% removed (14227 out of 61803; 47576 remain)
 > 20.65% removed (9823 out of 47576; 37753 remain)
Filter chants whose incipit is identical to the full text
 > 14.59% removed (5507 out of 37753; 32246 remain)
Include only chants with a certain genre (here: antiphons)
 > 52.06% removed (16787 out of 32246; 15459 remain)
Exclude chants that do not start with a G clef
 > 0.05% removed (7 out of 15459; 15452 remain)
Exclude chants that contain an F clef
 > 0.00% removed (0 out of 15452; 15452 remain)
Filter chants with missing pitches: containing the substring 6------6
 > 7.54% removed (1165 out of 15452; 14287 remain)
Exclude all chants with non-volpiano characters
 > 0.03% removed (5 out of 14287; 14282 remain)
Only include chants with '---' in their volpiano
 > 0.08% removed (11 out of 14282; 14271 remain)
Filter duplicate chants: whose notes occur multiple times
 > 2.84% removed (406 out of 14271; 13865 remain)
```

A3 Dataset statistics

The number of chants, their average length, and the number of notes for each dataset. We sort datasets by genre, then by subset (include melody variants in the full set, or exclude them in the subset), and finally by train/test split (or total for the two combined). The train/test splits are different in each run of the experiment. These statistics are computed from the data used in the first run, and others are comparable.

Genre	Subset	Split	# chants	# notes	Mean length (notes)
responsory	full	train	4 922	676 807	137.5
responsory	full	test	2 109	290 064	137.5
responsory	full	total	7 031	966 871	137.5
responsory	subset	train	1 234	169 642	137.5
responsory	subset	test	529	72 504	137.1
responsory	subset	total	1 763	242 146	137.3
antiphon	full	train	9 706	576 738	59.4
antiphon	full	test	4 159	248 405	59.7
antiphon	full	total	13 865	825 143	59.5
antiphon	subset	train	2 911	190 165	65.3
antiphon	subset	test	1 248	82 781	66.3
antiphon	subset	total	4 159	272 946	65.6

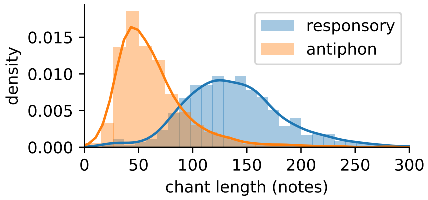
A4 Majority baselines

Below we show the frequency of the largest classes in each of the datasets. Boldfaced values correspond to the classification *accuracy* of the worst-performing conditions discussed in the main text. (The frequencies are marginally different in the five experimental runs; shown are the averages.)

genre	dataset	kind	top mode	frequency
responsory	full	train	8	20.85%
responsory	full	test	8	**21.13**%
responsory	subset	train	1	21.65%
responsory	subset	test	1	20.19%
antiphon	full	train	8	28.47%
antiphon	full	test	8	**28.13**%
antiphon	subset	train	1	23.50%
antiphon	subset	test	1	24.18%

A5 Chant lengths in two genres

Responsories are usually much longer than antiphons. The distribution is estimated from the training datasets without melody variants:



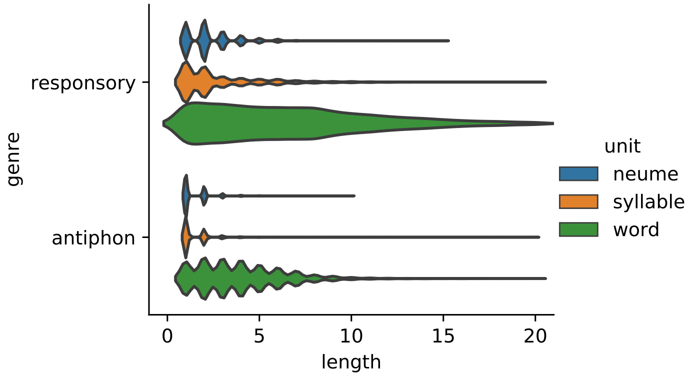
A6 Mean lengths of natural units

Natural units have different lengths in responsories and antiphons, as the mean lengths (in the number of notes) show. section A7 shows the full distribution. Means are estimated from the training datasets without melody variants.

	neume	syllable	word
antiphon	1.50	1.55	3.98
responsory	2.32	2.96	7.12

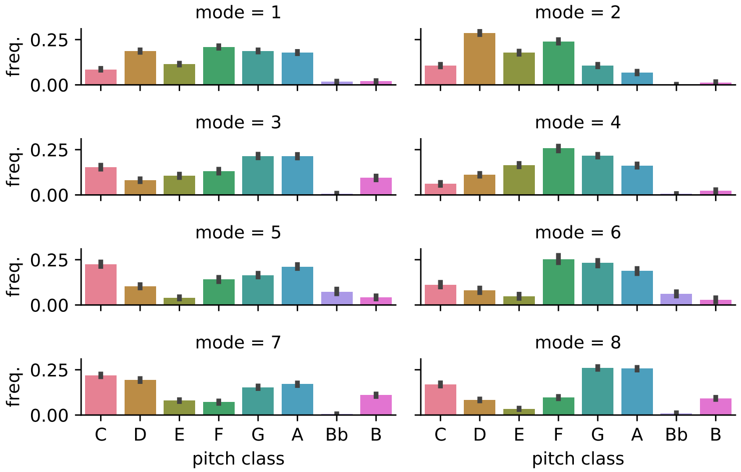
A7 Lengths of natural units

Natural units have different lenghts in responsories and antiphons. Responsories are more *melismatic*: they use more notes per syllable. As a result, a typical word is also much longer. This is shown in the figure using violin plots, a visualization of the length distribution using a kernel density estimate. Note that the total area has no meaning in this plot; we normalized the widths of the violins for better readability. The distributions are estimated from the training datasets without melody variants).



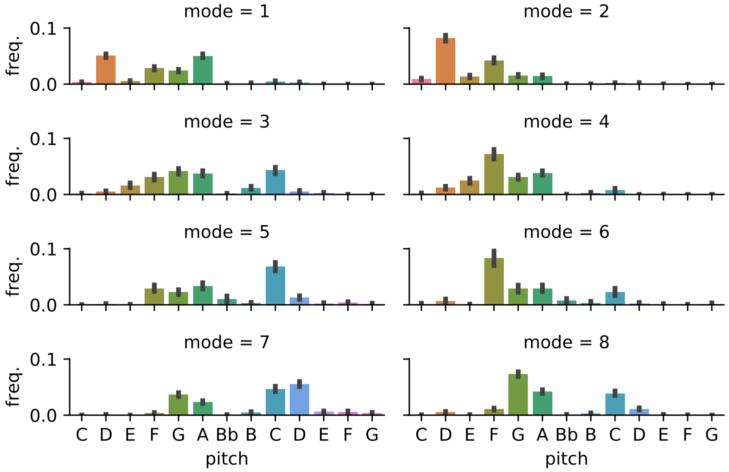
A8 Pitch class profiles

The pitch class profiles used in the profile approach. Shown are data for responsories, estimated from the training data without melody variants.



A9 Repetition profiles

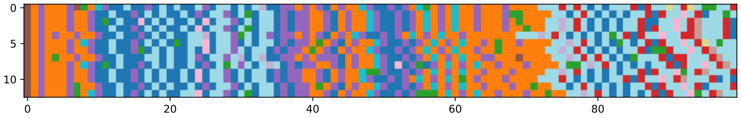
The repetition profiles that are used in the profile approach. Every bar shows the average number of repetitions of that note in a chant (see main text for details). Shown are data for responsories, estimated from the training data without melody variants.



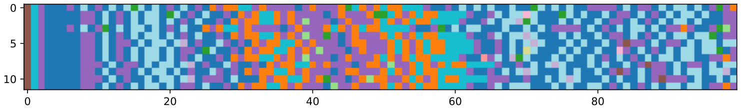
A10 Melody variants in Cantus.

The top two panels show examples of sets of melody variants: the first 100 notes of melodies sharing a Cantus ID. Different colors correspond to different pitches, or more precisely, different Volpiano characters after discarding dashes. As a comparison, the bottom panel shows 100 notes of 20 random melodies.

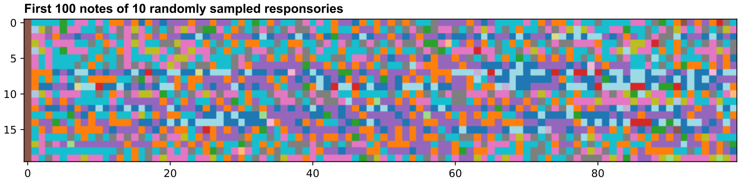
First 100 notes of the 13 chants with Cantus ID 007170



First 100 notes of the 12 chants with Cantus ID 007454

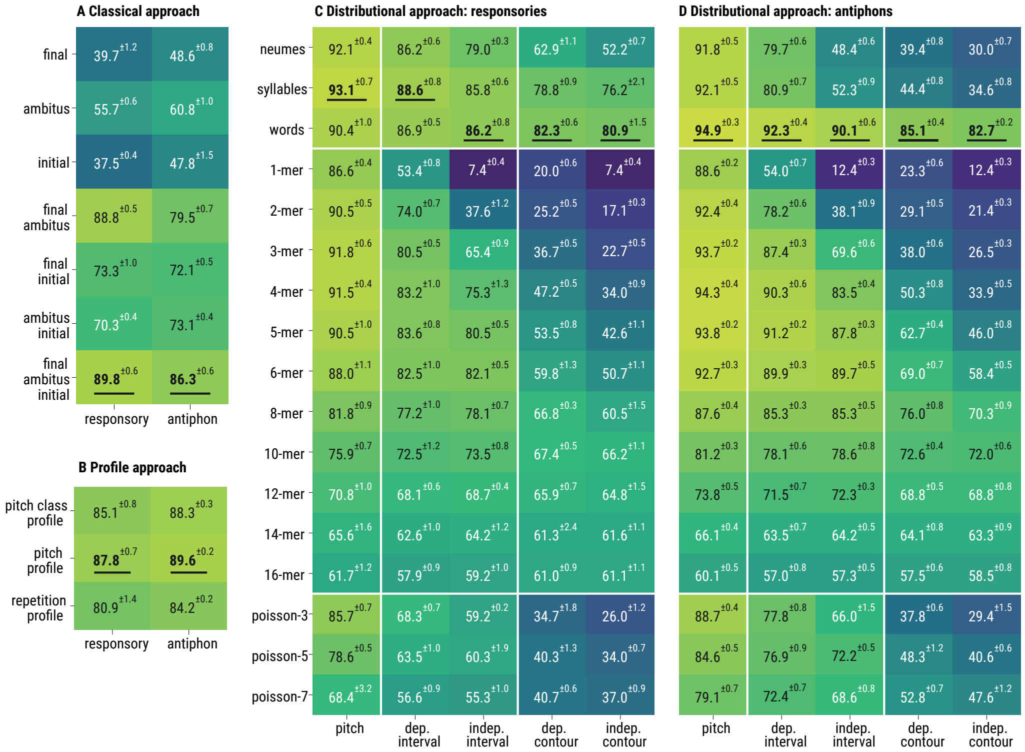


First 100 notes of 10 randomly sampled responsories



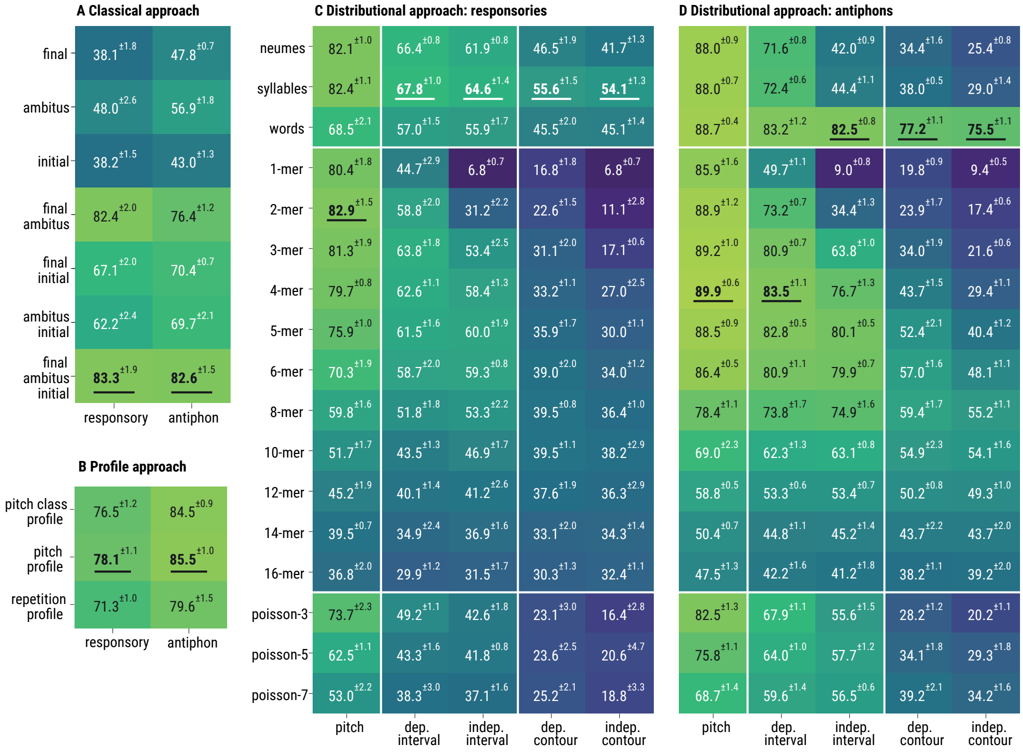
A11 Results with standard deviation

This is essentially the same figure as Figure 4.5 but now with the mean F_1-score μ and its standard deviation σ shown as $\mu^{\pm\sigma}$, computed from five independent runs of the experiment.



A12 Main results on subset

Cantus often contains several variants of the same melody, as shown in supplement A10. As discussed in the main text, this is a difficult issue that for example also applies to the Essen folk-song collection. We decided to repeat our experiments on a subset of the data where we excluded melody variants. We heuristically identified melody variants by randomly picking one chant from all sets of chants that have the same Cantus ID and mode. This resulted in a set of 1763 responsories and 4159 antiphons. In terms of the number of notes, this meant a 75% and 66% reduction in data size for responsories and antiphons respectively. This figure shows the main classification results on this subset of the data. The performance of all models decreases on this subset, and for responsories more than for antiphons. The drop is greatest for responsories across models. The main result that only natural units maintain high performance, even on contour representations, nevertheless stand. Our main findings that contours are sufficient and that natural units work best across representations stand. We do observe some reorderings: some already high-performing *n*-grams in antiphons now for example slightly overtake word segmentations, although only for pitch and dependent interval representations. The distributional approach works best for antiphons regardless of including or excluding chant variants, but for responsories, the distributional approach drops slightly below the classical approach on the subset (where the profile approach is worst). These findings might be explained by increased sparsity in the smaller dataset: natural units in responsories are, after all, longer.



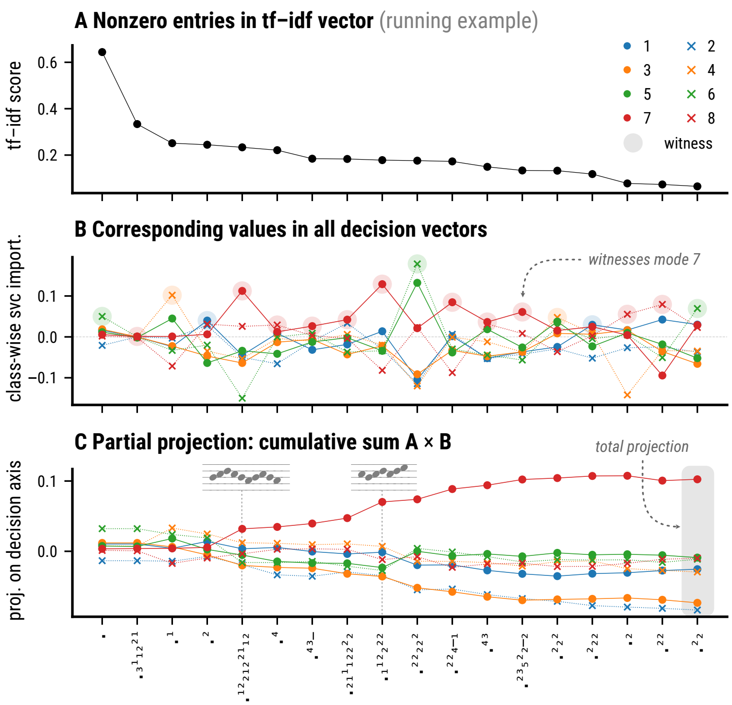


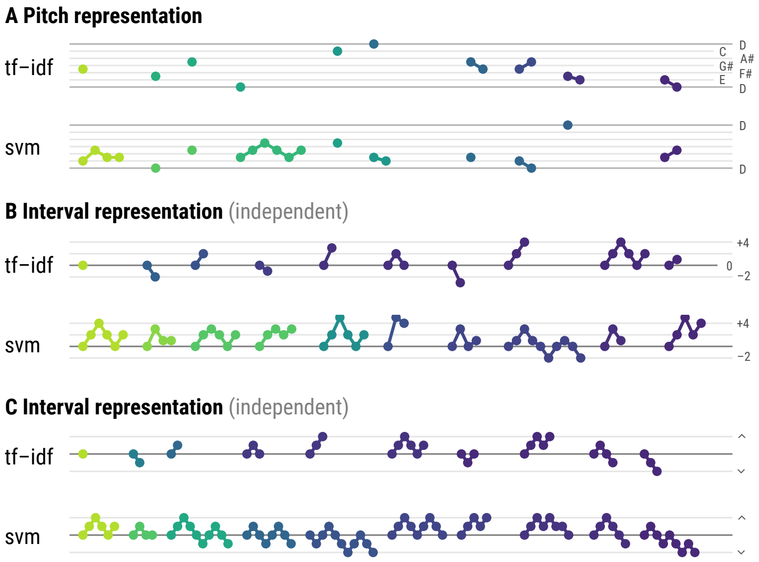
FIGURE A.1 – **Projection on decision axes. (A)** shows the sorted, nonzero entries of the the tf–idf vector, and **(B)** the corresponding entries of the decision vector. The partial projection is shown in **(C)**. It highlights how each feature contributes to the final projections. The solid red line for mode 7 for example increases markedly at the fifth and ninth motifs (shown with musical scores). Those motifs partly explain why the chant is classified to mode 7. The same chant as in Figure 4.7 is shown.

A13 Feature importance

Here we discuss the attribution method discussed in section 4.4 in more detail. We restrict the discussion to the syllable segmentation in responsories. Besides the *SVM-feature importance* discussed in the main text, we also discuss using raw *tf–idf* scores as a measure of feature importance. For each of these two measures, we distinguish two variants: a *class-specific* and a *general* one. The class-specific variant measures how important a feature is for determining whether a chant belongs to *one specific* mode. The general variant measures the importance for classifying to any class.

