Calculating Musical Mode in MIRtoolbox

Psych 713: Supervised by Michael Schutz

Konrad Swierczek SN001423065

13/10/22

Proposal 04/05/22

- Start Date May 9 - End Date June 20
- 6-10 Hours/Week *(see PNB Graduate Handbook guidelines)*

Modality, a structural feature of music, is an important cue which conveys the emotional character of a piece of music. In western music, the major and minor modes of the diatonic set are most commonly associated with positive and negative emotional valence respectively. Despite many music-theoretical and psychological frameworks for predicting the mode of a piece of music as represented in an audio file, definitions and operationalizations of mode vary. The proposed module will focus on using audio analysis tools (MIRtoolbox) to investigate mode. MIRtoolbox contains a "mirmode" function which returns a value between -1 (high confidence minor mode prediction) and 1 (high confidence major mode prediction). While MIRtoolbox has been used widely by researchers (cited over 1200 times), the effectiveness and accuracy of automated audio feature extraction tools like MIRtoolbox including its "Harmonic Pitch Class Profile" extraction module and mode extraction module based on the Krumhansl/Gomez key-finding algorithm should be explored further. Using a collection of piano preludes recorded by multiple artists, as well as comparisons to other lower-level audio features, the accuracy and consistency of the "Harmonic Pitch Class Profile" and "mirmode" modules can be tested.

Learning Objectives:

- Extracting musical/sound features from audio files
- Understanding of principles/mechanisms behind these musical/sound features
- Develop skills in MATLAB, primarily with MIRtoolbox

Research Objectives:

- Evaluate effectiveness of automated mode detection for audio files
- Evaluate consistency of automated mode detection for different recordings of the same piece of music
- Compare effectiveness of automated mode detection for audio files with automated mode detection for symbolic notation
- Investigate possible structural and psychological underpinnings of mode as discussed in literature cited in MIRtoolbox and other sources

Deliverable Goals:

- Visualizations of mode score variability between interpretations for each piece
- Visualizations of results (i.e., Harpsichord vs. Piano, Audio vs. Symbolic Notation)
- Write-up of methods

Link to Evaluation Form

Introduction

In music, mode refers to the type and order of a scale, or set of pitches, in a piece of music. In contemporary western music, the major and minor modes of the diatonic set are used most typically. Each mode has its own set of compositional norms and affective and associative cues. Mode unfolds over time, can change during a piece of music, is specific to the larger context of a piece, but is also a cultural and learned phenomenon. Composers of western classical music have traditionally labelled compositions with a nominal mode, that is the mode determined by the composer or editor/publisher. By contrast, since mode is a culturally informed phenomenon, the perceived mode of a piece of music might differ from the nominal and may differ between individuals. Despite a great deal of discourse on mode among musicians, musicologists, philosophers, and scientists, a operational definition and method for identifying either the perceived or nominal mode has not received widespread agreement.

Computational models of mode, and computational models of tonality more broadly have received considerable attention since the 1970's. A consistently computed measure of mode can be useful for the creation of stimuli in music cognition experiments, while observed links between the mode of a piece of music and its emotional appraisal have spurred interest in the use of computed mode in content-based music recommendation systems. Recent advances in the psychological study of structural music features such as mode, as well as the availability of sophisticated statistical and computation techniques has resulted in the creation and utilization of numerous tools for extracting features (whether they be acoustic, surface, or structural) from digital representations of music. In particular, interest in extracting acoustic, surface, and structural sound/musical features from digital audio files has grown in recent years due to the ecological validity of analyzing the same signal that listeners experience. However, due to limitations in audio analysis methods, many tools still favor symbolic representations of music such as scores, MIDI (Musical Instruments Digital Interface), expert annotations, and others due to their simplicity and absence of noise. Audio analysis remains more appealing despite the limitations since symbolic notation is often less representative and does not provide crucial cues such as timbre or fine-grained dynamics.

Despite the widespread use of these algorithms in music research, little focus has been devoted to determining if these algorithms are consistent with our music theoretical, psychological, or phenomenological understanding of experiences such as mode.

An immerging trend in MIR (music information retrieval) uses extracted features to perform music version identification, predominantly to identify "covers", or alternate versions of an original piece of music. While in contemporary music a version of a piece of music may vary across many musical dimensions (rhythm, tempo, harmony, etc.) while maintaining others (melody, text, form, etc.), other styles of music are less variable across versions. The western classical music tradition has produced a large body of versions with limited variability in the structural features: in the case of classical solo piano, timbre, tempo, and dynamics might vary significantly, but the pitch and rhythm are maintained more consistently and correspond closely to the score (symbolic notation). Since the pitch content of a piece of music defines its mode, different versions of the same piece without large variation in pitch content should not have differing computed structural pitch features such as mode. Here we use multiple versions of the same set of classical piano pieces to evaluate the consistency of a computational model of mode as well as its underlying components.