CLASS-SPECIFIC SVM-IMPORTANCE This is the measure discussed in the main text, but to understand it better, it is instructive to go through the projection of a tf–idf chant vector on a decision axis in more detail. Let $\mathbf{u}^{(m)} = (u_1, \dots, u_n)$ be the decision vector of mode m, orthogonal to the decision boundary, and let $\mathbf{x} = (x_1, \dots, x_n)$ be the tf–idf feature vector of some chant. If $\mathbf{u}$ is normalized, the projection is given by the inner product $\mathbf{u}^T\mathbf{x} = u_1x_1 + u_2x_2 + \cdots + u_nx_n$. In practice, there are only a few terms in this sum as terms for which x_i is zero do not contribute to the total. In a sparse tf–idf vector, there will be many such terms. In Figure A.1 we visually illustrate this to highlight which motifs contribute to the projection—and the eventual classification. Concretely, if the k-the entry $\mathbf{u}_k^{(m)}$ of decision axis $\mathbf{u}^{(m)}$ has a large value, the k-th motif contributes to classifying a chant to mode m. The class-wise SVM-importance of motif k for mode m is:

$$\text{SVM-importance}_m(k) = \mathbf{u}_k^{(m)} \tag{A.1}$$



A Pitch representation

B Interval representation (independent)

C Interval representation (independent)

FIGURE A.2 – Top most important motifs according to different general importance measures. Single notes stand out in the pitch representation **(A)**, suggesting mostly scalar information is encoded, whereas the interval and contour representation **(B, C)** emphasise larger motifs, encoding more melodic information. Tf-idf importance seems to rank common, short motifs higher, whereas SVM importance appear to favour motifs that discriminate modes. Lighter colours indicate more important features.

GENERAL SVM-IMPORTANCE Importance for any particular class indicates general importance, and so the general measure is essentially an average of importance scores. However, it takes into account counter-evidence. If coordinate k of a decision vector is strongly negative, an occurrence of motif k can be seen as strong counter-evidence for mode m. The class-specific measure does not consider k to be important *for mode m*, but the motif should have general importance. To ensure that both strong evidence and strong counter-evidence indicate importance, the general variant takes the mean of the *absolute values*:

$$\text{SVM-importance}(k) = \text{average}(|\mathbf{u}_k^{(1)}|, \dots, |\mathbf{u}_k^{(8)}|). \tag{A.2}$$

CLASS-SPECIFIC TF–IDF IMPORTANCE Next, we experimented with using tf–idf scores as measures of feature importance. To that end we simply computed average scores across all chants of a particular mode:

$$\text{tf–idf-importance}_m(k) = \text{average}(\mathbf{x}_k : \text{chant } \mathbf{x} \text{ has mode } m) \tag{A.3}$$

GENERAL TF–IDF IMPORTANCE The general variant is identical except that it now averages over all chants:

$$\text{tf–idf-importance}(k) = \text{average}(\mathbf{x}_k : \mathbf{x} \text{ is any chant}) \tag{A.4}$$

COMPARISON OF GENERAL IMPORTANCE MEASURES First, Figure A.2 shows the top-ranking items according to the general tf–idf and SVM importance measures. It clearly stands out that the tf–idf scores emphasize short, relatively common motifs across modes, whereas SVM importance emphasizes larger motifs. This is also why witness coloring using the SVM scores seems more informative than using tf–idf scores. Despite the differences in the top-ranking motifs, there is some correlation between the two importance measures. This can be seen in Figure A.3 which plots the SVM-rankings against the tf-idf rankings.

Finally, we assessed how classification performance is impacted by only using the top-n features according to each of the importance measures. Figure A.4 shows that pruning all except the top n features results in much higher performance than a baseline that uses n randomly chosen features. The two importance measures behave similarly in this pruning experiment, but for contour and interval representations, the SVM importance seems to produce better rankings than the tf–idf importance. The figure also suggests that the classifier relies much more on the top features, than it does for interval and contour representations, as the performance increases very rapidly for the pitch representation when increasing n.

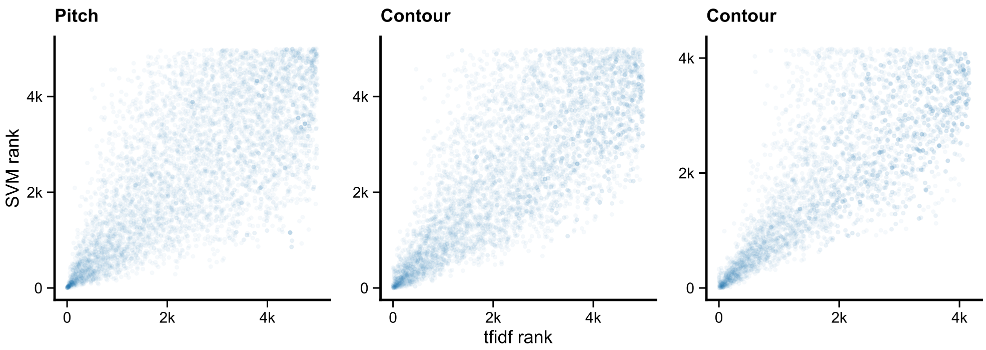


FIGURE A.3 – **Rank correlation of general feature importance measures.** We compare tf–idf and SVM feature importance by looking at their rank correlation in three representations. There seems to be some consistency in how these measures rank different motifs.

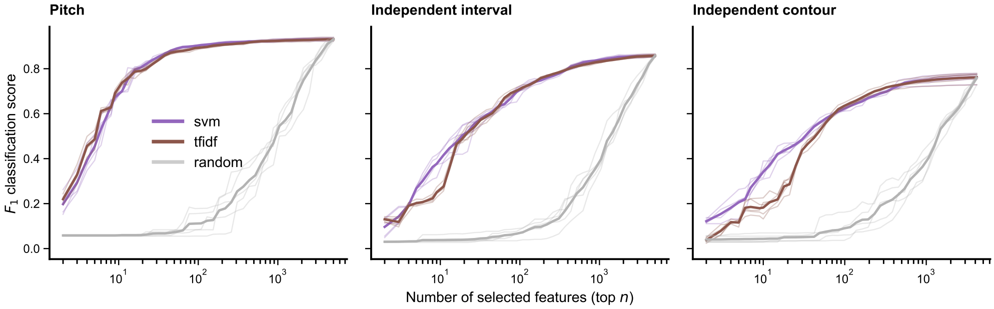
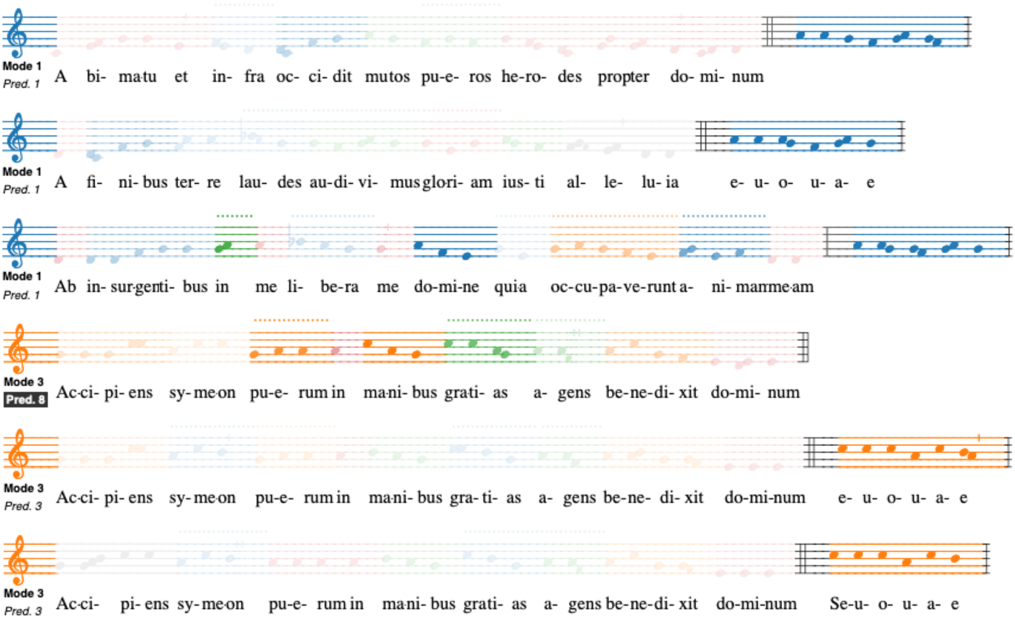


FIGURE A.4 – **Pruned model performance.** We compare measures of feature importance by evaluating the classifier when eliminating all but the top-n features according to an importance measure. Pruning the model based on the SVM importance measures seems to impact performance less than using tf-id for the contour representation. Although the differences are small in terms of model performance, the top-ranking motifs are noticeably different. Data are for responsories using a syllable segmentation, thick lines are averages over the five runs.

A14 Witness coloring antiphons

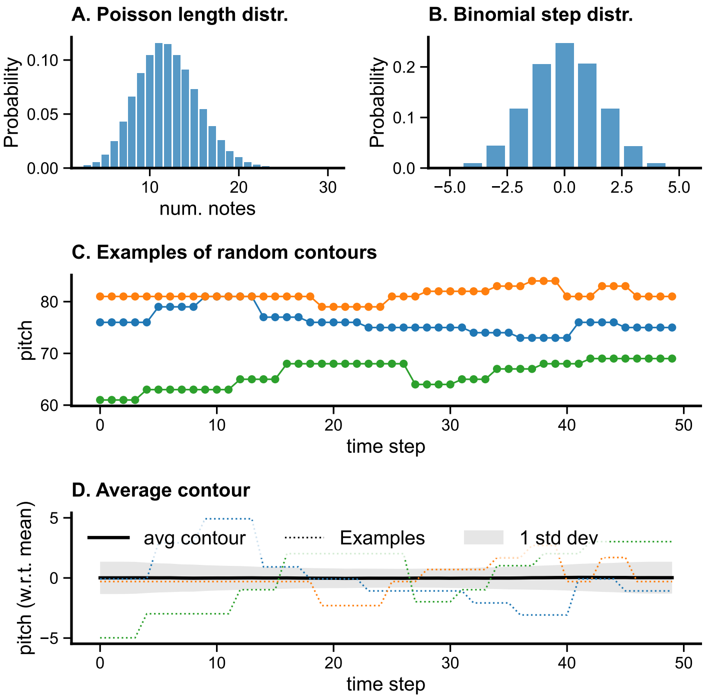
We show six antiphons, three from mode 1 and three from mode 3. The coloring shows which motifs (in interval representation) contribute to the classification. The figure illustrates that the final sections, called differentiae, play an important role. Not only are differentiae indicative of mode, but these motifs are also longer than syllables in antiphons.


Mode 1
Pred. 1
A bi- matu et in- fra oc- ci- dit mutos pu-e- ros he-ro- des propter do- mi- num
Mode 1
Pred. 1
A fi- ni- bus ter- re lau- des au-di- vi- mus glori- am ius- ti al- le- lu- ia e- u- o- u- a- e
Mode 1
Pred. 1
Ab in- sur-genti- bus in me li- be-ra me do-mi-ne quia oc-cu-pa-ve-runt a- ni- mam me-am
Mode 3
Pred. 8
Ac-ci- pi- ens sy- me-on pu-e- rum in ma-ni- bus grati- as a- gens be-ne-di- xit do-mi- num
Mode 3
Pred. 3
Ac-ci- pi- ens sy- me-on pu-e- rum in ma-ni- bus gra- ti- as a- gens be-ne- di- xit do-mi-num e- u- o- u- a- e
Mode 3
Pred. 3
Ac-ci- pi- ens sy- me-on pu-e- rum in ma-ni- bus grati- as a- gens be-ne-di- xit do-mi-num Se-u- o- u- a- e

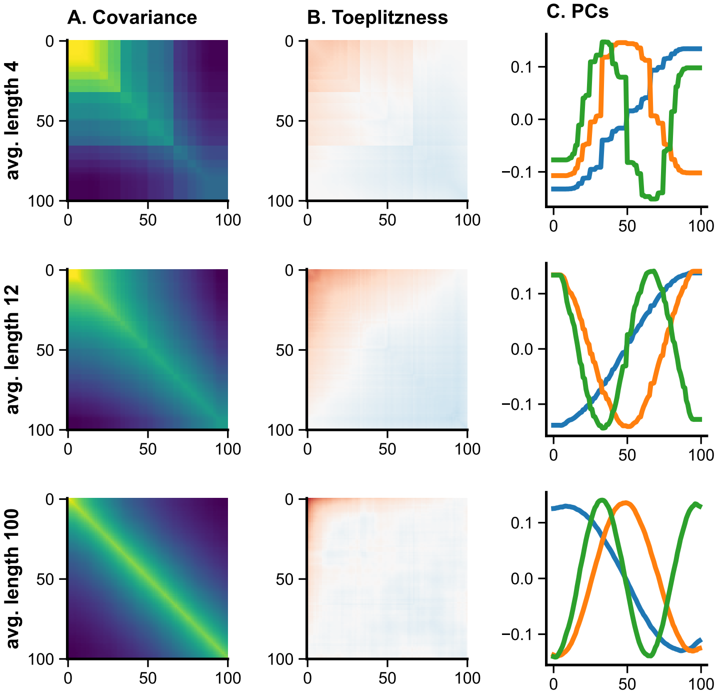
SUPPLEMENT B

Supplements for chapter 6: Cosine Contours

B1 Random walk baseline



We compared the principal components of phrases to a random walk baseline that was intended to be fairly similar to actual phrase contours. First, we draw the length (number of notes) K of the random walk from a Poisson distribution with mean $\lambda = 12$ (truncated below 3). The value 12 was chosen so as to approximate the length distribution of phrases. Then we draw an initial pitch x_0 uniformly between 60 and 85 (in MIDI pitch space). Next, at every step k we draw the size of a step r_k (the interval) from a Binomial distribution with parameters $n = 10$ and $p = 0.5$, shifted to have mean 0, and let the next pitch be $x_k = x_{k-1} + r_k$. We constrain the step sizes to lie between -12 and $+12$, meaning that jumps cannot exceed an octave. This results in small, approximately normally distributed step sizes. This process yields a sequence of pitches $x_0, \dots, x_{K-1}$. As usual, we interpolate a step function through these pitches and sample $N = 100$ equally spaced pitches to obtain a random contour. In the figure above we use $N = 50$ for readability.



Here we vary the average length λ of the random walk baseline. This affects the number of notes K, but we still have $N = 100$ throughout. We generate 10,000 random contours, and compute the covariance matrices (A). The longer the melodies (larger K), the more the covariance matrix starts to resemble a Toeplitz matrix, which has constant values along each of its diagonals. As an ad-hoc measure of *Toeplitzness*, we measure how much every entry of the covariance matrix differs from the mean value on that diagonal. For a Toeplitz matrix, that should be zero everywhere: all diagonals are constant, so every entry also equals the mean of that diagonal. Column (B) makes clear that the covariance matrix differs from a Toeplitz matrix mostly in the upper left corner, which contains the covariance in the first timesteps. All this is also reflected in the principal components (C).

B2 Analyses of other datasets

In this supplement we visualize the principal components of melodic material from motifs to songs in different traditions. We include a subset here, please refer to the original supplements for the rest. That can be found on github.com/bacor/cosine-contours/blob/master/documents/supplements.pdf Here we show the following for every dataset:

A. The first four principal components. The first one is usually a flat line (gray), the second a descending shape (blue), the third a convex shape (orange), and the fourth one undulating (green). The corresponding cosines are shown as thin dashed lines in the same colors.

B. The length distribution of the melodic material, where length is measured in quarter notes. For Gregorian chant we assume all notes are quarter notes.

C. The covariance matrix.

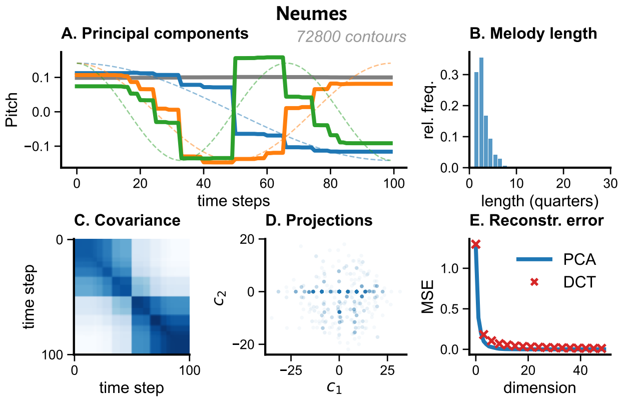
D. A scatterplot showing the representations of 2000 contours in 2d cosine contour space.

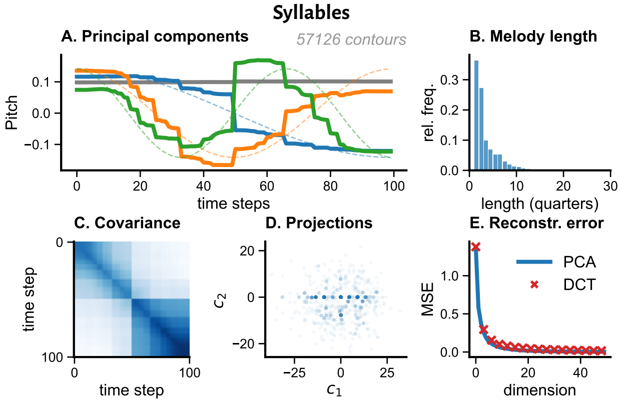
E. The reconstruction error using the discrete cosine transform compared to a principal component analysis.

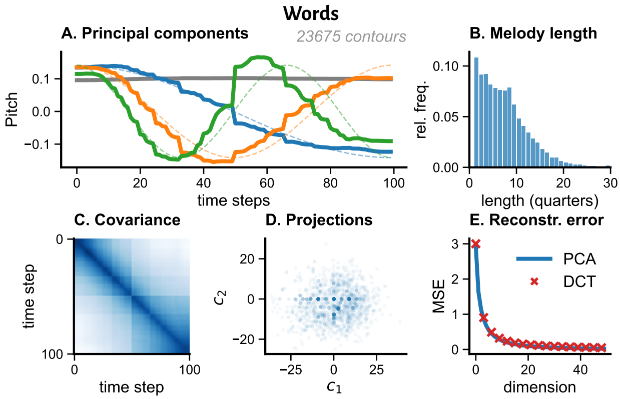
It is clear that the cosine approximation is most accurate at the phrase level. For very short melodic fragments (such as neumes or syllables), you see clear effects of the typical number of notes. For example, neumes often have only 2 notes, meaning there is a jump in the middle of the contour. You can see this in the principal components, but also in the covariance matrix. Such effects are weaker, but sometimes still visible at the phrase level: German folksongs apparently often have durations of 8 quarter notes, with jumps in the middle, or after 2 of 6 quarter notes. For complete songs, finally, the principal components are often difficult to interpret. Only for a very large number of songs (such as when combining all chants in GregoBase) does a pattern reminiscent of the cosines emerge. But for very small datasets, such as those in the Densmore collection, the principal components are very irregular.

Motifs

All motifs come from Gregorian chant (responsories from CantusCorpus).

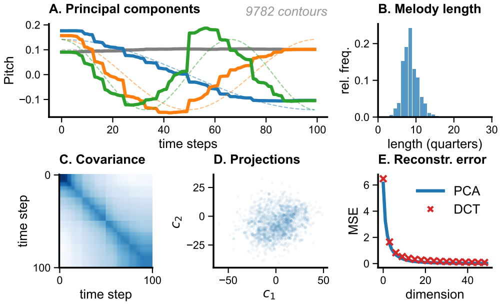




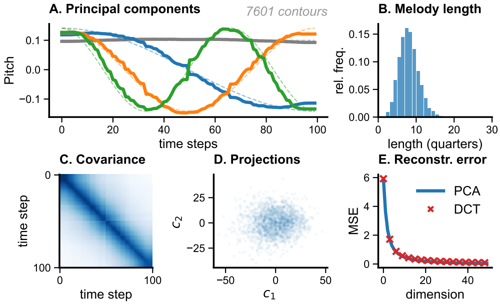


Phrases

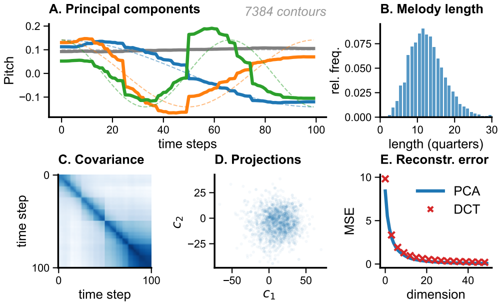
Erk (phrases)



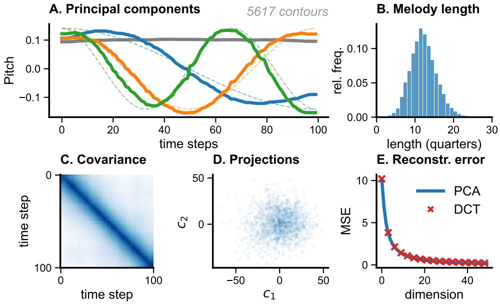
Erk (random segments)



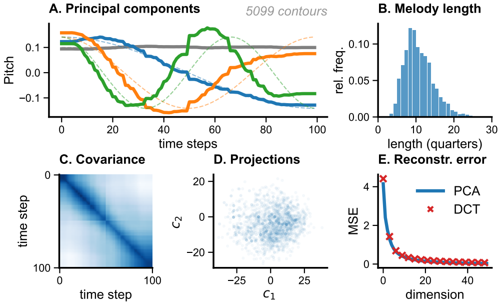
Han (phrases)



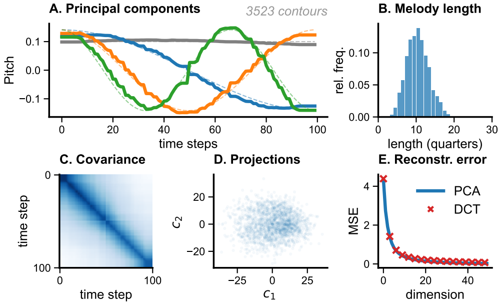
Han (random segments)



Antiphons (phrases)

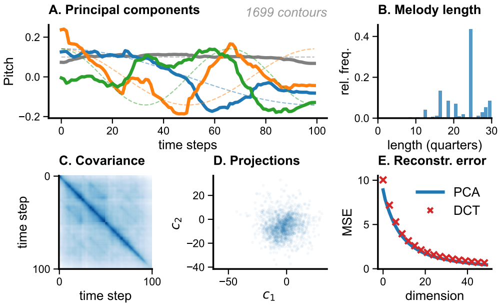


Antiphons (random segments)

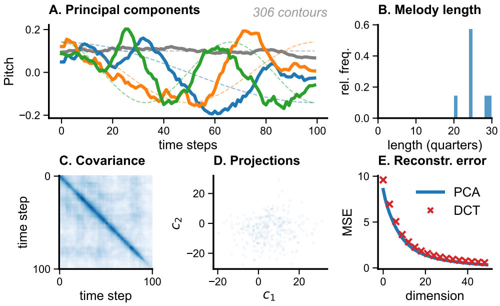


Songs

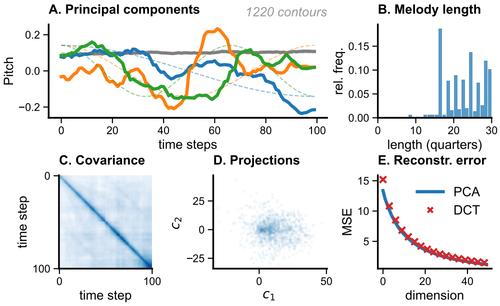
Erk



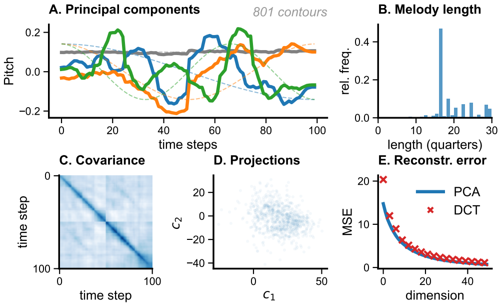
Böhme



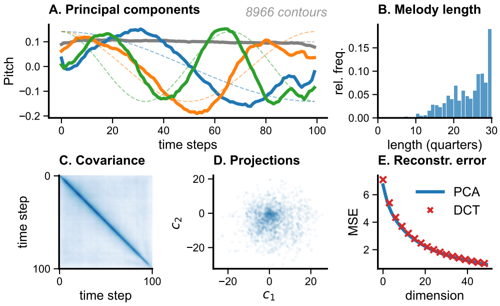
Han



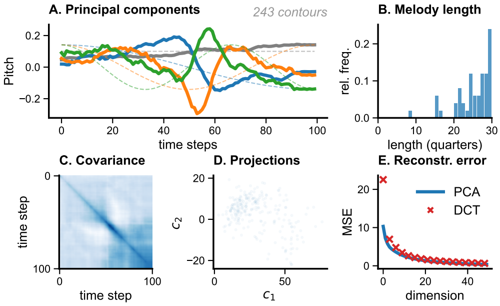
Shanxi



All chants in Gregobase



Lakota



B3 Mathematical background

In this supplement we provide some more mathematical background to illustrate why we observe cosine-shaped principal components. The aim is to make some of the key points a bit more accessible; we refer to Jolliffe (2002) for a detailed discussion of principal component analysis, to Gray (2006) for a rigorous treatment of Toeplitz matrices and their limiting behaviour, and to Rao and Yip (1990) for the discrete cosine transform.

NOTATION We write N for the length of a contour, or the number of steps in a random walk, and M denotes the number of contours. Consider a dataset $\{\mathbf{x}_1, \dots, \mathbf{x}_M\}$ of points $\mathbf{x}_m = (x_{m1}, \dots, x_{MN})$ in $\mathbb{R}^N$. We denote the sample mean by $\bar{\mathbf{x}}$ and the centered data points by $\hat{\mathbf{x}}_m$:

$$\bar{\mathbf{x}} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{x}_m \qquad \text{and} \qquad \hat{\mathbf{x}}_m = \mathbf{x}_m - \bar{\mathbf{x}}, \tag{B.1}$$

and both of course live in $\mathbb{R}^N$. An $M \times N$ matrix $\mathbf{X}$ has entries $(\mathbf{X})_{m,n} = x_{mn}$, and for $N \times N$ matrices we generally index rows by n and columns by k.

Principal components

MAXIMIZE PROJECTED VARIANCE The goal of a principal component analysis is to find a subspace of lower dimensionality $D < N$ that maximizes the variance of the data when it is projected on this subspace. First, we project the data on a one-dimensional subspace spanned by the unit vector $\mathbf{u}_1 \in \mathbb{R}^N$. You can think of the projection of $\mathbf{x}_n$ as a point in the N-dimensional ambient space, but we rather treat it as the scalar $\mathbf{u}_1^T\mathbf{x}_n$: the coordinate in the one-dimensional subspace. The projected data then has mean $\mathbf{u}_1^T\bar{\mathbf{x}}$ and variance

$$\frac{1}{M}\sum_{m=1}^{M}\left(\mathbf{u}_1^T\mathbf{x}_m - \mathbf{u}_1^T\bar{\mathbf{x}}\right)^2 = \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1, \tag{B.2}$$

where $\mathbf{S}$ is the $N \times n$ covariance matrix given by

$$\mathbf{S} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{x}_m - \bar{\mathbf{x}})(\mathbf{x}_m - \bar{\mathbf{x}})^T \tag{B.3}$$

We want to choose $\mathbf{u}_1$ in such a way that it maximizes the projected variance $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$. It can be shown (see e.g. Jolliffe, 2002), using a Lagrange multiplier, that under the constraint $\|\mathbf{u}_1\| = 1$, the projected variance is maximized when

$$\mathbf{S}\mathbf{u}_1 = \lambda_1\mathbf{u}_1. \tag{B.4}$$

Left-multiplying by $\mathbf{u}_1^T$, and using that $\mathbf{u}_1^T\mathbf{u}_1 = 1$, this is the case when

$$\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 = \lambda_1. \tag{B.5}$$

Equation (B.4) shows that $\mathbf{u}_1$ must be an eigenvector of the covariance matrix $\mathbf{S}$ corresponding to eigenvalue λ_1, which is exactly the projected variance according to (B.5). The first principal component, in short, is the eigenvector of the covariance matrix corresponding to the largest eigenvalue. The argument can be extended inductively to identify all principal components as eigenvectors of the covariance matrix, ordered according to their eigenvalues.

MINIMIZE RECONSTRUCTION ERROR It should be noted that one can also motivate principal components in another way. Consider a dataset $\{x_m \in \mathbb{R}^N\}_m$ as before, and a set of basis vectors $\{\mathbf{u}_1, \dots, v_N\}$ for $\mathbb{R}^N$ with norm 1. As before, the projection of $\mathbf{x}$ on the $\mathbf{u}_n$ is $c_n = \mathbf{u}_n^T \mathbf{x}$, and so we can represent $\mathbf{x}$ as a coordinate vector $(c_0, \dots, c_N)$. Now suppose we only use the first D coordinates to represent $\mathbf{x}$, so we get the truncated representation:

$$\tilde{\mathbf{x}} = \sum_{i=1}^{D} c_i \mathbf{u}_i. \tag{B.6}$$

Now measure the *reconstruction error* as

$$\text{MSE} = \frac{1}{M} \sum_{m=1}^{M} (\mathbf{x} - \tilde{\mathbf{x}})^2 \tag{B.7}$$

We ask: how should we choose the basis vectors so that the reconstruction error MSE is minimized? The answer is the same: as the eigenvectors, ranked in descending order (Rao & Yip, 1990).

Toeplitz and circulant matrices

Toeplitz matrices are matrices were every diagonal has the same value. They are usually indexed as follows:

$$\mathbf{T} = \begin{bmatrix} t_0 & t_{-1} & t_{-2} & \cdots & t_{-(N-1)} \\ t_1 & t_0 & t_{-1} & & \\ t_2 & t_1 & t_0 & & \vdots \\ \vdots & & & \ddots & \\ t_{N-1} & & & \cdots & t_0 \end{bmatrix} \tag{B.8}$$

That means that $T_{i,j} = t_{j-i}$. Before we discuss Toeplitz matrices further, let's focus on the special subset of circulant matrices. A *circulant matrix* is a Toeplitz matrix where every row equals the previous row, rotated one step to the right:

$$\mathbf{C} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{N-1} \\ c_{N-1} & c_0 & c_1 & & c_{N-2} \\ c_{N-2} & c_{N-1} & c_0 & & \\ \vdots & & \ddots & & \vdots \\ & & & c_0 & c_1 \\ c_1 & & \cdots & c_{N-1} & c_0 \end{bmatrix} \tag{B.9}$$

It is convenient to start indexing at 0 rather than 1 so that we can write $\mathbf{C}_{n,k} = c_{k-n \bmod N}$. We will also read the subscripts periodically: for example, $c_{N+3} = c_3$. For circulant matrices, matrix multiplication takes the form of a *circular convolution*: if $\mathbf{y} = \mathbf{C}\mathbf{x}$, we have

$$y_n = \sum_{k=0}^{N-1} c_{k-n} x_k. \tag{B.10}$$

EIGENVECTORS OF CIRCULANT MATRICES Suprisingly, all circulant matrices have the same eigenvectors. These eigenvectors consist of *(N-th) roots of unity*: the complex

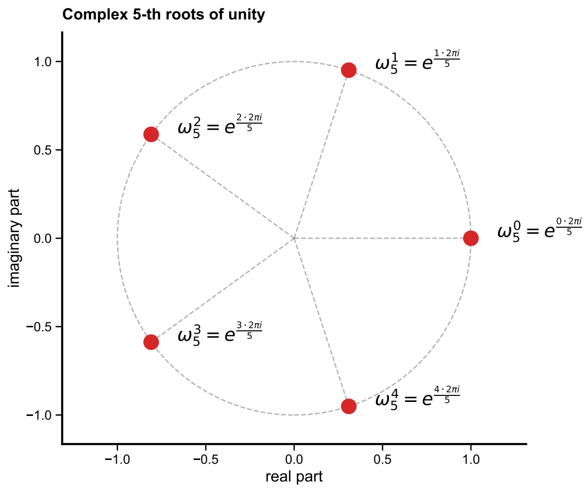


FIGURE B.1 – The N-th roots of unity for $N = 5$ are points on the complex unit circle.

numbers z satisfying $z^N = 1$. The first complex root of unity is

$$\omega = e^{\frac{2\pi i}{N}}, \tag{B.11}$$

and its powers ω^k are other roots of unity, since $(\omega^k)^N = (\omega^N)^k = 1$. The numbers $\omega^0, \dots, \omega^{N-1}$ can be visualized as evenly spaced points on the unit circle in the complex plane (see Figure B.1). Importantly, these numbers (like the coefficients c_k) are periodical: $\omega^{N+k} = \omega^N \cdot \omega^k = \omega^k$.

This property allows us to show that the N eigenvectors of a circulant matrix are

$$\boldsymbol{\omega}_n = (\omega^{n \cdot 0}, \dots, \omega^{n \cdot (N-1)}),, \tag{B.12}$$

for $n = 0, \dots, N-1$. You can verify this directly when $n = 0$, since $\boldsymbol{\omega}_0$ is then an an all-ones vector, but let's consider the general case. We have to show that $\mathbf{C}\boldsymbol{\omega}_n = \lambda_n \boldsymbol{\omega}_n$ for some constant λ_n. Using (B.10), we can show that k'the entry of the left hand side indeed equals $\lambda_n \omega^{nk}$:

$$(\mathbf{C}\boldsymbol{\omega}_n)_k = \sum_{j=0}^{N-1} c_{j-k} \cdot \omega^{n \cdot j} \tag{B.13}$$

$$= \omega^{nk} \cdot \sum_{j=0}^{N-1} c_{j-k} \cdot \omega^{n(j-k)} \tag{B.14}$$

$$= \omega^{nk} \cdot \underbrace{\sum_{j'=0}^{N-1} c_{j'} \cdot \omega^{n \cdot j'}}_{\lambda_n} . \tag{B.15}$$

Here we first multiplied by $\omega^{-nk}/\omega^{-nk}$ to align the indices of the coefficients and the powers. Then we used the periodicity of the roots of unity to reorder the sum, so it no longer depends on k and must equal the eigenvalue λ_n. The general case is similar.

Summarizing, every $N \times N$ circulant matrix $\mathbf{C}$ has the same N eigenvectors $\omega_0, \dots, \omega_n$, with (different) corresponding eigenvalues:

$$\lambda_n = c_0\omega^0 + c_1\omega^n \dots c_{N-1}\omega^{n(N-1)} \tag{B.16}$$

$$= \sum_{j=0}^{N-1} c_j e^{\frac{2\pi i \cdot nj}{N}}, \tag{B.17}$$

for $n = 0, \dots, N-1$. From the second expression one sees that the eigenvalues $(\lambda_0, \dots, \lambda_{N-1})$ are the discrete Fourier transform of $(c_0, \dots, c_{N-1})$.

REAL CIRCULANT MATRICES In the scenario we are interested in, the matrix $\mathbf{C}$ is real and symmetric, and such matrices have real eigenvalues and eigenvectors. To see that the eigenvalues are real, first note that a symmetric circulant matrix satisfies the additional constraint $c_k = c_{N-k}$. Also observe that ω^k and $\omega^{N-k} = \omega^{-k}$ are each others mirror image in the real axis (see Figure B.1). They have the same real part,

$$\mathrm{Re}(\omega^k) = \cos\left(\frac{2\pi k}{N}\right), \tag{B.18}$$

and when adding them, the complex part cancels out: $\omega^k + \omega^{-k}$ lies on the real axis, at the point $2\mathrm{Re}(\omega^k)$. This means that

$$c_k\omega^k + c_{N-k}\omega^{N-k} = 2c_k\mathrm{Re}(\omega^k) \tag{B.19}$$

is a real number. From (B.16) we see that the eigenvalues λ_n consist of many such sums: all complex parts cancel out and the eigenvalues are real[1]

Now we can also choose real eigenvectors: the real part of ω_n. After all, if ω_n is an eigenvector for the real eigenvalue λ_n, so are ω_{-n} and $\mathbf{v}_n = 1/2(\omega_n + \omega_{-n})$. By the same argument as before, equations (B.19) and (B.18) show that this is a real eigenvector:

$$\mathbf{v}_n = \Big(1,\ \cos\theta,\ \dots,\ \cos N\theta\Big), \quad \theta = \frac{2\pi n}{N}. \tag{B.20}$$

This is a discrete cosine function consisting of N points, where higher n implies in higher frequencies. This is illustrated in Figure B.2.

TOEPLITZ IS ASYMPTOTICALLY CIRCULANT The reason circulant matrices are interesting here, is that Toeplitz matrices can be shown to be asymptotically equivalent to circulant matrices, and that eigenvalues are preserved. We refer to Gray (2006) for a detailed discussion of that result. What this implies is that the eigenvectors of large Toeplitz matrices are well approximated by those of circulant matrices: sinusoidal functions. That in turn means that approximately Toeplitz covariance matrices (which are real and symmetric) will have cosine-shaped eigenvectors.

PCs of random processes

We want to end by discussing two examples where Toeplitz covariance structures arise, and we thus would expect cosine eigenvectors, at least asymptotically.

WEAKLY STATIONARY PROCESS Toeplitz matrices arise in the study of weakly stationary processes. These are random processes where the mean is constant over time, and where the covariance does not change by shifts in time: it only depends on

[1] The expression for the eigenvalues is slightly different depending on whether N is odd or even.