MIRtoolbox mirmode

MIRtoolbox is a collection of MATLAB tools developed for extracting a range of musical features from audio files (Lartillot & Toiviainen (2007), Lartillot (2021)). Mirmode is a module of MIRtoolbox intended to extract a quantification of the perceived musical mode in a given audio file. MIRtoolbox accepts digital audio waveforms in a variety of common file formats (.wav, .mp3, etc.). Prior to analysis, the algorithm is capable of segmenting the waveform into time windows for a unfolding analysis, and is also capable of focusing on specific spectral bands of the audio spectrum. Neither of these features are used in this analysis, however an unfolding analysis of mode may be helpful for analyzing larger pieces of music which include modulation, the changing of mode or key, while frequency-dependant analyses might control for some of the compression artifacts associated with commercial audio recordings or unwanted spectral bands (for instance, high frequency regions associated with non-pitched sounds such a cymbals). Following this, a Discrete Fourier Transform is performed on the waveform, which is then processed into a chromagram, or Harmonic Pitch Class Profile which represents the octave-generalized energy of each pitch class in the Twelve Tone Equal Temperament scale tuned to 440Hz (alternative tuning systems can be accommodated). Next, a modified version of the Krumhansl-Schmuckler keyfinding algorithm is performed on the chromagram, returning key-coefficients for all 24 major and minor keys (Gómez (2006)). By default, the highest minor-key correlation is subtracted from the highest major-key correlation to return a value which can only lie between -1 and 1. Alternatively, mirmode can be computed by subtracting the sum of all minor-key correlations with all major-key correlations (discussed below). Figure 1 is a visual schematic of the mirmode algorithm.

Methods

In order to test the effectiveness of the mirmode algorithm, the first eight measures of 14 commercial audio recordings (versions) of all 24 preludes (pieces) from J.S. Bach's Well Tempered Clavier Book 1 by notable performers were analyzed with mirmode. These performances were recorded from 1934 to 2015 and include both harpsichord and piano interpretations (see Table 1) for metadata). A subset of these are included in Palmer's analysis of The Well Tempered Clavier (Bach & Palmer (2004)). In addition, MIDI representations of the preludes were pro-

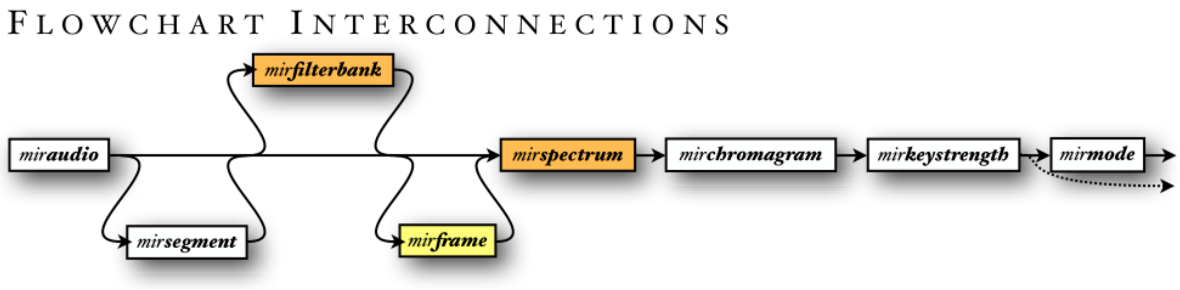

Figure 1: MIRtoolbox Schematic of mirmode from MIRtoolbox Manual 1.8.1

cessed through a Python program utilizing the music21 module Cuthbert & Ariza (2010) replicating MIRtoolbox's mirmode using a pitch class distribution in place of the chromagram. An additional audio version based upon the MIDI representations synthesized with a piano timbre was analyzed. All the audio recordings were .wav files with a sampling rate of 44.1kHz and bit-depth of 16 bits. Both the default method for mirmode as well as the sum option were analyzed. As an additional point of comparison, the MIRtoolbox mirevents algorithm was computed for each audio file as well as the number of events in the MIDI files, including events per second for both audio and MIDI.

Artist	Year Recorded	Year Released	Instrument
Edwin Fischer	1934	2007	Piano
Wanda Landowska	1951	1987	Harpsichord
Rosalyn Tureck	1953	1999	Piano
Jorg Demus	1956	1992	Piano
Martin Galling	1964	2006	Harpsichord
Glenn Gould	1965	1965	Piano
Friedrich Gulda	1972	1995	Piano
Sviatoslav Richter	1972	1992	Piano
Gustav Leonhardt	1973	1989	Harpsichord
Anthony Newman	1973	1983	Harpsichord
Joao Carlos Martins	1981	1994	Piano
Anthony Newman	2000	2001	Harpsichord
Anthony Newman	2001	2001	Piano
Pietro De Maria	2015	2015	Piano

Table 1: Metadata of 13 commercial audio recordings included in analysis.

Results

mirmode Overall Accuracy

To evaluate if the algorithm accurately determines the nominal mode, all calculated mirmode values above 0 were evaluated as computed major, while all values below zero were evaluated as computed minor. Table 2 shows the computed mode consistency for both major and minor nominal modes. While over 75% of nominally major piece versions were identified consistently with the nominal mode by the algorithm, less than half of the nominally minor piece versions were identified consistently. While previous research on the Krumhansl-schmuckler key finding algorithm performed on symbolic notation has shown a high degree of accuracy in identifying the nominal key Krumhansl (2001), it appears that the algorithm is less robust when performed on chromagrams from audio files.

	Nominal Major	Nominal Minor
Computed Major	76.786	56.548
Computed Minor	23.214	43.452

Table 2: Calculated mode of MIDI and Synthesize Audio preludes compared to the nominal mode.

However, the nominal mode of a piece of music is only a reflection of the composers (or in some cases, the editor/publishers) intentions and does not necessarily reflect the perceived mode: particularly in the case of music that was composed in a different cultural context than that in which the mode algorithm was developed and informed by human behaviour. However, while the nominal mode for a given piece of music might be inconsistent with the perceived or computed mode (as appears to be the case for a few pieces in this corpus, for instance 5 Major in Figure 2), the algorithm should perform relatively consistently across multiple versions of the same piece. Figure 2 further divides the analysis into individual pieces, where no single piece had all its versions identified consistently, while one third of the pieces had less than half of their versions identified consistently. There does not appear to be a pattern based on instrument or year of recording. These results suggest that the computed mode, which should generally not vary in different versions of the same composition, as well as the chromagram that underlies the algorithm, seem to be susceptible to factors that do vary between versions: for instance, variability between individual performers, recording equipment, instruments, and room acoustics.

In addition to considering the differences across pieces, it is possible that an individual version may simply be less conducive to MIR-style analyses than others. In particular, older recordings tend to have larger noise floors, poorer dynamic range, and un-even spectral balance

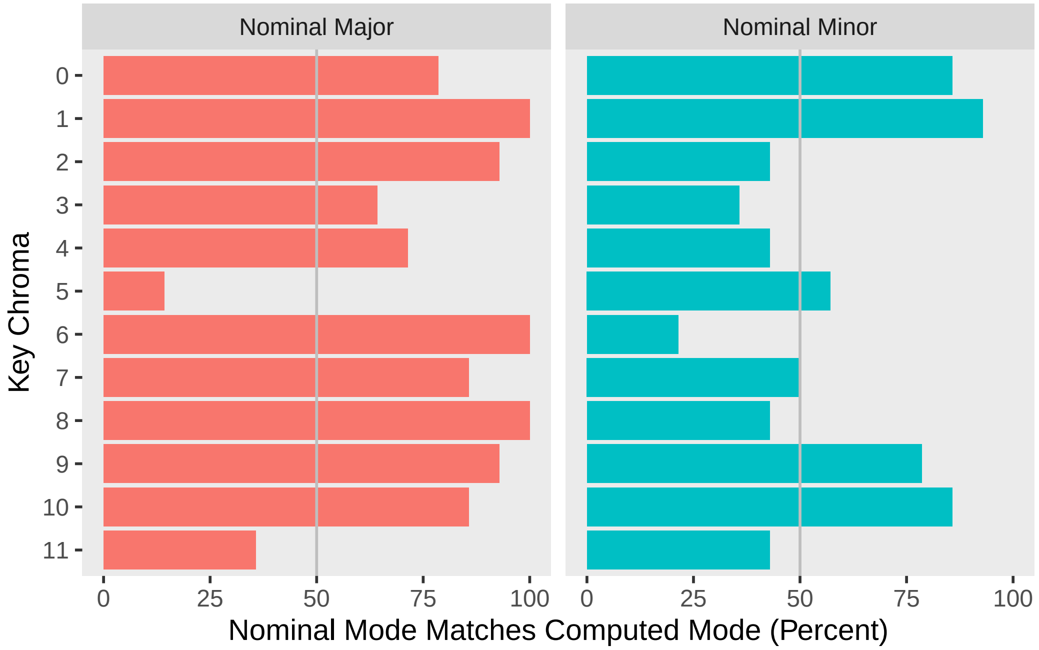

Figure 2: Computed mode (Percentage of audio recordings within a prelude) consistent with the nominal mode.

in comparison to newer recordings. Figure 3 shows the percentage of pieces within a version that match the nominal mode. On average, only 67% of pieces within a version are consistent with the nominal mode. While some albums appear to have more mismatches than others, the majority have more than 40% matching. In addition, the older recordings do not appear to have a greater degree of mismatch.