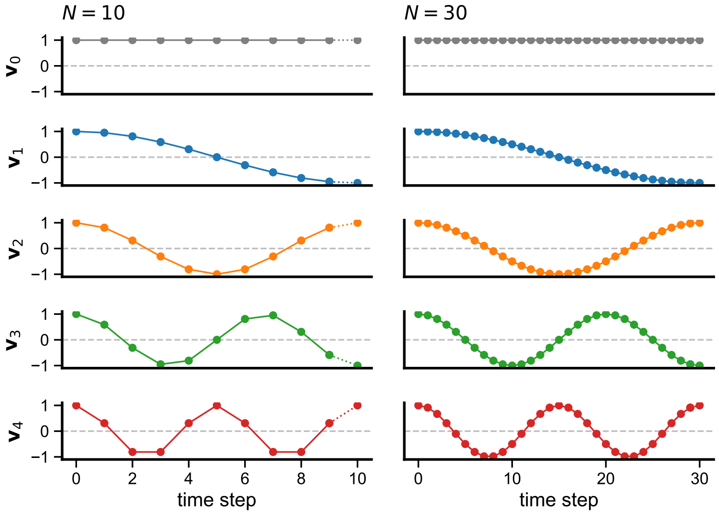


FIGURE B.2 – The eigenvectors of a symmetric, circulant matrix are discrete cosine functions with different periods.

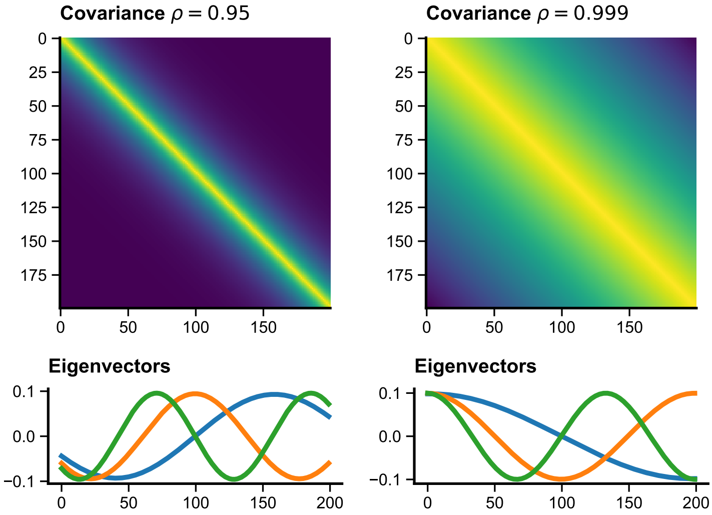


FIGURE B.3 – The autocovariance matrix for an autoregressive process AR(1) for two values of ρ. When $\rho \rightarrow 1$ it approximates the discrete cosine transform.

the distance between two time steps. That is, when $\mathrm{Cov}(x_i, x_j) = K(j - i)$ is some function of $j - i$, and thus results in a Toeplitz covariance matrix.

One example of such a process is a first order autoregressive process AR(1), where

$$x_n = \rho x_{n-1} + r_n, \tag{B.21}$$

where r_n is a random step with mean zero and variance σ^2, and we assume $x_0 = 0$. It can be shown that this process has mean $E[x_n] = 0$ and variance $\mathrm{Var}[x_t] = {}^1/_{1-\rho^2}$ if $|\rho| < 1$. In that case, the covariance is

$$\mathrm{Cov}(x_i, x_j) = \frac{\sigma^2}{1-\rho^2} \cdot \rho^{|j-i|}. \tag{B.22}$$

This is actually one of the few cases where an analytic expression for the eigenvectors is known, although it is rather complex (Rao & Yip, 1990; Ray & Driver, 1970). Interestingly, one can use this to show that for AR(1) processes, the discrete cosine transform DCT-II becomes equivalent to the 'principal component transform' (Karhunen-Loève transform) as $\rho \to 1$ (Rao & Yip, 1990, section 3.3.2).

HIGH-DIMENSIONAL RANDOM WALK In the limit $\rho \to 1$ one obtains a random walk. Antognini and Sohl-Dickstein (2018) analyse the principal components of high-dimensional random walks. We briefly summarise their results. Consider a random walk in $\mathbb{R}^M$ with N steps given by

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \mathbf{r}_n \tag{B.23}$$

where $\mathbf{r}_n$ is a random step drawn from a probability distribution with zero mean and a finite, normalized covariance matrix. We start from $\mathbf{x}_0 = \mathbf{0}$ in $\mathbb{R}^M$.

We can express all this as matrix multiplications. Collect the points $\mathbf{x}_n$ and steps $\mathbf{r}_n$ as the rows of the $N \times M$ matrices $\mathbf{X}$ and $\mathbf{R}$ respectively. Let $\mathbf{W}$ be a $N \times N$ matrix with 1's on the diagonal, -1's on the subdiagonal and zeros elsewhere. This implements the walking mechanism in the sense that $\mathbf{WX} = \mathbf{R}$, hence

$$\mathbf{X} = \mathbf{W}^{-1}\mathbf{R}. \tag{B.24}$$

To compute the covariance matrix $\mathbf{S}$ we need the centered datapoints $\hat{\mathbf{x}}_n = \mathbf{x}_n - \bar{\mathbf{x}}_n$. The centering operation be conveniently expressed as multiplication by the $N \times N$ *centering matrix* $\mathbf{C} = \mathbf{I} - \frac{1}{M}\mathbf{J}$, where $\mathbf{J}$ is the all-ones matrix. This gives

$$\hat{\mathbf{X}} = \mathbf{CX} = \mathbf{CW}^{-1}\mathbf{R} \tag{B.25}$$

and allows us to express the covariance matrix as $\mathbf{S} = \frac{1}{N}\hat{\mathbf{X}}^T\hat{\mathbf{X}}$. Instead of finding the eigenvectors of $\hat{\mathbf{X}}^T\hat{\mathbf{X}}$, we can look for those of $\hat{\mathbf{X}}\hat{\mathbf{X}}^T$. After all, if $\mathbf{u}$ is an eigenvector for $\hat{\mathbf{X}}^T\hat{\mathbf{X}}$ with nonzero eigenvalue λ, then $\mathbf{v} = \hat{\mathbf{X}}\mathbf{u}$ is the corresponding eigenvector for $\hat{\mathbf{X}}\hat{\mathbf{X}}^T$.

Putting all this together, Antognini and Sohl-Dickstein (2018) look for the eigenvalues and eigenvectors of

$$\hat{\mathbf{X}}\hat{\mathbf{X}}^T = \mathbf{CW}^{-1}\mathbf{RR}^T\mathbf{W}^{-T}\mathbf{C} \tag{B.26}$$

where we used symmetry of $\mathbf{C}$. Note that this matrix contains the covariance between timesteps, rather than dimensions. They observe that in the limit of infinte dimensionality $M \to \infty$, we have that $\mathbf{RR}^T$ tends to the $N \times N$ identity matrix. This allows us to simplify (B.26) to

$$\hat{\mathbf{X}}\hat{\mathbf{X}}^T = \mathbf{C}\mathbf{W}^{-1}\mathbf{W}^{-T}\mathbf{C}. \tag{B.27}$$

Since **W** is a so-called banded Toeplitz matrix, and **C** is circulant, the whole expression can be shown to be asymptotically equivalent to a circulant matrix, meaning that the eigenvectors are cosines. This analysis can be related to melodic contours, when we consider a collection of M contours of length N as one high-dimensional walk through $\mathbb{R}^M$.

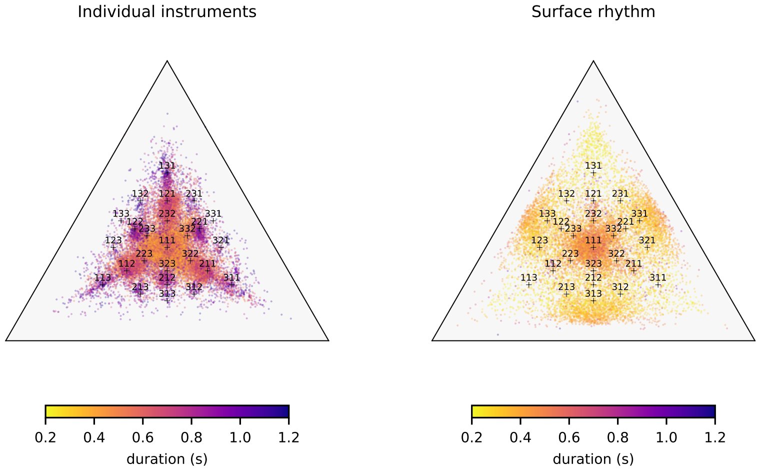
SUPPLEMENT C

Supplements for chapter 7: Rhythm triangles

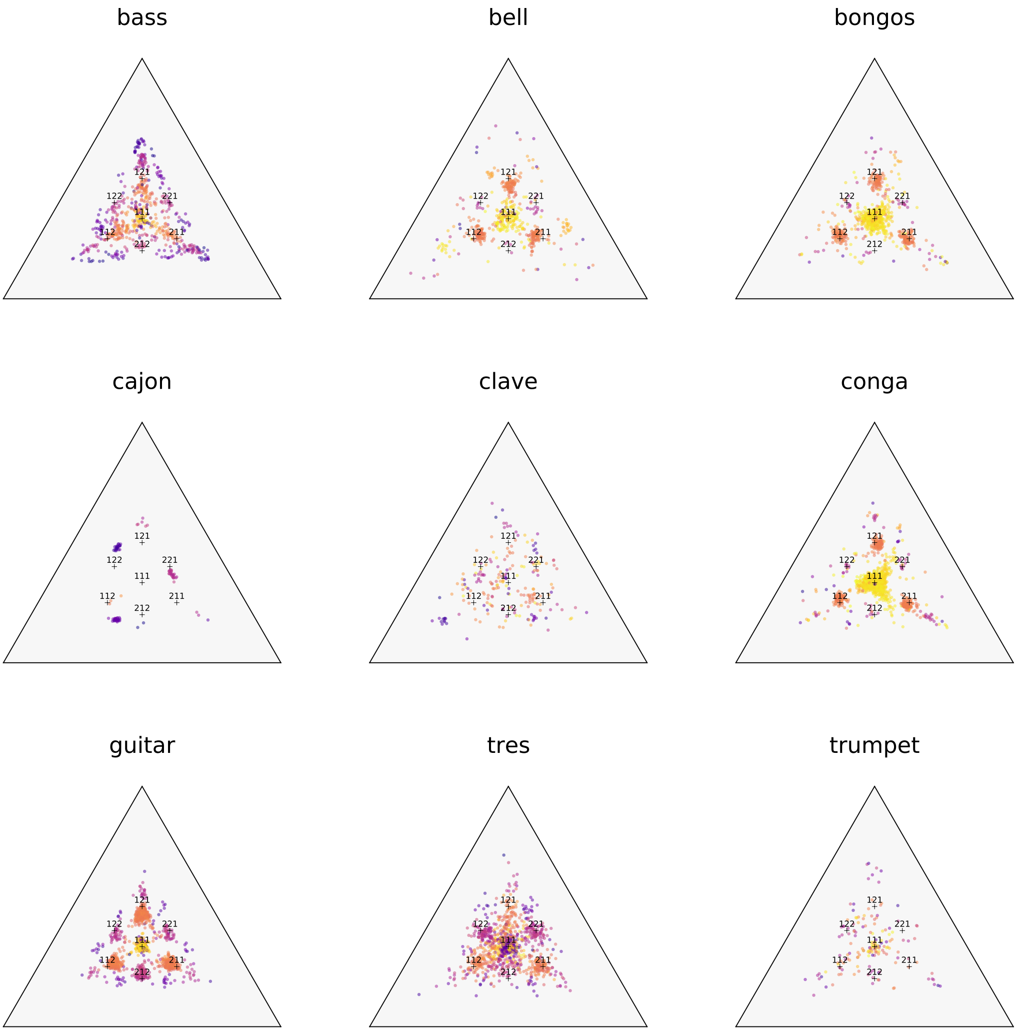
C1 IEMP Cuban Salsa and Son

Here we show the IEMP Cuban Salsa and Son data in more detail.

INDIVIDUAL INSTRUMENTS VS. SURFACE First, we compare the *individual instruments* versus the *surface rhythm*. The former considers the intervals between onsets of a single instrument and then concatenates the intervals for all instruments. The latter, surface rhythm, is obtained by computing intervals between successive onsets of any two instruments. Onsets fewer than 25ms apart are considered to be simultaneous and ignored (cf. Roeske et al., 2020). The two are quite different, which is surprising given that Roeske et al. (2020) write that overall, “the two types of extraction (separate for each instrument and simultaneous for combined ‘surface’ rhythm) produced similar results”.



DIFFERENT INSTRUMENTS Second, we compare different instruments within *Song 1*. Different instruments clearly play different rhythms, but there is also a lot of overlap. The triangle plots also appear to show *timing*: the mode around 1 : 1 : 1 for the conga, for example, has a peculiar triangular shape. And the Cajon is very reliably avoiding the 1 : 2 : 2 ratios.



bass

bell

bongos

cajon

clave

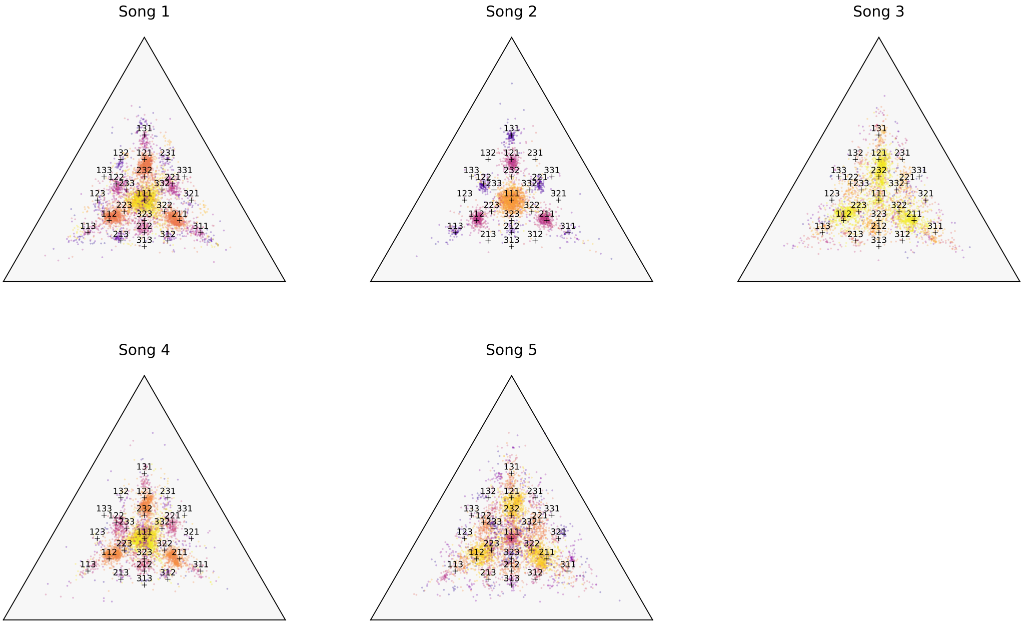
conga

guitar

tres

trumpet

DIFFERENT SONGS Third, we compare the five different songs. Most of the rhythmic categories are found in all songs, but song 2 appears to be more clearly timed and song 3 is a bit slower.



C2 Measuring isochrony

In this section, I give a formal definition of the proposed measure of isochrony, first for motifs of length $n = 2$ and then for general lengths. It will be convenient to define isochrony in terms of its opposite, which I will call *anisochrony*.

LENGTH 2 Take a sequence of inter-onset intervals $i_1, \dots, i_K$, and group them into *normalized* motifs of length 2 (normalized 2-*grams*):

$$r_k = \Big(\frac{i_k}{i_k + i_{k+1}}, \frac{i_{k+1}}{i_k + i_{k+1}}\Big). \tag{C.1}$$

By normalizing, we capture the duration of each interval *relative to* the total duration of the motif. The sequence of intervals $(1, 2, 4, 3, 3)$ for example gives motifs $(1/3, 2/3)$, $(1/3, 2/3)$, $(4/7, 3/7)$ and $(1/2, 1/2)$.

One can think about the nPVI as measuring the average “irregularity” of such motifs. Or, to propose a technical term, the *anisochrony*, from Greek *ánisos* “unequal”: it really measures how distant a motif (a, b) is from the isochronous motif $(1/2, 1/2)$; how not-isochronous a motif is. We can measure the distance from isochrony as $|a - 1/2| + |b - 1/2|$. For example, the anisochrony of the motif $(0.6, 0.4)$ is 0.2, precisely the adjustments needed to turn the motif into an isochronous rhythm.

More formally, we define the *anisochrony* of a motif $r_k = (a, b)$ to be its distance to the isochronous motif $(1/2, 1/2)$:

$$\text{anisochrony}((a,b)) := \; |a - \frac{1}{2}| + |b - \frac{1}{2}| \; = \; |a - b| \tag{C.2}$$

To see that the second equality holds, note that $b = 1 - a$, so that the left-hand side equals $|a - 1/2| + |1/2 - (1-a)| = 2|a - 1/2| = |a - (1-a)| = |a - b|$. We define the nPVI as 200 times the average anisochrony. To see that this is exactly equivalent to the conventional definition of nPVI, we fill in the definitions of a and b in terms of intervals:

$$\text{nPVI} = 200 \times \frac{1}{K-1} \sum_{k=1}^{K-1} \left| \frac{i_k}{i_k + i_{k+1}} - \frac{i_{k+1}}{i_k + i_{k+1}} \right| \tag{C.3}$$

$$= \frac{100}{1/2} \times \frac{1}{K-1} \sum_{k=1}^{K-1} \left| \frac{i_k - i_{k+1}}{i_k + i_{k+1}} \right| \tag{C.4}$$

$$= \frac{100}{K-1} \sum_{k=1}^{K-1} \left| \frac{i_k - i_{k+1}}{1/2(i_k + i_{k+1})} \right|. \tag{C.5}$$

And this latter form is the usual definition.