Symbolic Notation (MIDI) and Synthesized Recording

A shortcoming of the mirmode algorithm is the accuracy of the chromagram used to compute the mode value. Since the chromagram is more prone to error than a pitch class distribution from symbolic notation due to the temporal precision of FFT, noise, spectral variation, etc., calculating the mirmode value based on symbolic notation should be more accurate than audio. Synthesized audio is similarly not subject to the same noise (i.e., dynamic compression, tonal equalization, digital compression, acoustic and electrical noise, timbre variation, etc) associated with commercial recordings and should also be more accurate than commercial recordings, while less accurate than MIDI due to still requiring the generation of a chromagram from a Fourier transform. Table 3 shows the amount of preludes with a calculated mode consistent with the nominal mode for both the MIDI representation and the synthesized audio. While the algorithm performs similarly for nominally major MIDI files, synthesized audio,

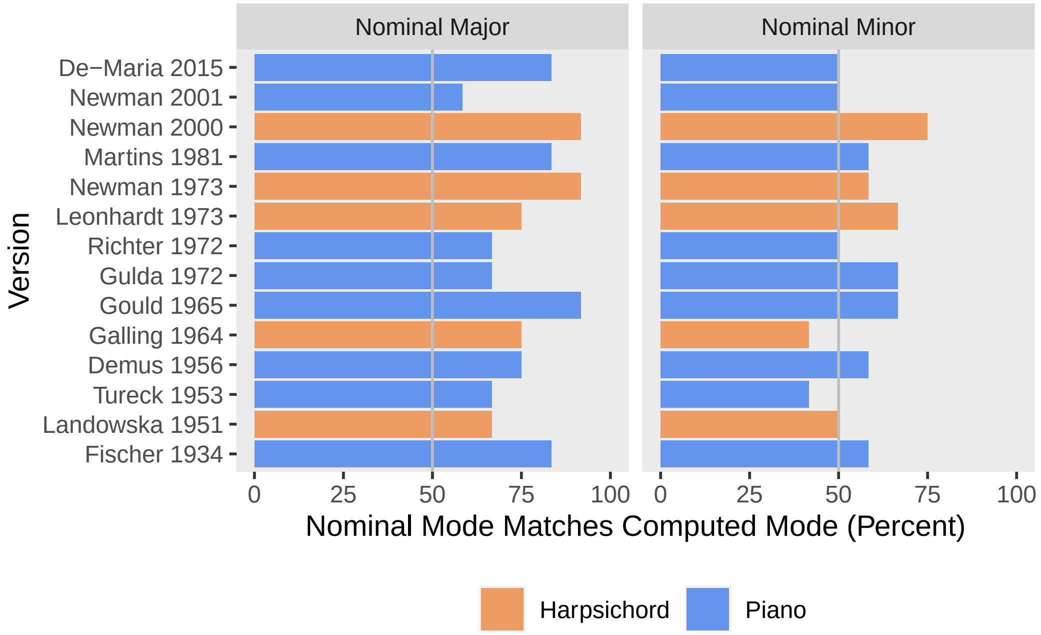

Figure 3: Computed mode (Percentage of audio recordings within a version) consistent with the audio mode. Colours indicate the instrument of the version.

and commercial audio, nominally minor MIDI files were more consistently computed as minor than synthesized audio recordings. However, in both cases, the range of values are limited to -0.3 - 0.3 calling into question the relative confidence of any given mode determination, regardless of the stimulus format. While the computed mode for MIDI representations appears to be quite consistent with the nominal mode, The introduction of a chromagram computation consistently leads to more major mode predictions by the mirmode algorithm. Figure 4 more specifically outlines the discrepancy between the MIDI files and synthesized audio. While in 16 pieces the difference between the two versions was negligible in terms of the predicted mode, eight pieces saw the mode prediction change, with variability present in many of the pieces that did not change mode according to the criteria used here.

	MIDI		Synthesized Audio	
	Major$_{nom}$	Minor$_{nom}$	Major$_{nom}$	Minor$_{nom}$
Major$_{calc}$	9	1	8	6
Minor$_{calc}$	3	11	4	6

Table 3: Calculated mode (Percentage) of commercial audio versions compared to the nominal mode.