LONGER MOTIFS The more general definition suggests a natural generalization to longer motifs. In the rhythm triangle, for example, we find 3-gram motifs (a, b, c), and their anisochrony would be their distance to the isochronous triplet $(1/3, 1/3, 1/3)$ at the center of the triangle. And this can easily be extended to even longer motifs of arbitrary length n. To do so, observe that we have implicitly used the L_1 norm to define distances. Essentially, for a normalized motif $r = (r_1, \dots, r_n)$, we defined

$$\text{anisochrony}(r) = C_n \cdot \|r - I\|_1, \qquad I = (1/n, \dots, 1/n) \tag{C.6}$$

where I is the isochronous motif and C_n should be a normalizing constant such that the anisochrony falls between 0 and 1. Now note that the motifs furthest away from I are the motifs at the corners of the space: those with $n-1$ zeros and a single one. Their distance to I is

$$\left|1 - \frac{1}{n}\right| + (n-1) \cdot \left|0 - \frac{1}{n}\right| = \frac{n-1}{n} + \frac{n-1}{n} = \frac{2(n-1)}{n}. \tag{C.7}$$

And so we choose $C_n = n/(2(n-1))$:

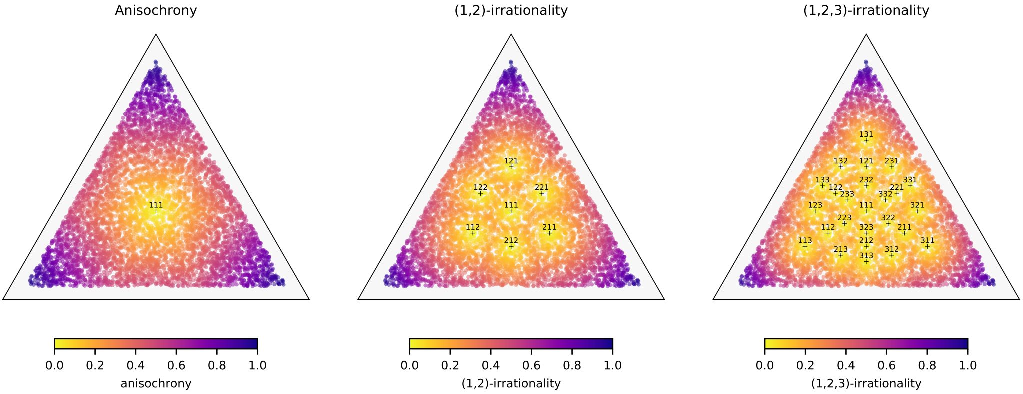
$$\text{anisochrony}(r) = \frac{n}{2(n-1)} \sum_{k=1}^{n} \left| r_k - \frac{1}{n} \right| \tag{C.8}$$

When $n = 2$ all this boils down to the exact same definition as before. And indeed to wrap things up, we let isochrony$(r) = 1 -$ anisochrony(r), which takes values between 0 (maximally non-isochronous) and 1 (perfectly isochronous). For a finishing touch, let's turn the definition around and define the *(n-gram) isochrony of r* as

$$\text{isochrony}(r) = 1 - \alpha(r), \tag{C.9}$$

which takes values between 0 (maximally non-isochronous) and 1 (perfectly isochronous).

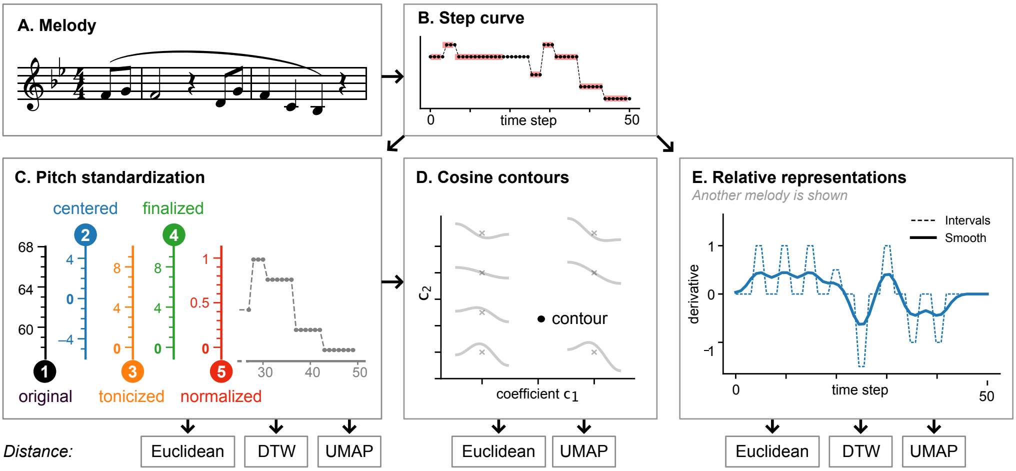
AND BEYOND? The core idea of all this is simple: you measure distances from a motif to some reference motif like the isochronous one. But you can use other reference points. All small-integer ratios, for example, and if you then take the minimum over all those distances, we get the *irrationality* of a motif: how far it is from the closest small-integer ratio motif:



SUPPLEMENT D

Supplements for chapter 8: Contour typology

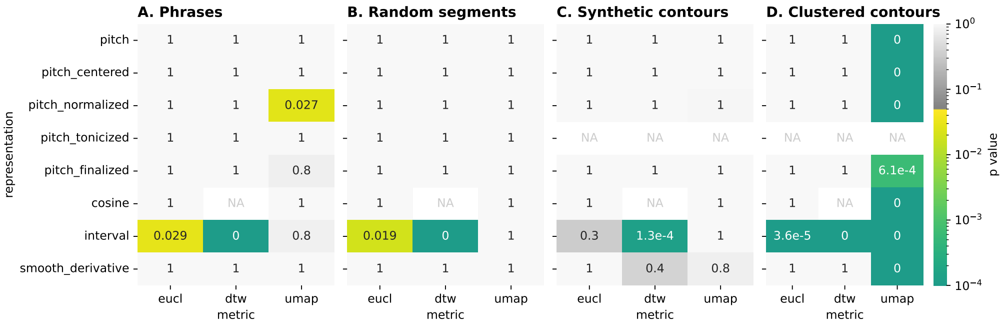
D1 Experimental setup



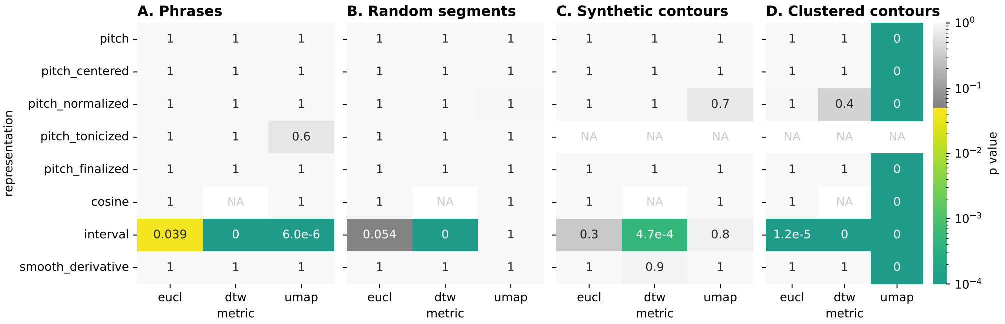
We represent all melodic fragments, be it phrases or random segments **(A)**, as sequences of 50 pitches, sampled from step curves interpolating the melodies **(B)**. Then we standardize the pitch in 5 ways **(C)**: not at all, by centering, normalizing, tonicizing (not possible for synthetic contours), and finalizing. Moreover, we compute a 50-dimensional cosine contour representation (**D**, illustrated in 2d), and two relative representations **(E)**: an interval representation and a smoothed version thereof. Finally, we compute pairwise distances using Euclidean distance, DTW dissimilarity or (Euclidean) distance in a 10-dimensional UMAP projection. It is worth noting that when we compute DTW similarity on (smooth) interval representations (i.e., on the derivative of the time series) we are effectively using *derivative dynamic time warping* (Keogh & Pazzani, 2001). This variant of DTW was proposed to make DTW more robust to small changes in the time series, and usually results in better alignments. We do not use DTW for cosine contours, as it consists of discrete cosine transform coefficients, which do not form a time series.

D2 Clusterability of contours

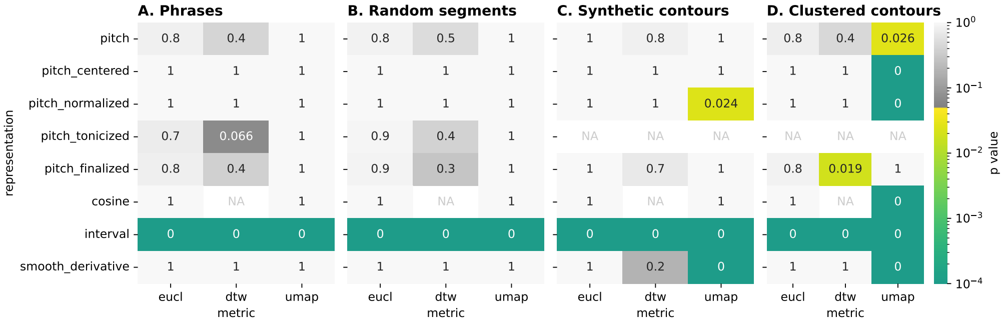
Here we show the *p*-values of the Hartigans' dip test on the set of pairwise distances between contours, using Euclidean, DTW and UMAP distance. The color coding is the same as in Figure 8.4 and is yellow-green for significant results, and gray for insignificant results, using a significance threshold of 0.05. Only with UMAP distance does the test correctly provide evidence for multimodality in the clustered dataset **(D)**. Note, also, that the interval representation finds more evidence for multimodality across all four datasets—even in the uniform, synthetic dataset **(C)**. But since that dataset is synthesized to contain *no* clusters, we treat this as a false positive.



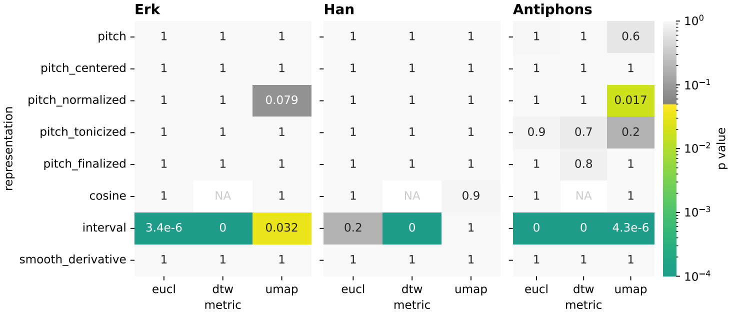
UNIQUE CONTOURS ONLY We repeated the analyses on samples of unique contours, and the overall pattern remains the same.



LOWER DIMENSIONALITY We repeated the analyses on lower-dimensional contour representations of 10 rather than 50 pitches, by subsampling the 50-dimensional contours, or in the case of cosine contours, taking only the first 10 coefficients. Again, the overall pattern remains the same:

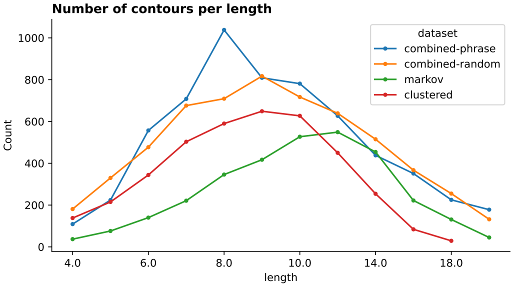


PER DATASET The results for ‘phrases’ above all use the aggregate, cross-cultural dataset. Here we show the results for three datasets separately. Again, the overall pattern remains the same.

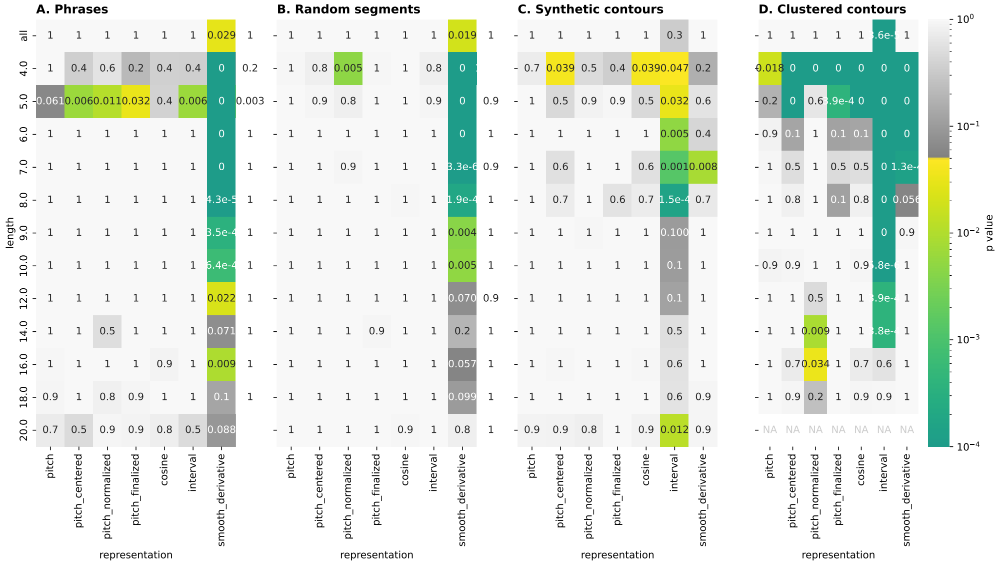


D3 Length-wise analysis

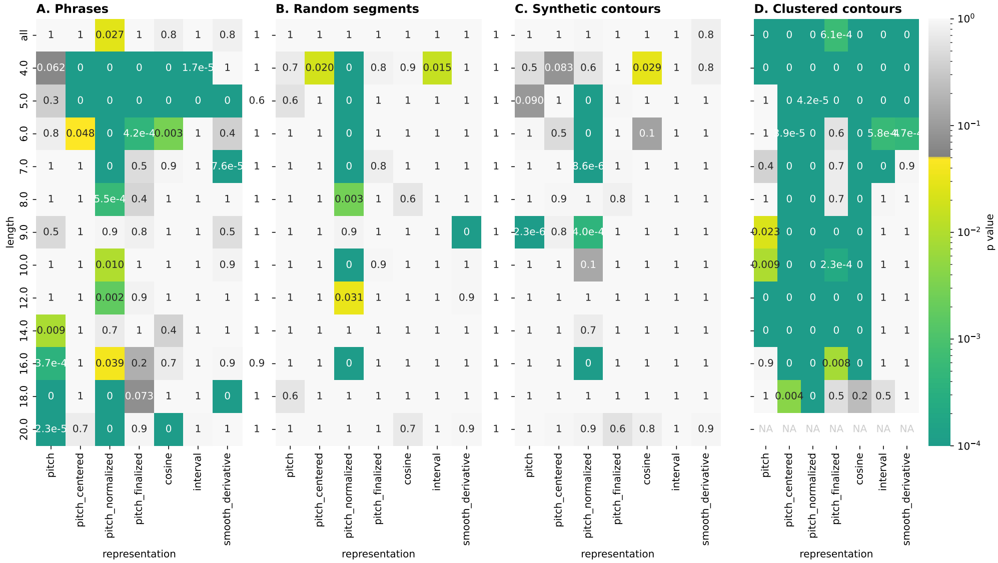
The length of a phrase may affect its shape, and perhaps we don't find clusters because we aggregate all lengths. We thus repeat the analyses, but now for every length (measured in the number of notes) separately. First, this it the distribution of lengths in the datasets:



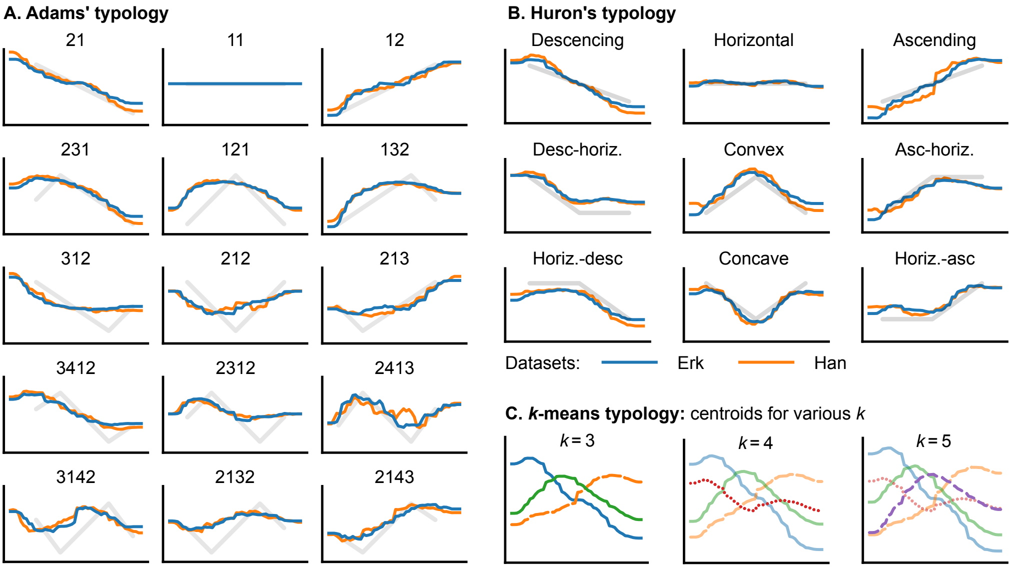
EUCLIDEAN DISTANCE Next, we show the same *p*-values as before, but now the length is shown vertically, and the representation horizontally. With Euclidean distance, we only see evidence appearing for some clusters of very short contours of 4–5 notes. This is not surprising: the space of possibilities is small, and there are only a few such contours. Indeed, even uniform synthetic contours of length four can appear clustered. Note that many of the synthetically clustered contours still avoid detection.



UMAP-DISTANCE With UMAP distance, we see more evidence for clustering, but again primarily for shorter motifs, and in particular with the normalized pitch representation. There is also some clustering for longer phrases. But for the most common phrases of average length, that evidence is largely absent and certainly not nearly as strong as the evidence for clustering in the synthetic, clustered dataset.



D4 Average shapes



We show the average of all contours with a certain type for Adams' **(A)** and Huron's typology **(B)**, for two datasets: Erk (blue) and Han (orange). The theoretical shape is shown in the background (see Figure 8.1). For the k-means typology **(C)** we show the centroids for $k = 3$, 4 and 5 clusters. Similarly shaped centroids are similarly colored across values of k. The shapes in smaller typologies (k-means or Huron's) are more recognizable than those in Adam's typology. Also note the characteristic flattening at the beginning and end of each contour, caused by every first and final note necessarily being flat.

SUPPLEMENT E

Supplements for chapter 10: Algo Pärt

Contents

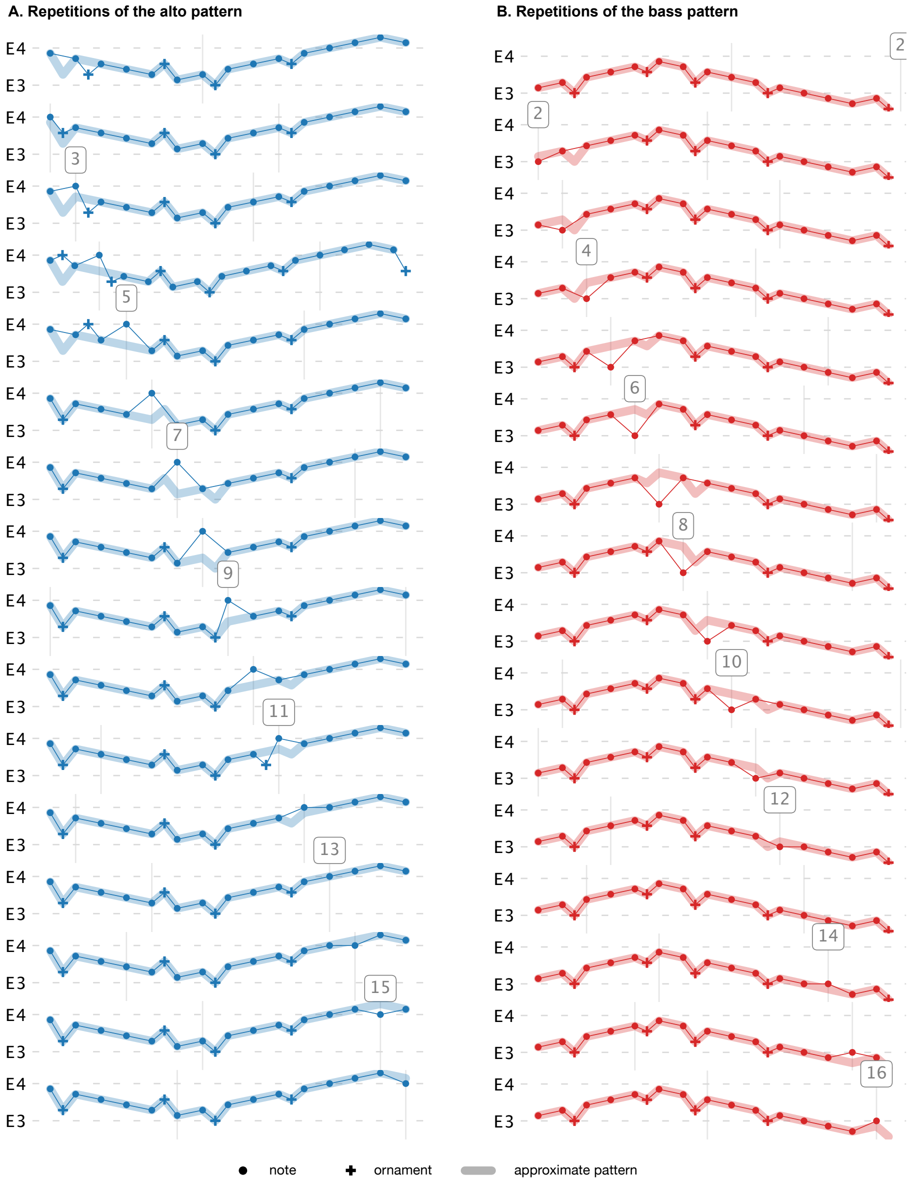
E1 Textual structure of *Summa*

The piece consists of 16 sections, marked by rehearsal numbers, of each 3 bars. Measures in a section contain 7, 9 and 7 syllables respectively, and use different voices: SA, then SATB and finally TB. The next section mirrors this structure. Syllables are distributed across the bars following this scheme, even if this means that a bar line falls in the middle of a word.

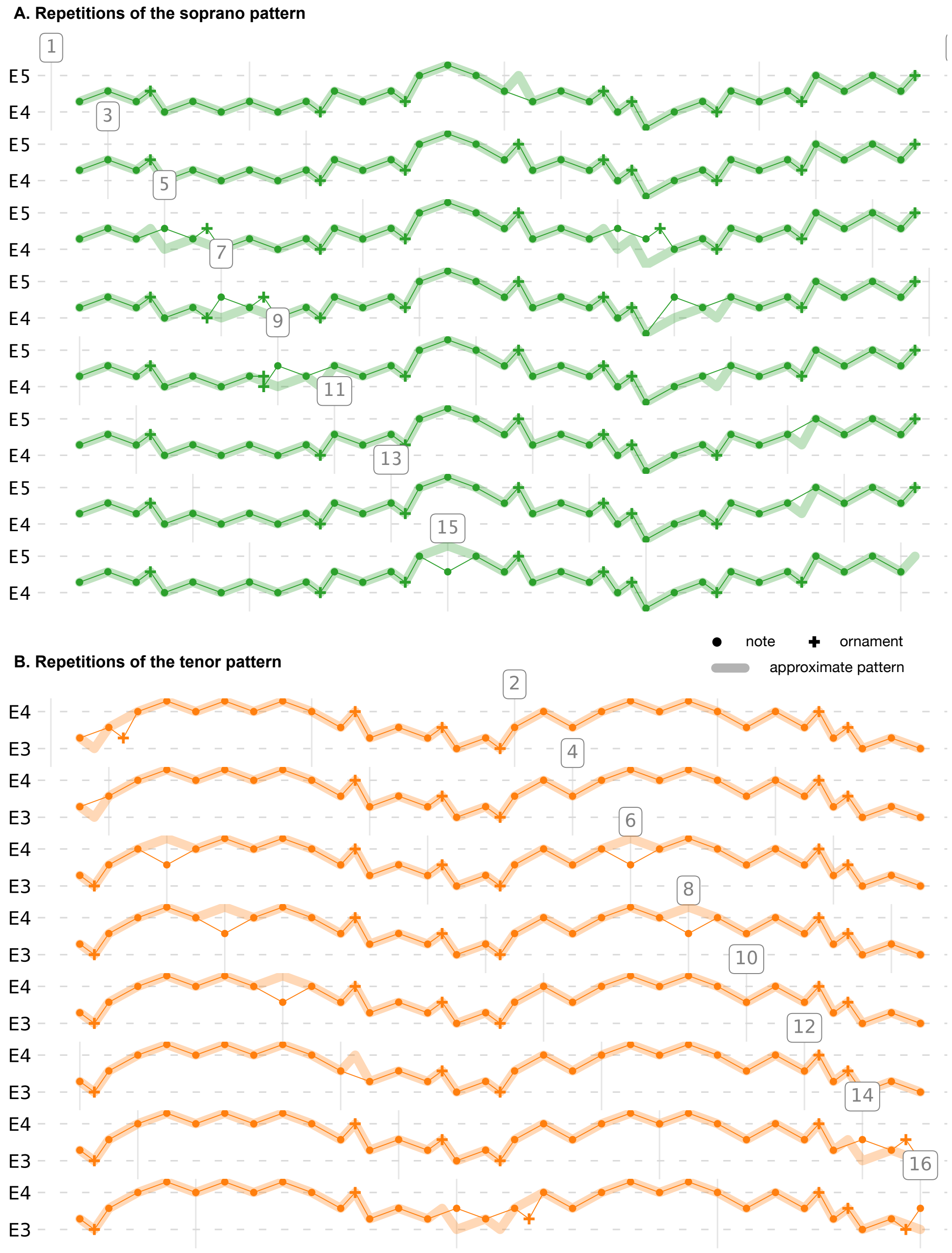
Section	Bar	Voices	Syllables	Text								
1	1	SA	7	Cre-	do	in	u-	num	de-	um,		
	2	SATB	9	Pa-	trem	o-	mni-	po-	ten-	tem,	fa-	cto-
	3	TB	7	rem	coe-	li	et	ter-	rae,	vi-		
2	4	TB	7	si-	bi-	li-	um	o-	mni-	um,		
	5	SATB	9	et	in-	vi-	si-	bi-	li-	um,	et	in
	6	SA	7	u-	num	Do-	mi-	num	Je-	sum		
3	7	SA	7	Chri-	stum,	Fi-	li-	um	De-	i		
	8	SATB	9	u-	ni-	ge-	ni-	tum,	et	ex	Pa-	tre
	9	TB	7	na-	tum	an-	te	o-	mni-	a		
4	10	TB	7	sae-	cu-	la.	De-	um	de	De-		
	11	SATB	9	o,	lu-	men	de	lu-	mi-	ne,	De-	um
	12	SA	7	ve-	rum	de	De-	o	ve-	ro,		
5	13	SA	7	ge-	ni-	tum,	non	fa-	ctum,	con-		
	14	SATB	9	sub-	stan-	ti-	a-	lem	Pa-	tri:	per	quem
	15	TB	7	o-	mni-	a	fac-	ta	sunt.	Qui		
6	16	TB	7	prop-	ter	nos	ho-	mi-	nes,	et		
	17	SATB	9	pro-	pter	no-	stram	sa-	lu-	tem	de-	scen-
	18	SA	7	dit	de	coe-	lis.	Et	in-	car-		
7	19	SA	7	na-	tus	est	de	Spi-	ri-	tu		
	20	SATB	9	San-	cto	ex	Ma-	ri-	a	Vir-	gi-	ne:
	21	TB	7	Et	ho-	mo	fa-	ctus	est.	Cru-		
8	22	TB	7	ci-	fi-	xus	e-	ti-	am	pro		
	23	SATB	9	no-	bis	sub	Pon-	ti-	o	Pi-	la-	to
	24	SA	7	pas-	sus	et	se-	pul-	tus	est.		
9	25	SA	7	Et	re-	sur-	re-	xit	ter-	ti-		
	26	SATB	9	a	di-	e,	se-	cun-	dum	scri-	ptu-	ras.
	27	TB	7	Et	a-	scen-	dit	in	coe-	lum,		
10	28	TB	7	se-	det	ad	dex-	te-	ram	Pa-		
	29	SATB	9	tris.	Et	i-	te-	rum	ven-	tu-	rus	est
	30	SA	7	cum	glo-	ri-	a,	ju-	di-	ca-		
11	31	SA	7	re	vi-	vos	et	mor-	tu-	os,		
	32	SATB	9	cu-	jus	re-	gni	non	e-	rit	fi-	nis.
	33	TB	7	Et	in	Spi-	ri-	tum	San-	ctum,		
12	34	TB	7	Do-	mi-	num,	et	vi-	vi-	fi-		
	35	SATB	9	can-	tem:	qui	ex	Pa-	tre	Fi-	li-	o-
	36	SA	7	que	pro-	ce-	dit.	Qui	cum	Pa-		
13	37	SA	7	tre	et	Fi-	li-	o	si-	mul		
	38	SATB	9	ad-	o-	ra-	tur,	et	con-	glo-	ri-	fi-
	39	TB	7	ca-	tur,	qui	lo-	cu-	tus	est		
14	40	TB	7	per	Pro-	phe-	tas.	Et	u-	nam		
	41	SATB	9	san-	ctam	ca-	tho-	li-	cam	et	a-	po-
	42	SA	7	sto-	li-	cam	Ec-	cle-	si-	am.		
15	43	SA	7	Con-	fi-	te-	or	u-	num	ba-		
	44	SATB	9	pti-	sma	in	re-	mis-	si-	o-	nem	pec-
	45	TB	7	ca-	to-	rum.	Et	ex-	spe-	cto		
16	46	TB	7	re-	sur-	re-	cti-	o-	nem	mor-		
	47	SATB	9	tu-	o-	rum,	et	vi-	tam	ven-	tu-	ri
	48	SA	4	sae-	cu-	li.	A-					
	49	SATB	1	men								

E2 Approximate patterns

The first plot shows the pattern of repetitions of the melodic voices. All repetitions of the alto (A) and bass (B) are plotted above one another. We manually identified an *approximate pattern* of notes and ornaments (shown in the background) that best matches all of the repetitions. In other words, it minimizes the number of deviations. For the T-voices (next two plots), this turned out to be a crucial step in understanding their construction.



The pattern of repetitions of the tintinnabuli voices is shown in a similar way as for the melodic voices. The patterns are now however twice as long.

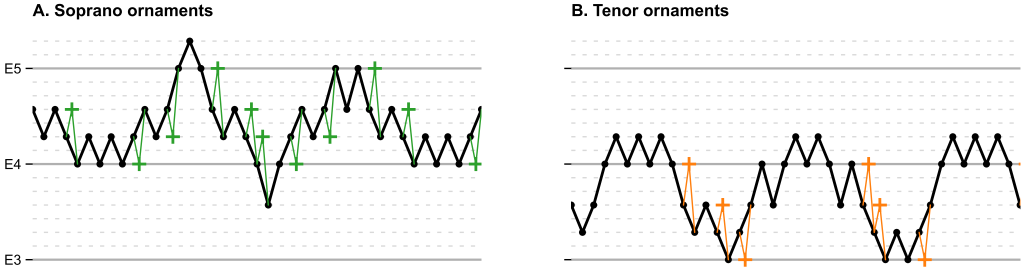


A. Repetitions of the soprano pattern

B. Repetitions of the tenor pattern

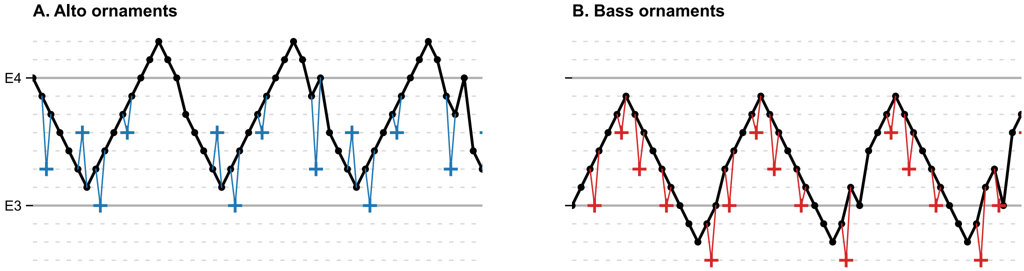
E3 Tenor and soprano ornaments

The RepeatPrevious process generates the ornaments for the soprano **(A)** and tenor **(B)** by repeating the previous note if this is not equal to the next note. As explained in the main text, this process has several parameters that constrain the range of the ornaments. For the soprano the ornaments have to lie between $b = \mathrm{E}_4$ and $B = \mathrm{E}_5$, while the next note has to fall below $C = \mathrm{E}_5$. For the tenor, we use $b = \mathrm{E}_3$, $B = \mathrm{E}_4$ and $C = \mathrm{B}_3$.



E4 Alto and bass ornaments

The TailRotatedPatternProcess process generates the ornaments for the alto **(A)** and bass **(B)** parts. The black lines show the respective melodies, and ornaments are indicated by colored plusses. It essentially repeats a 16-note pattern of ornamentation but rotates the tail every time to keep the ornamentation in sync with the melody (see main text for details).



E5 Implementation: code sample

Fragment of the implementation. Using tintinnabulipy, all notes and ornaments of the alto and soprano can be constructed in just a few lines of code. The tenor and bass are similar. The majority of the remaining code is needed to turn this into an actual score (i.e., a musicxml file).

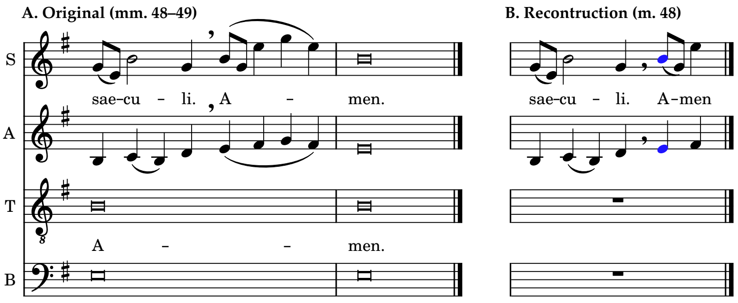
```
# Define the melodic spaces
M = MelodicSpace(MinorScale('E4'))
T = TintinnabuliSpace(Chord(['E4', 'G4', 'B4']))

# Construct the alto melody and ornaments
alto_pattern = glue(M.mode2(6), M.mode4(6), M.mode1(2), M.mode3(2))[:-1]
repetitions = [rotate_tail(alto_pattern, i) for i in range(16)]
alto = concatenate(*repetitions)
ornament_pattern = [None, 'G3', None, None, None, 'B3', None, 'E3',
                    None, None, 'B3', None, None, None, None, None]
alto_orn_process = TailRotatedPatternProcess(T, alto_pattern)
alto_ornaments = alto_process(alto, t0=False)

# Construct the soprano melody and ornaments
soprano = StepProcess(T, position=2)(alto)
sop_orn_process = RepeatPreviousProcess(T, ['E4', 'E5'], [None, 'E5'])
sop_ornaments = sop_orn_process(soprano, t0=soprano[0])
```

E6 Ending of *Summa*

The ending of *Summa* **(A)** is more freely composed than the rest of the piece and deviates from the patterns observed before. The alto finishes the last repetition of the basic pattern in a four-note melisma, to end on the tonic. Meanwhile, the bass and tenor hold an open fifth on 'Amen'. The reconstruction **(B)** of course cannot accurately reproduce these measures. We treat the errors as ornament insertions and count 2 extra insertions for all voices.



Bibliography

Bibliography

A

Abdoli, S. (2011). Iranian traditional music dastgah classification. *Proceedings of the 12th International Conference on Music Information Retrieval*, 275–280.

Adams, C. R. (1976). Melodic contour typology. *Ethnomusicology*, *20*(2), 179. DOI 10/d7p2xx

Adolfsson, A., Ackerman, M., & Brownstein, N. C. (2019). To cluster, or not to cluster: An analysis of clusterability methods. *Pattern Recognition*, *88*, 13–26. DOI 10/ggh2gq

Ahmed, N., Natarajan, T., & Rao, K. R. (1974). Discrete cosine transform. *IEEE Transactions on Computers*, *C-23*(1), 90–93. DOI 10/c4kqx4

Alem, S., Perry, C. J., Zhu, X., Loukola, O. J., Ingraham, T., Søvik, E., & Chittka, L. (2016). Associative mechanisms allow for social learning and cultural transmission of string pulling in an insect. *PLOS Biology*, *14*(10), e1002564. DOI 10/f89jjs

Anagnostopoulou, C., & Buteau, C. (2010). Can computational music analysis be both musical and computational? *Journal of Mathematics and Music*, *4*(2), 75–83. DOI 10/fqqnx6

Antognini, J. M., & Sohl-Dickstein, J. (2018). PCA of high dimensional random walks with comparison to neural network training. *32nd Conference on Neural Information Processing Systems*, 10328–10337.

Atalay, N. B., & Yöre, S. (2020). Pitch distribution, melodic contour or both? Modeling makam schema with multidimensional scaling and self-organizing maps. *New Ideas in Psychology*, *56*, 100746. DOI 10/gf97gs

B

Berten, O. (2013–2020). GregoBase: A database of Gregorian scores. https://gregobase.selapa.net

Bertram, F. (2014). Learning elementary musical programming with extempore: Translating Arvo Pärt's Fratres into live code snippets. *Proceedings of the 7th International Conference of Students of Systematic Musicology (SysMus14)*.

Biró, D. P., van Kranenburg, P., Ness, S., Tzanetakis, G., & Volk, A. (2012). Stability and variation in cadence formulas in oral and semi-oral chant traditions – a computational approach. *Proceedings of the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music*, 98–105.

Bittner, R. M., Salamon, J., Bosch, J. J., & Bello, J. P. (2017). Pitch contours as a mid-level representation for music informatics. *Audio Engineering Society Conference on Semantic Audio*.

Bittner, R. M., Salamon, J., Essid, S., & Bello, J. P. (2015). Melody extraction by contour classification. *Proceedings of the 16th International Conference on Music Information Retrieval*, 500–506.

Böhme, F. M. (1877). *Altdeutsches Liederbuch: Volkslieder der Deutschen nach Wort und Weise aus dem 12. bis zum 17. Jahrhundert*. Breitkopf und Härtel.

Böhme, F. M. (1895). *Volksthümliche Lieder der Deutschen im 18. und 19. Jahrhundert: Nach Wort und Weise aus alten Drucken und Handschriften, sowie aus Volksmund zusammengebracht*. Breitkopf und Härtel.

Bouteneff, P. C., Engelhardt, J., & Saler, R. (Eds.). (2021). *Arvo Pärt: Sounding the sacred* (1st ed.). Fordam University Press.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. DOI 10/d8zjwq

Brinkman, A. (2020). *Exploring the structure of Germanic folksong* (Doctoral dissertation). The Ohio State University. Ohio.