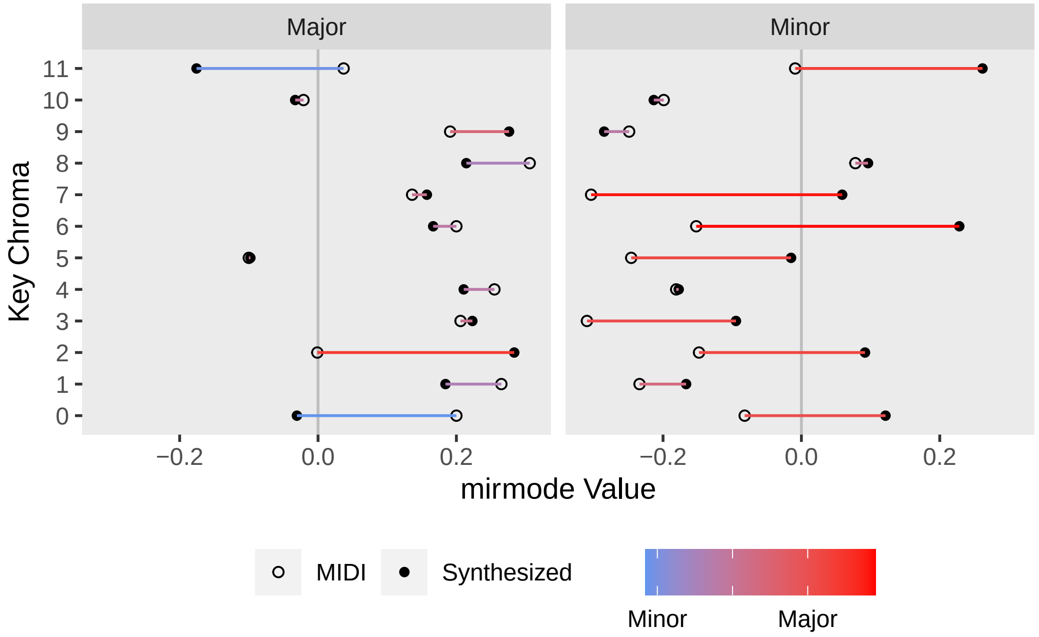

Figure 4: mirmode values for MIDI and Synthesized audio versions across key chroma and nominal mode. Lines indicate the variability between the two versions, where a red line indicates a synthesized audio file was predicted to be more major than the MIDI file, while a blue line indicates a synthesized audio file was predicted to be more minor.

Variability Across Versions

While substantial variability can be seen when considering a binary decision of < 0 = Minor / > 0 = Major, the variability of raw values across versions is also an important metric for evaluating the algorithm. Figure 5 shows the distribution of mirmode values for commercial audio in comparison to the MIDI files (see Figure 6 for individual datapoints).

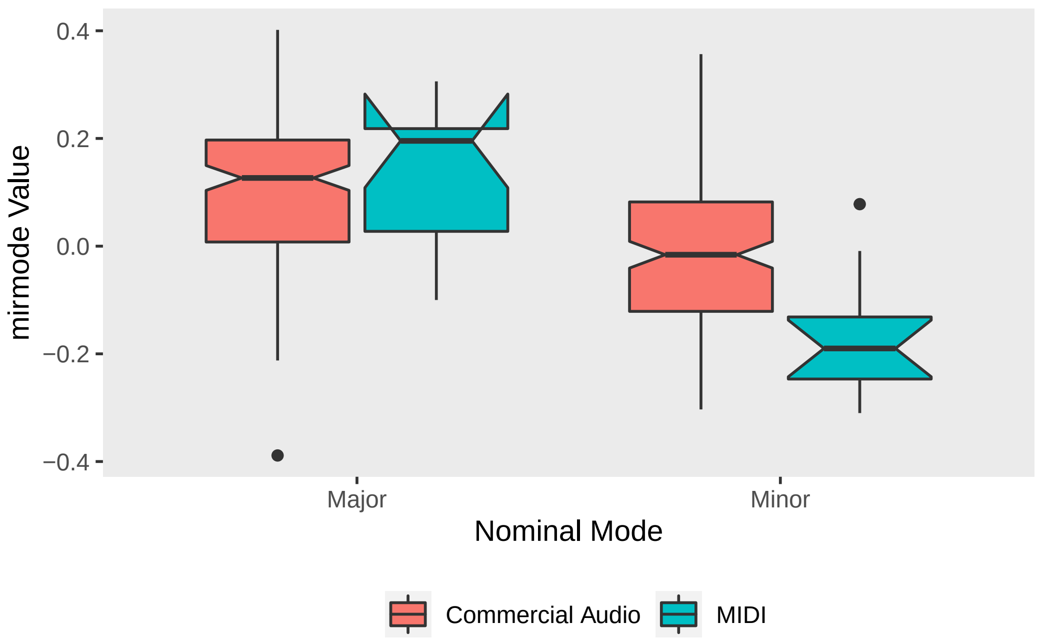

Figure 5: Distributions of mirmode values for commercial audio and MIDI files based on nominal mode.

mirmode Sum Method

In addition to the default method for calculating the mirmode coefficient, MIRtoolbox offers the option to calculate the coefficient by subtracting the sum of all minor key coefficients from the sum of all major key coefficients. However, since the values no longer necessarily lie between -1 and 1, where to draw a line between major and minor predictions is unclear, making the application of this method limited. In the case of the MIDI representations, the values lie between -0.003 and 0.003 while the commercial audio values lie between 2.381 and 6.222 (Figure 7). Additionally, discriminability between the nominally major and minor pieces is considerably less defined in comparison to the default method (Figure 5). As a result, the default method appears to be more conducive to real world analysis. Since it is difficult to establish where the line might be drawn between major and minor, tables for computed-nominal matches have not been included.