Brinkman, A. (2021). *Analyses on the history, form, and use of the Essen Folksong Collection* (Talk). Spring 2021 Meeting of the South Central Society for Music Theory. Retrieved 03/01/2023, from https://www.scsmt.org/2021_conf_files/brinkman_2021_text.pdf

Brown, S., & Jordania, J. (2011). Universals in the world's musics. *Psychology of Music*, *41*(2), 229–248. DOI 10/bhnxdh

Burchardt, L. S., & Knörnschild, M. (2020). Comparison of methods for rhythm analysis of complex animals' acoustic signals. *PLOS Computational Biology*, *16*(4), e1007755. DOI 10/jx9x

Burchardt, L. S., Picciulin, M., Parmentier, E., & Bolgan, M. (2021). A primer on rhythm quantification for fish sounds: A Mediterranean case study. *Royal Society Open Science*, *8*(9), 210494. DOI 10/jx9t

C

Cavalli-Sforza, L., Menozzi, P., & Piazza, A. (1993). Demic expansions and human evolution. *Science*, *259*(5095), 639–646. DOI 10/fmj4h5

Chordia, P., & Rae, A. (2007). Raag recognition using pitch-class and pitch-class dyad distributions. *Proceedings of the 8th International Society for Music Information Retrieval Conference*, 431–436.

Claidiere, N., Smith, K., Kirby, S., & Fagot, J. (2014). Cultural evolution of systematically structured behaviour in a non-human primate. *Proceedings of the Royal Society B: Biological Sciences*, *281*(1797), 20141541. DOI 10/gfr9gt

Clayton, M., Eerola, T., Tarsitani, S., Jankowsky, R., Jure, L., Poole, A., Rocamora, M., & Jakubowski, K. (2022). Interpersonal entrainment in music performance. DOI 10/jx92

Cleveland, W. S., & McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, *79*(387), 531–554. DOI 10/cbtq7v

Condit-Schultz, N. (2019). Deconstructing the nPVI: A methodological critique of the normalized pairwise variability index as applied to music. *Music Perception*, *36*(3), 300–313. DOI 10/jx93

Cornelissen, B. (2017). *Bayesian language games: Unifying and evaluating agent-based models of horizontal and vertical language evolution* (Master's thesis). University of Amsterdam. Amsterdam. https://eprints.illc.uva.nl/id/eprint/1556

Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2021a). *Fixing Huron's typology* (poster). Locomus Meeting 2021. http://locomus.net/locomus21/abstracts/

Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2021b). Catafolk: Cataloguing folk music datasets for comparative musicology. In J. Stupacher & S. Hagner (Eds.), *Proceedings of the 14th International Conference of Students of Systematic Musicology (SysMus21)*. DOI 10/jx9r

Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2022). Understanding automatic mode classification in Western plainchant. *50th Medieval and Renaissance International Music Conference (MedRen 2022)*.

Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2020a). Mode classification and natural units in plainchant. *Proceedings of the 21st International Conference on Music Information Retrieval*, 869–875. DOI 10/jx94

Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2020b). Studying large plainchant corpora using Chant21. *7th International Conference on Digital Libraries for Musicology*, 40–44. DOI 10/ghmnrk

Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2021c). Cosine contours: A multipurpose representation for melodies. *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 135–142. DOI 10/gpdr3w

Creighton, H. (1932). *Songs and ballads from Nova Scotia*. J.M. Dent & Sons.

Cuthbert, M. S., & Ariza, C. (2010). Music21: A toolkit for computer-aided Musicology and Symbolic Music Data. *Proceedings of the 11th International Conference on Music Information Retrieval*, 637–642.

D

De Gregorio, C., Valente, D., Raimondi, T., Torti, V., Miaretsoa, L., Friard, O., Giacoma, C., Ravignani, A., & Gamba, M. (2021). Categorical rhythms in a singing primate. *Current Biology*, *31*(20), R1379–R1380. DOI 10/gnpm8w

De Paiva Santana, C., & Bresson, J. (2012). *Vers la modélisation des pensées musicales: Le cas du «tintinabuli» d'Arvo Pärt*. Poster presented at the Conférence sur la Modélisation Mathématique et Informatique des Systèmes Complexes (COMMISCO 2012).

Dejanović, I., Milosavljević, G., & Vaderna, R. (2016). Arpeggio: A flexible PEG parser for Python. *Knowledge-Based Systems*, *95*, 71–74. DOI 10/f79cb2

Densmore, F. (1910). *Chippewa music*. Government Printing Office.

Densmore, F. (1913). *Chippewa music*. Government Printing Office.

Densmore, F. (1918). *Teton sioux music*. Government Printing Office.

Densmore, F. (1922). *Northern Ute music*. Government Printing Office.

Densmore, F. (1929a). *Papago music*. Government Printing Office.

Densmore, F. (1929b). *Pawnee music*. Government Printing Office.

Densmore, F. (1932). *Menominee music*. Government Printing Office.

Densmore, F. (1939). *Nootka and Quileute music*. Government Printing Office.

Densmore, F. (1943). *Chocktaw music*. Government Printing Office.

Densmore, F. (1957). *Music of Acoma, Isleta, Cochiti, and Zuñi Pueblos*. Government Printing Office.

Densmore, F. (1958). *Music of the Maidu Indians of California*. The Southwest Museum.

Desain, P., & Honing, H. (2003). The formation of rhythmic categories and metric priming. *Perception*, *32*(3), 341–365. DOI 10/c75gbm

Dowling, W. J. (1978). Scale and contour: Two components of a theory of memory for melodies. *Psychological Review*, *85*(4), 341–354. DOI 10/ckm55j

E

Edwards, M. (2011). Algorithmic composition: Computational thinking in music. *Communications of the ACM*, *54*(7), 58–67. DOI 10/fpbwgm

Erk, L., & Böhme, F. M. (1893a). *Deutscher Liederhort: Auswahl der vorzüglicheren deutschen Volkslieder, nach Wort und Weise aus der Vorzeit und Gegenwart* (Vol. 1). Breitkopf und Härtel.

Erk, L., & Böhme, F. M. (1893b). *Deutscher Liederhort: Auswahl der vorzüglicheren deutschen Volkslieder, nach Wort und Weise aus der Vorzeit und Gegenwart* (Vol. 2). Breitkopf und Härtel.

Erk, L., & Böhme, F. M. (1894). *Deutscher Liederhort: Auswahl der vorzüglicheren deutschen Volkslieder, nach Wort und Weise aus der Vorzeit und Gegenwart* (Vol. 3). Breitkopf und Härtel.

F

Fernandez, A. A., Burchardt, L. S., Nagy, M., & Knörnschild, M. (2021). Babbling in a vocal learning bat resembles human infant babbling. *Science*, *373*(6557), 923–926. DOI 10/gsbc

Filer, A., Burchardt, L. S., & Rensburg, B. J. (2021). Assessing acoustic competition between sibling frog species using rhythm analysis. *Ecology and Evolution*, *11*(13), 8814–8830. DOI 10/gj7wkh

Ford, B. (2004). Parsing expression grammars: A recognition-based syntactic foundation. *POPL '04: Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 111–122. DOI 10/ddkr4k

G

Gerazov, B., & Wagner, M. (2021). ProsoBeast prosody annotation tool. *Proc. Interspeech 2021*, 2621–2625. DOI 10/jzbr

Gray, R. M. (2006). Toeplitz and circulant matrices: A review. *Foundations and Trends in Communications and Information Theory*, *2*(3), 155–239. DOI 10/bxhj7f

Griffiths, T. L., & Kalish, M. L. (2007). Language evolution by iterated learning with bayesian agents. *Cognitive Science*, *31*(3), 441–480. DOI 10/dtbng4

Gulati, S., Serra, J., Ishwar, V., Senturk, S., & Serra, X. (2016). Phrase-based rāga recognition using vector space modeling. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, 66–70. DOI 10/gf97b2

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., & Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1195–1205. DOI 10/ggvzcg

H

Harrison, P. M. C., Marjieh, R., Adolfi, F., van Rijn, P., Anglada-Tort, M., Tchernichovski, O., Larrouy-Maestri, P., & Jacoby, N. (2020). Gibbs sampling with people. *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 0659–10671.

Hartigan, J. A., & Hartigan, P. M. (1985). The dip test of unimodality. *The Annals of Statistics*, *13*(1), 70–84. DOI 10/ctwtms

Haspelmath, M. (2018). How comparative concepts and descriptive linguistic categories are different. In D. Olmen, T. Mortelmans, & F. Brisard (Eds.), *Aspects of Linguistic Variation* (pp. 83–114). De Gruyter. DOI 10/gf2xw2

Haydn, J. (1792). *A selection of original Scots songs (Hob.XXXIa)* (Vol. 2). William Napier.

Helsen, K., Bain, J., Fujinaga, I., Hankinson, A., & Lacoste, D. (2014). Optical music recognition and manuscript chant sources. *Early Music*, *42*(4), 555–558. DOI 10/gfx9sv

Helsen, K., & Lacoste, D. (2011). A report on the encoding of melodic incipits in the Cantus database with the music font 'Volpiano'. *Plainsong and Medieval Music*, *20*(01), 51–65. DOI 10/bptj5d

Heydarian, P., & Bainbridge, D. (2019). Dastgàh recognition in Iranian music: Different features and optimized parameters. *Proceedings of the 6th International Conference on Digital Libraries for Musicology*, 53–57. DOI 10/ggcrd4

Hiley, D. (2009). *Gregorian chant*. Cambridge University Press. DOI 10/jzbt

Hillier, P. (1989). Arvo Pärt: Magister ludi. *The Musical Times*, *130*(1753), 134–137. DOI 10/b6739k

Hillier, P. (1997). *Arvo Pärt*. Oxford University Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. DOI 10/bxd65w

Honing, H. (2018). Musicality as an upbeat to music: Introduction and research agenda. In H. Honing (Ed.), *The Origins of Musicality* (pp. 3–20). MIT Press. DOI 10/jzbw

Honing, H., ten Cate, C., Peretz, I., & Trehub, S. E. (2015). Without it no music: Cognition, biology and evolution of musicality. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *370*(1664), 20140088. DOI 10/gfc4x4

Hoppitt, W., & Laland, K. N. (2013). *Social learning: An introduction to mechanisms, methods and models*. Princeton university press.

Huron, D. (1996). The melodic arch in Western folksongs. *Computing in Musicology*, *10*, 3–23.

Huron, D., & Veltman, J. (2006). A cognitive approach to medieval mode: Evidence for an historical antecedent to the major/minor system. *1*(1), 33–55. DOI 10/ggbgdc

J

Jacoby, N., & McDermott, J. H. (2017). Integer ratio priors on musical rhythm revealed cross-culturally by iterated reproduction. *Current Biology*, (27), 1–12. DOI 10/f9rjct

Jacoby, N., Polak, R., Grahn, J., Cameron, D. J., Lee, K. M., Godoy, R., Undurraga, E. A., Huanca, T., Thalwitzer, T., Doumbia, N., Goldberg, D., Margulis, E. H., Wong, P., Jure, L., Rocamora, M., Fujii, S., Savage, P. E., Ajimi, J., Konno, R., … Mcdermott, J. (2021). *Universality and cross-cultural variation in mental representations of music revealed by global comparison of rhythm priors* (preprint). DOI 10/jzb4

Janssen, B., van Kranenburg, P., & Volk, A. (2017). Finding occurrences of melodic segments in folk songs employing symbolic similarity measures. *Journal of New Music Research*, *46*(2), 118–134. DOI 10/gfn44g

Jeffery, P. (1992). *Re-envisioning past musical cultures: Ethnomusicology in the study of Gregorian chant*. University of Chicago Press.

Johnson, K. P., Burns, P., Stewart, J., & Cook, T. (2014–2021). CLTK: The classical language toolkit. https://github.com/cltk/cltk

Jolliffe, I. (2002). *Principal component analysis* (2nd ed.). Springer.

K

Kalogeratos, A., & Likas, A. (2012). Dip-means: An incremental clustering method for estimating the number of clusters. *Advances in Neural Information Processing Systems 25*, 2393–2401.

Kelkar, T., Roy, U., & Jensenius, A. R. (2018). Evaluating a collection of sound-tracing data of melodic phrases. *19th International Society for Music Information Retrieval Conference*, 74–81.

Keogh, E. J., & Pazzani, M. J. (2001). Derivative Dynamic Time Warping. *Proceedings of the 2001 SIAM International Conference on Data Mining*, 1–11. DOI 10/gfn7sr

Knörnschild, M., Behr, O., & von Helversen, O. (2006). Babbling behavior in the sac-winged bat (Saccopteryx bilineata). *Naturwissenschaften*, *93*(9), 451–454. DOI 10/c3xxnz

Kolinski, M. (1959). The evaluation of tempo. *Ethnomusicology*, *3*(2), 45–57. DOI 10/bnk36q

Kolinski, M. (1965a). Classification of tonal structures: Illustrated by a comparative chart of American Indian, African Negro, Afro-American and English-American structures. *Studies in Ethnomusicology*, *1*, 38–76.

Kolinski, M. (1965b). The general direction of melodic movement. *Ethnomusicology*, *9*(3), 240–264. DOI 10/bjm6x2

Kosak, G. (1994). *Arvo Pärt's Summa: An example of systematic construction* (Master's thesis). Northern Arizona University.

Krämer, R. (2015). *Algorithmic music analysis: A case study of a prelude from David Cope's "From Darkness, Light"* (Doctoral dissertation). University of North Texas.

Kranenburg, P. van, Biro, D. P., Ness, S., & Tzanetakis, G. (2011). A computational investigation of melodic contour stability in Jewish Torah trope performance traditions. *Proceedings of the 12th International Conference on Music Information Retrieval*, 163–168.

Kranenburg, P. van, & Maessen, G. (2017). Comparing offertory melodies of five medieval Christian chant traditions. *Proceedings of the 18th International Conference on Music Information Retrieval*, 163–168.

L

Lacoste, D., Bailey, T., Steiner, R., & Koláček, J. (1987–2019). Cantus: A database for Latin ecclesiastical chant [Directed by Debra Lacoste (2011–), Terence Bailey (1997–2010), and Ruth Steiner (1987–1996). Web developer, Jan Koláček (2011–).]. http://cantus.uwaterloo.ca

Ladd, D. (2001). Intonational universals and intonational typology. In M. Haspelmath, E. König, W. Oesterreicher, & W. Raible (Eds.), *Language Typology and Language Universals: An International Handbook,* (pp. 1380–1390). Mouton de Gruyter.

Levinson, S. C., & Dediu, D. (2013). The interplay of genetic and cultural factors in ongoing language evolution. In P. J. Richerson & M. H. Christiansen (Eds.), *Cultural Evolution: Society, Technology, Language, and Religion* (pp. 219–231). MIT Press.

Lewin, D. (1987). *Generalized musical intervals and transformations*. Oxford University Press.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., & Stoica, I. (2018). Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

Lorch, E. (2016). Visualizing deep network training trajectories with PCA. *ICML Workshop on Visualization for Deep Learning*.

Loukola, O. J., Perry, C. J., Coscos, L., & Chittka, L. (2017). Bumblebees show cognitive flexibility by improving on an observed complex behavior. *Science*, *355*(6327), 24–26. DOI 10/bz98

M

Mampe, B., Friederici, A. D., Christophe, A., & Wermke, K. (2009). Newborns' cry melody is shaped by their native language. *Current Biology*, *19*(23), 1994–1997. DOI 10/dqw8vr

McInnes, L., Healy, J., & Melville, J. (2018). *Umap: Uniform manifold approximation and projection for dimension reduction*. arXiv preprint. DOI 10/gqzqzn

Mehr, S. A., Singh, M., Knox, D., Ketter, D. M., Pickens-Jones, D., Atwood, S., Lucas, C., Jacoby, N., Egner, A. A., Hopkins, E. J., Howard, R. M., Hartshorne, J. K., Jennings, M. V., Simson, J., Bainbridge, C. M., Pinker, S., O'Donnell, T. J., Krasnow, M. M., & Glowacki, L. (2019). Universality and diversity in human song. *Science*, *366*(6468), eaax0868. DOI 10/ggdvjp

Moore, J., Ahmed, H., & Antia, R. (2018). High dimensional random walks can appear low dimensional: Application to influenza H3N2 evolution. *Journal of Theoretical Biology*, *447*, 56–64. DOI 10/gdmbxm

Motte-Haber, H. de la. (1996). Struktur als programm: Analytische bemerkungen zur komposition summa von arvo pärt. In W. Gratzer (Ed.), *Nähe und distanz: Nachgedachte musik der gegenwart* (pp. 14–25). Wolke.

Müllensiefen, D., & Frieler, K. (2004). Cognitive adequacy in the measurement of melodic similarity: Algorithmic vs. human judgments. In *Music query: Methods, models, and user studies* (pp. 147–176). MIT Press.

Müllensiefen, D., & Wiggins, G. A. (2012). Polynomial functions as a representation of melodic phrase contour. In *Systematic musicology: Empirical and theoretical studies*. Peter Lang. DOI 10/jzb9

N

Nettl, B. (2005). *The study of ethnomusicology: Thirty-one issues and concepts* (Vol. 7). University of Illinois Press.

Neubarth, K., Shanahan, D., & Conklin, D. (2018). Supervised descriptive pattern discovery in Native American music. *Journal of New Music Research*, *47*(1), 1–16. DOI 10/gfwb8d

Novembre, J., & Stephens, M. (2008). Interpreting principal component analyses of spatial population genetic variation. *Nature Genetics*, *40*(5), 646–649. DOI 10/bmk33z

Nuttall, T., Casado, M. G., Tarifa, V. N., Repetto, R. C., & Serra, X. (2019). Contributing to new musicological theories with computational methods: The case of centonization in Arab-Andalusian music. *Proceedings of the 20th International Conference on Music Information Retrieval*, 223–228.

O

O'Boyle, S. (1976). *The Irish song tradition*. Gilbert Dalton.

O'Sullivan, D. (1981). *Songs of the Irish: An anthology of Irish folk music and poetry with English verse translations*. Mercier Press.

P

Panteli, M., Bittner, R., Bello, J. P., & Dixon, S. (2017). Towards the characterization of singing styles in world music. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 636–640. DOI 10/gf85jt

Panteli, M., & Purwins, H. (2013). A computational comparison of theory and practice of scale intonation in byzantine chant. *Proceedings of the 14th International Conference on Music Information Retrieval*, 169–174.

Parsons, D. (1975). *Directory of tunes and musical themes*. Spencer Brown.

Pärt, A. (1996). Zu Summa. In W. Gratzer (Ed.), *Nähe und Distanz: Nachgedachte Musik der Gegenwart* (p. 13). Wolke.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Patrick, D. (2011). *Analysis of Summa by Arvo Pärt* (Master's thesis).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Perricone, J. (2018). *Great songwriting techniques*. Oxford University Press.

Pinck, L. (1926). *Verklingende Weisen: Lothringer Volkslieder* (Vol. 1). Lothringer Verlags- und Hilfsverein.