Discussion

Below are some points for discussion on the mirmode algorithm. Most of these deal with technical problems that might be contributing to the pattern of results found here.
Sampling rates: The chromagram used in this method is calculated using FFT. The temporal

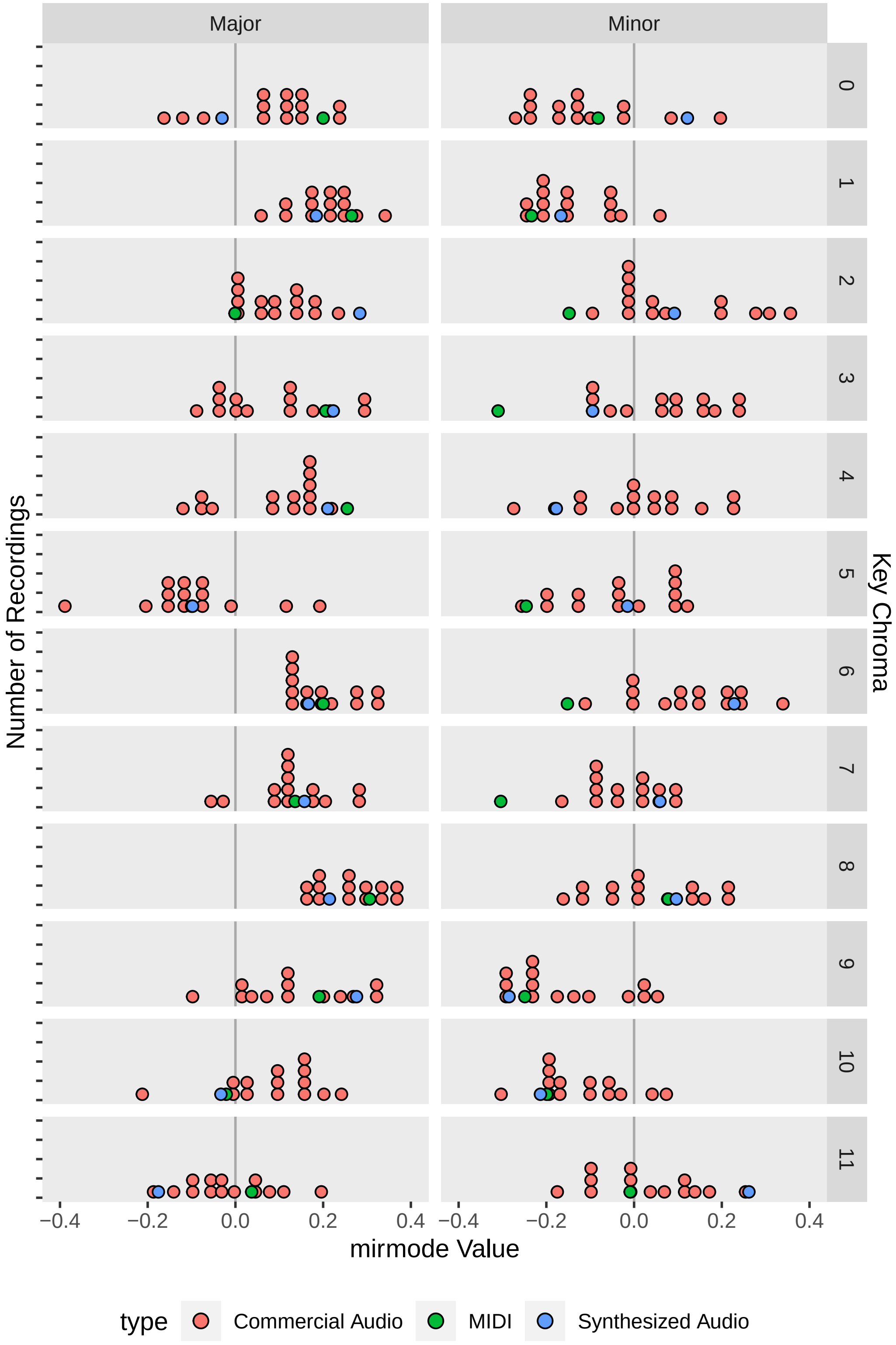

Figure 6: Distribution of individual versions across each prelude. Colours indicate type of file.

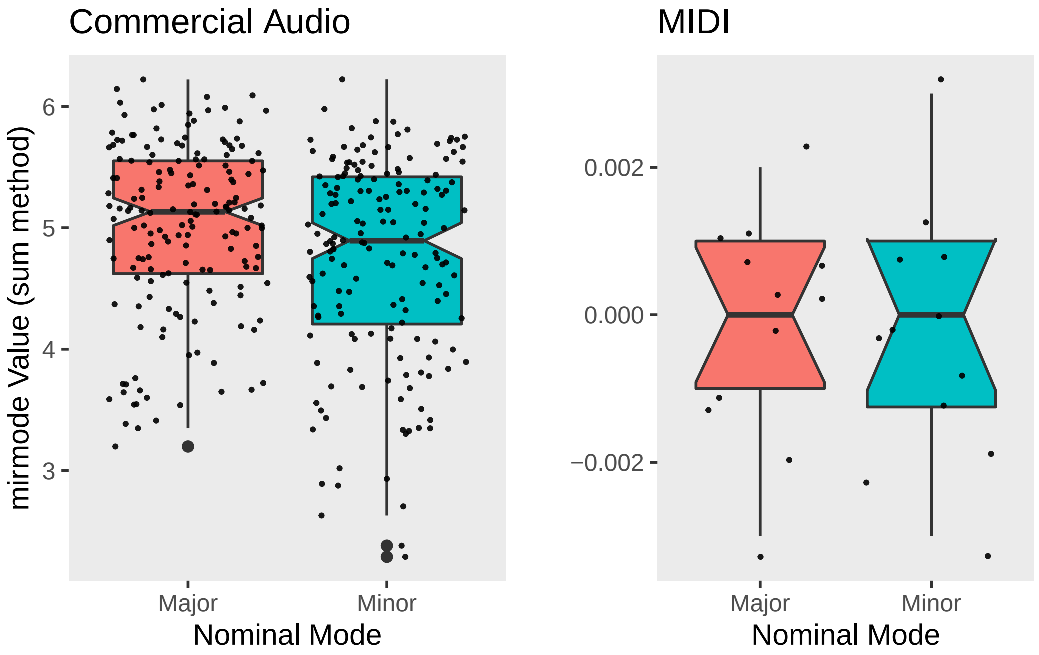

Figure 7: Distributions of mirmode values for commercial audio and MIDI files based on nominal mode.

precision of FFT is limited by the Nyquist limit (the limit of distinguishable frequencies) of a signal. In the case of commercial audio, our Nyquist limit is 20kHz, which is fine given the range of human hearing. However, the temporal precision of FFT at the Nyquist limit is defined by the sampling unit: 1 second. So at 20kHz, our temporal precision is 1s, at 10kHz our temporal precision is 500ms, etc, etc. I suspect that moving to higher sampling rates such as 96kHz or 192kHz (well within the range of our current technology) might result in a chromagram more accurate.

Chromagram Noise: Regardless of sampling rate, the chromagram will always be susceptible to some noise because instruments are seldom tuned exactly and consistently to 440Hz, and most microphones and preamplifiers (not to mention equalizers and compressors) produce some degree of non-linear distortion. This shouldn't be an issue for the synthesized audio, though. I think it would be worthwhile to explore beyond sample rates and see if this method works well even under ideal circumstances.

Krumhansl Schmuckler: While Krumhansl (1990) found the key finding algorithm had a 91.7% match rate to the nominal key in all 48 preludes in the WTC, these results do not seem to replicate with audio. One way to look at these results is that the theory behind this is fine, but the chromagrams are inaccurate. With more time I would have liked to explore some measure of comparing the chromagrams to the pitch class distribution. A final consideration of this algorithm is it does not consider the order of events: likely a large part of mode determination.

Confidence: The framework used here labels all values below zero as minor, while all values above zero are major. Given that the overall values fell into a limited range of the mathematically possible space, some consideration of confidence may be appropriate. For instance, some versions of the Krumhansl-Schmuckler keyfinding algorithm provide a value of confidence that considers not just the highest key coefficient, but how much higher that key coefficient is relative to the others. Given that a value of 1 or -1 is impossible in the mirmode scheme due to how Krumhansl-Schmuckler is calculated, what role do the coefficient values play beyond their sign?

The Minor Mode: Identification of the minor mode from statistical patterns of pitch sets presents some issues. On one hand, the relative minor of any given major key could contain the same pitch class set as it's relative major, albeit with different distributions: the algorithm does acknowledge this by having different weights for minor keys. However, music theory typically recognizes three forms of the minor scale: natural, harmonic, and melodic. While all three forms are used interchangeably in music, the weights used here are attempting to compensate for all three. The result is a scheme where none of the forms are really favoured. Pieces in the natural minor will likely skew to their relative major, pieces in the harmonic minor will outperform the melodic minor, and the elusive fourth form (the bebop harmonic minor used in jazz) is a more natural fit to these weights (see Figure 8). I'm not sure if when we listen to music we really make that level of distinction. A large body of research has tried to remedy this by creating new weighting schemes for the algorithm, but I think this approach just might not work. In much of this work (on symbolic notation), they find that identifying the relative or dominant or subdominant key is quite common.

Future Directions

The present exploratory analysis may be useful as a framework for testing audio file musical feature extraction more generally. Structural features such as tonality and mode are generally thought to be relatively unaffected by changes in surface and expressive features. However, this study shows that for statistical feature extraction algorithms, an extracted feature like mode might be highly variable across different versions of the same piece of music. Analyzing multiple versions of the same piece of classical music offers the opportunity to see if a given extracted feature is influenced by version-specific changes. Further work on human participants should explore how version influences judgment of structural features: while the nominal mode is often agreed upon by music experts, naive listeners may not be as binary in their judgments. Future testing of audio feature extraction algorithms like those discussed here should examine what characteristics are directly responsible for the variability across versions. While analyzing commercially produced audio files may have ecological validity and obvious applications in music information retrieval (such as recommendations systems), these files may not be precise enough for our current methods: files with larger sampling rates and