Pinck, L. (1928). *Verklingende Weisen: Lothringer Volkslieder* (Vol. 2). Lothringer Verlags- und Hilfsverein.

Pinck, L. (1933). *Verklingende Weisen: Lothringer Volkslieder* (Vol. 3). Lothringer Verlags- und Hilfsverein.

Pinck, L. (1939). *Verklingende Weisen: Lothringer Volksliede* (Vol. 4). Lothringer Verlags- und Hilfsverein.

Piston, W. (1970). *Counterpoint* (6th ed.). Victor Gollancz.

Powers, H. S., Wiering, F., Porter, J., Cowdery, J., Widdess, R., Davis, R., Perlman, M., Jones, S., & Marett, A. (2001). Mode. In *Grove music online*. Oxford University Press. DOI 10/jzcb

R

Rao, K. R., & Yip, P. C. (1990). *Discrete cosine transform: Algorithms, advantages, applications*. Academic Press.

Ravignani, A., Delgado, T., & Kirby, S. (2016). Musical evolution in the lab exhibits rhythmic universals. *Nature Human Behaviour*, *1*(1), 0007. DOI 10/gfr9gz

Ray, W. D., & Driver, R. M. (1970). Further decomposition of the Karhunen-Loève Series Representation of a stationarv random process. *IEEE Transactions on Information Theory*, *16*(6), 663–668.

Riester, J. (1978). *Canción y producción en la vida de un pueblo indígena: Los Chimane del oriente Boliviano*. Los Amigos del Libro.

Roeder, J. (2011). Transformational aspects of Arvo Pärt's tintinnabuli music. *Journal of Music Theory*, *55*(1), 1–41. DOI 10/cmqpj5

Roeske, T. C., Tchernichovski, O., Poeppel, D., & Jacoby, N. (2020). Categorical rhythms are shared between songbirds and humans. *Current Biology*, P3544–3555.e6. DOI 10/ghhxnf

Rosenzweig, S., Scherbaum, F., Shugliashvili, D., Arifi-Müller, V., & Müller, M. (2020). Erkomaishvili Dataset: A curated corpus of traditional Georgian vocal music for computational musicology. *Transactions of the International Society for Music Information Retrieval*, *3*(1), 31–41. DOI 10/jzcd

Ruthmann, A., Heines, J. M., Greher, G. R., Laidler, P., & Saulters, C. (2010). Teaching computational thinking through musical live coding in Scratch. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE '10)*, 351. DOI 10/fkrb9j

S

Salamon, J., Rocha, B., & Gomez, E. (2012). Musical genre classification using melody features extracted from polyphonic music signals. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 81–84. DOI 10/gmbqdx

Savage, P. E. (2019). Cultural evolution of music. *Palgrave Communications*, *5*(1), 16. DOI 10/gf7q7c

Savage, P. E. (2022). An overview of cross-cultural music corpus studies. In D. Shanahan, J. A. Burgoyne, & I. Quinn (Eds.), *The Oxford handbook of music and corpus studies*. Oxford University Press. DOI 10/jzcf

Savage, P. E., & Brown, S. (2013). Toward a new comparative musicology. *Analytical Approaches To World Music*, 23.

Savage, P. E., Brown, S., Sakai, E., & Currie, T. E. (2015). Statistical universals reveal the structures and functions of human music. *Proceedings of the National Academy of Sciences*, *112*(29), 8987–8992. DOI 10/f7j74k

Savage, P. E., Merritt, E., Rzeszutek, T., & Brown, S. (2012). CantoCore: A new cross-cultural song classification scheme. *Analytical Approaches to World Music*, *2*, 87–137.

Savage, P. E., Tierney, A. T., & Patel, A. D. (2017). Global music recordings support the motor constraint hypothesis for human and avian song contour. *Music Perception: An Interdisciplinary Journal*, *34*(3), 327–334. DOI 10/ggmd4f

Schaffrath, H. (1995). *The Essen Folksong Collection in the humdrum kern format* (tech. rep.). Center for Computer Assisted Research in the Humanities.

Schmuckler, M. A. (1999). Testing models of melodic contour similarity. *Music Perception*, *16*(3), 295–326. DOI 10/gjmhg6

Schmuckler, M. A. (2016). Tonality and contour in melodic processing. In S. Hallam, I. Cross, & M. Thaut (Eds.), *The Oxford handbook of music psychology* (2nd ed.). Oxford University Press. DOI 10/jzcg

Shanahan, D., & Shanahan, E. (2014). The Densmore Collection of Native American songs: A new corpus for studies of effects of geograpy, language and social function on folk song. *Proceedings for the 13th International Conference for Music Perception and Cognition*, 206–208.

Shaw, R. (2018). Differentiae in the Cantus manuscript Database: Standardization and musicological application. *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, 38–46. DOI 10/gg572p

Shenton, A. (Ed.). (2012). *The Cambridge companion to Arvo Pärt*. Cambridge University Press.

Shvets, A. (2014). Mathematical bases of the form construction in Arvo Pärt's music. *Lietuvos muzikologija*, *15*, 88–101.

Shvets, A., & De Paiva Santana, C. (2014). Modelling Arvo Pärt's music with OpenMusic. *Electronic Visualisation and the Arts*, 9–16. DOI 10/gn7nqc

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., & Hirschberg, J. (1992). TOBI: A standard for labeling English prosody. *2nd International Conference on Spoken Language Processing*, 867–870. DOI 10/jzch

Small, C. (1998). *Musicking: The meanings of performing and listening*. Wesleyan University Press.

Spike, M. (2020). Fifty shades of grue: Indeterminate categories and induction in and out of the language sciences. *Linguistic Typology*, *24*(3), 465–488. DOI 10/gh5shf

Steinbeck, W. (1982). *Struktur und Ähnlichkeit: Methoden automatisierter Melodieanalyse*. Bärenreiter.

Strang, G. (1999). The discrete cosine transform. *SIAM Review*, *41*(1), 135–147. DOI 10/dxf5vm

T

Tierney, A. T., Russo, F. A., & Patel, A. D. (2011). The motor origins of human and avian song structure. *Proceedings of the National Academy of Sciences*, *108*(37), 15510–15515. DOI 10/b65mxj

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

U

Ünal, E., Bozkurt, B., & Karaosmanoğlu, M. K. (2012). N-gram based statistical makam detection on makam music in Turkey using symbolic data. *Proceedings of the 13th International Conference on Music Information Retrieval*, 43–48.

V

Velarde, G., Meredith, D., & Weyde, T. (2016). A wavelet-based approach to pattern discovery in melodies. In D. Meredith (Ed.), *Computational Music Analysis* (pp. 303–333). Springer International Publishing. DOI 10/jzcm

Veldhoen, S., Hupkes, D., & Zuidema, W. (2016). Diagnostic classifiers: Revealing how neural networks process hierarchical structure. *Pre-Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*.

Voigt, C. C., Caspers, B., & Speck, S. (2005). Bats, bacteria and bat smell: Sex-specific diversity of microbes in a sexually selected scent organ. *Journal of Mammalogy*, *86*(4), 745–749. DOI 10/fdpjd5

Volk, A., & van Kranenburg, P. (2012). Melodic similarity among folk songs: An annotation study on similarity-based categorization in music. *Musicæ Scientiæ*, *16*(3), 317–339. DOI 10/gfjz62

W

Wermke, K., Robb, M. P., & Schluter, P. J. (2021). Melody complexity of infants' cry and non-cry vocalisations increases across the first six months. *Scientific Reports*, *11*(1), 4137. DOI 10/gh6hvb

Whitehead, H., & Rendell, L. (2015). *The cultural lives of whales and dolphins*. University of Chicago Press.

Wiering, F. (2006). Comment on Huron and Veltman: Does a cognitive approach to medieval mode make sense? *Empirical Musicology Review*, *1*(1), 56–60. DOI 10/ggbgd2

Winkler, I., Haden, G. P., Ladinig, O., Sziller, I., & Honing, H. (2009). Newborn infants detect the beat in music. *Proceedings of the National Academy of Sciences*, *106*(7), 2468–2471. DOI 10/dj7wsb

X

Xu, J., & Griffiths, T. L. (2010). A rational analysis of the effects of memory biases on serial reproduction. *Cognitive Psychology*, *60*(2), 107–126. DOI 10/fdtjtb

Z

Zellers, M., Gubian, M., & Post, B. (2010). Redescribing intonational categories with functional data analysis. *11th Annual Conference of the International Speech Communication Association*. DOI 10/jzcn

Appendices

Appendices

List of publications

Some of the chapters in this dissertation are directly based on published work. Full references and author contributions are listed at the end of each chapter. The main publications are also summarized here, together with the author contributions, where BC is short for Bas Cornelissen, JAB for John Ashley Burgoyne, and WZ for Willem Zuidema.

CHAPTER 2 This chapter is directly based on Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2020b). Studying large plainchant corpora using Chant21. *7th International Conference on Digital Libraries for Musicology*, 40–44. DOI 10/ghmnrk. BC designed and conducted the research and wrote the paper. JAB and WZ supervised the research and edited the manuscript.

CHAPTER 4 This chapter is primarily based on Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2020a). Mode classification and natural units in plainchant. *Proceedings of the 21st International Conference on Music Information Retrieval*, 869–875. DOI 10/jx94. BC designed and conducted the research and drafted the original conference paper. JAB and WZ supervised the research and edited the manuscript.

CHAPTER 6 This chapter is directly based on Cornelissen, B., Zuidema, W., & Burgoyne, J. A. (2021c). Cosine contours: A multipurpose representation for melodies. *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 135–142. DOI 10/gpdr3w. BC designed and conducted the research and drafted the original conference paper. JAB and WZ supervised the research and edited the manuscript.

Summary

Measuring musics
Notes on modes, motifs, and melodies

Humans are a musical species: we sing, dance, play, or listen, no matter where we are from. To understand why, musicologists have long studied the rich diversity of musical traditions—or *musics*—found around the world. One can, for example, compare musics to identify properties that many musical traditions share or properties that very few share. But such questions require you to somehow *measure* the properties of interest. And that idea motivates this dissertation: can we develop computational methods to *measure musics*, so that we might compare them? A series of studies, interspersed with lighter interludes, discusses ways to measure *modes* in plainchant, inventories of melodic and rhythmic *motifs*, and the shapes of *melodies*, ending with an intricate rarity: music by Arvo Pärt.

This dissertation primarily analyzes sheet music from a range of musical traditions. In the Catafolk project, we collect a sizeable cross-cultural corpus by bundling several existing corpora, mainly containing German, Chinese, and Native American songs. We also present two corpora of Western plainchant (Cantus Corpus and GregoBase Corpus) and a Python package to parse the plainchant formats. This leads to a series of studies of plainchant. We confirm the melodic arch hypothesis—that phrases tend to be arch-shaped—in plainchant, analyze the predictability of a particular musical connection, and train a small recurrent neural language model to compose new chant artificially.

The centerpiece, however, is a study in which we measure the main organizational structure of plainchant: the eight modes. Modes are melody *types* that lie somewhere between abstract scales and concrete melodies. We compare three ways to classify musical mode: two approaches that largely view mode as a scale and one distributional approach that focuses on its melodic character. We find that this latter approach can still determine mode fairly accurately even when all pitch information has been discarded. However, this only really works when the mode is segmented in the 'right' way: in units corresponding to textual units such as syllables and words.

The smaller units into which music can be decomposed, here called *motifs*, form the second thread in this dissertation. In the case of plainchant melodies, variable-length motifs corresponding to textual units proved fruitful, but fixed-length motifs can also be helpful when studying rhythmic data. We show how plotting all motifs of three successive temporal intervals in a so-called *rhythm triangle* effectively highlights rhythmical structures in music and animal vocalizations. It motivates a novel measure of *isochronicity*—how steady, pulse-like a rhythm is—that generalizes a more commonly used measure (the nPVI). Extending these ideas to melodies, we propose to visualize motifs of three successive notes (or two intervals) in what might be called a *melody square* to help identify common and rare melodic patterns.

The third thread in this dissertation concerns the shapes of melodies. How can one best represent—measure, if you like—melodic contour? It turns out that one can efficiently describe variability in contour shapes using cosine functions as they closely approximate the principal components of melodies. This leads to a new contour representation, *cosine contours*, effectively representing the melodic shape using a discrete cosine transform. Cosine contours give a *continuous* description of contour, while most previous work describes shapes in a discrete fashion, using

a fixed set of contour types. We ask if such discrete typologies can accurately describe the variability in contour shape. Rephrasing this as a clustering problem, we propose a way to measure the presence of statistical modes—but find none. This suggests that melodic phrase contours do not cluster into separate types and that discrete typologies may not provide the most appropriate description of melodic contour.

This dissertation ends with a somewhat dissonant finale. Whereas earlier chapters are distant readings of large music collections, the final chapter is a close reading of a single piece: a rarity. Instead of analyzing 'informal' music by formal means, we now use formal means to understand the 'formal' music of Arvo Pärt. His music is well known to be constructed according to precise mathematical rules, and we attempt to reconstruct the full score of his piece *Summa* using formal procedures. This formalization makes the constructions that possibly underlie the composition completely transparent. It also highlights the vast range of musical diversity, from a formal composition to a simple folk song. To our understanding that diversity, this dissertation makes only modest contributions. But, if this dissertation inspires new research or new music, its hopes have been fulfilled.

Samenvatting

Muziek meten
Over modi, motieven en melodieën

De mens is een muzikale soort: we zingen, dansen, spelen of luisteren, ongeacht waar we vandaan komen. Om te begrijpen waarom dat zo is, bestuderen muziekwetenschappers de grote rijkdom aan muziektradities die je over de hele wereld kunt vinden. Door muziektradities te vergelijken, kun je bijvoorbeeld proberen te achterhalen welke eigenschappen in veel tradities voorkomen, of welke eigenschappen juist heel zeldzaam zijn. Maar om dat te doen, moet je die eigenschappen wel op een of andere manier kunnen *meten*. En dat is de motivatie achter dit proefschrift: kunnen we computationele methoden ontwikkelen om muziektradities te meten en ze zo te kunnen vergelijken? In een reeks studies, afgewisseld met lichtere interludes, worden manieren besproken om *modi* in gezangen, melodische en ritmische *motieven*, en de vormen van *melodieën* te meten, om af te sluiten met een complexe zeldzaamheid: de muziek van Arvo Pärt.

In dit proefschrift analyseren we voornamelijk bladmuziek, uit een aantal verschillende tradities. In het *Catafolk*-project bundelen we bestaande corpora, met voornamelijk Duitse, Chinese en inheems Noord-Amerikaanse muziek, tot een crosscultureel corpus. We presenteren ook twee corpora met Westerse kerkgezangen (Cantus Corpus en GregoBase Corpus), samen met Python-software om de muziek uit te kunnen lezen. Deze corpora gebruiken we in een aantal studies naar kerkgezangen. We bevestigen bijvoorbeeld de bekende hypothese dat de melodieën van frases doorgaans boogvormig zijn, analyseren de regelmatigheid van een specifieke muzikale overgang en trainen een klein, recurrent neuraal taalmodel om nieuwe, kunstmatige gezangen te componeren.

Het middelpunt is echter een studie naar de centrale organisatiestructuur van kerkgezangen: de acht modi. Modi zijn melodietypen die het midden houden tussen abstracte toonladders en concrete melodieën. We vergelijken verschillende manieren om de modus van een gezang te bepalen: twee benaderingen die modus grotendeels als toonladder beschouwen, en een meer gedistribueerde benadering die het melodische karakter benadrukt. Die laatste benadering maakt het zelfs mogelijk om met redelijke nauwkeurigheid de modus van een gezang te bepalen, vrijwel zonder toonhoogte-informatie te gebruiken. Het lijkt dan wel belangrijk te zijn om de melodie op de juiste manier te verdelen in eenheden die overeenkomen met tekstuele eenheden als lettergrepen en woorden.

De kleinere eenheden waarin muziek uiteenvalt, die we hier motieven noemen, zijn een tweede thema in dit proefschrift. In het geval van gezangen bleken motieven van variable lengte behulpzaam om modus te bepalen, maar motieven met vaste lengte kunnen nuttig zijn om ritmische data te bestuderen. In het geval van gezangen blijken motieven van variable lengte behulpzaam voor modusbepaling, maar motieven met vaste lengte kunnen nuttig zijn om ritmische data te bestuderen. We laten zien hoe je ritmische structuren in zowel muziek als dierengeluiden effectief kunt visualiseren in een *ritmedriehoek*. Zo'n driehoek laat alle ritmische motieven zien, die uit drie opeenvolgende tijdsintervallen bestaan. Deze visualisatie brengt ons bij een nieuwe maat voor *isochroniteit*—hoe gelijkmatig, puls-achtig een ritme is—die bovendien een generalisatie is van een gangbare maat (nPVI). We breiden deze ideeën ook uit naar melodieën en laten zien hoe motieven van drie opeenvolgende noten (twee intervallen) in een *melodieënvierkant* kunnen worden weergegeven om zo veelvoorkomende en zeldzame motieven uit te lichten.

De vormen of contouren van melodieën zijn het derde thema in dit proefschrift. Hoe kun je de contour van een melodie het beste meten? Blijkbaar kun je de variatie in melodische contouren efficiënt beschrijven met behulp van cosinussen, omdat die de principale componenten van een verzameling melodieën goed lijken te benaderen. We stellen daarom een nieuwe contourrepresentatie voor, de *cosinuscontour*, die de vorm van een melodie in wezen beschrijft aan de hand van een discrete cosinustransformatie. Cosinuscontouren geven een *continue* beschrijving van de vorm van melodieën, terwijl eerdere studies de vormen juist aan de hand van discrete typen beschrijven: stijgend, dalend, boogvormig, enzovoorts. Geeft zo'n discrete typologie een goede beschrijving van de variatie in melodievormen? We vertalen dit naar een clusteringprobleem en stellen een methode voor om de aanwezigheid van statistische modi te testen—maar vinden er geen. Dit suggereert dat melodievormen niet in verschillende typen uiteenvallen en dat een discrete typologie daarom misschien niet de beste beschrijving van melodische contour geeft.

Dit proefschrift eindigt met een dissonante finale. Waar in eerdere hoofdstukken door een verrekijker naar grote collecties muziek werd gekeken, wordt in het laatste hoofdstuk juist één werk onder de loep genomen. En in plaats van 'informele' muziek met formele methoden te benaderen, gebruiken we nu formele methoden om de 'formele' muziek van Arvo Pärt te bestuderen: composities waar vaak precieze, wiskundige patronen aan ten grondslag liggen. We proberen daarom om de volledige partituur van het werk *Summa* te reconstrueren met behulp van formele procedures. Zo'n formalisering legt de mogelijke constructie bloot waar de compositie omheen is gebouwd. Het illustreert ook weer de reikwijdte van muziek: van formele composities tot eenvoudige deuntjes. Dit proefschrift draagt maar een klein steentje bij aan het begrip van die muzikale diversiteit, maar hopelijk prikkelt het voldoende om nieuw onderzoek te inspireren—of nieuwe muziek te laten klinken.