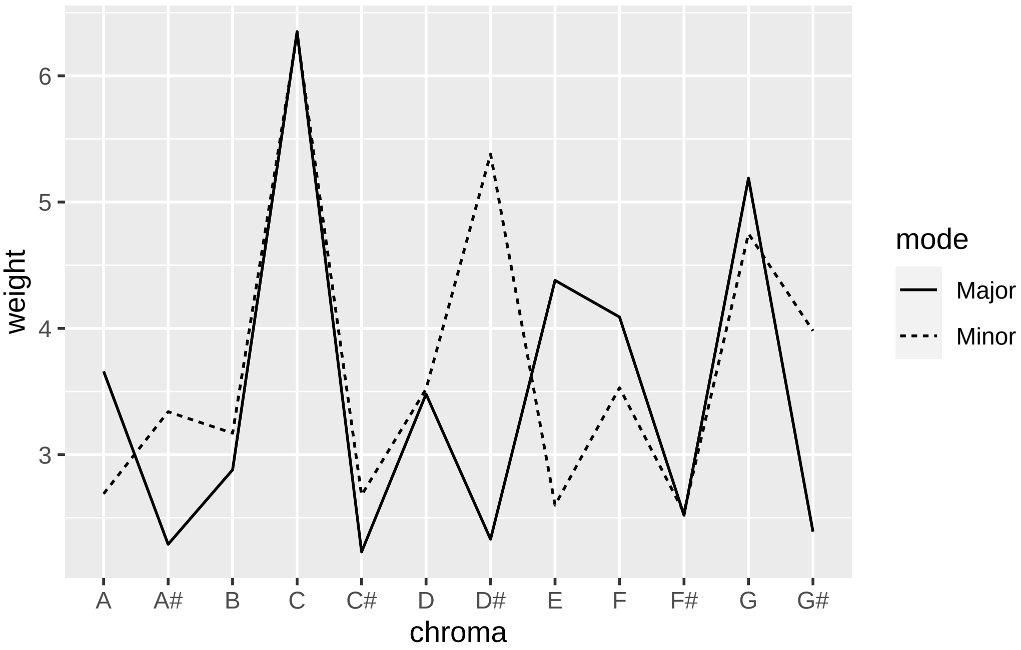

Figure 8: C Major/Minor weights for Krumhansl/Schmuckler keyfinding algorithm.

less compression (both dynamic and digital) may be more conducive to producing an accurate chromagram. Similarly, it may be the case that variables at the performer level may in fact have an influence on a statistical method of extracting mode. A “pipeline” approach analyzing audio from symbolic notation up to commercial grade recordings with expressive features and audio compression added in between may be useful for determining how each of these factors contributes to increasing extracted feature variability across piece. The present analysis focuses on one type of music from a single composer. Future research might extend version comparison to other styles, genres, time periods, and composers. That said, notated music for piano with limited version variation (in comparison to “covers” in contemporary music) are ideal for this analysis, as the pitch content, particularly the pitch set, of the versions will remain fairly similar. # Conclusion Here we analyzed multiple versions of the same set of pieces of classical piano music. While each version contains variation from multiple sources including performer-specific expressive features, recording technology, and room acoustic/instrument differences, these differences should not alter features that are considered to be compositional, such as mode. However, this exploratory analysis reveals a great deal of variability between versions indicating that the algorithm used by MIRtoolbox to determine mode is highly susceptible to variability between versions. In addition, the generally low coefficient values calls into question the confidence of any given mode determination by this algorithm. Moving forward, a more robust method for calculating mode that is not susceptible to these changes will be necessary, and other feature extractors should be examined with similar methods.

References

Bach, J. S., & Palmer, W. A. (2004). *The well-tempered clavier. Volume 1* (3rd ed). Alfred Pub. Co.

Cuthbert, M. S., & Ariza, C. (2010). *music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data*. 7.

Gómez, E. (2006). Tonal Description of Polyphonic Audio for Music Content Processing. *INFORMS Journal on Computing*, *18*(3), 294–304. https://doi.org/10.1287/ijoc.1040.0126

Krumhansl, C. L. (2001). *Cognitive foundations of musical pitch* (1. issued paperb). Oxford Univ. Press.

Lartillot, O. (2021). *MIRtoolbox 1.8.1 User Manual*. RITMO Centre for Interdisciplinary Studies in Rhythm, Time; Motion.

Lartillot, O., & Toiviainen, P. (2007). *A Matlab Toolbox for Musical Feature Extraction from Audio*. 8.

Appendix

Onsets

Piece	Standard Deviation	Mean	Coefficient of Variation
0 Major	22.248	139.4	0.160
0 Minor	20.401	133.067	0.153
10 Major	27.343	180.667	0.151
10 Minor	42.852	129	0.332
11 Major	25.614	142.733	0.179
11 Minor	20.978	84.667	0.248
1 Major	9.356	50.4	0.186
1 Minor	28.577	109.333	0.261
2 Major	17.999	120.133	0.150
2 Minor	28.156	165.2	0.170
3 Major	23.498	154.8	0.152
3 Minor	17.446	86.933	0.201
4 Major	20.053	115.467	0.174
4 Minor	36.257	145.533	0.249
5 Major	36.415	184.067	0.198
5 Minor	37.253	176.267	0.211
6 Major	15.379	104.667	0.147
6 Minor	17.361	123.467	0.141
7 Major	31.181	152.333	0.205
7 Minor	37.781	207.667	0.182
8 Major	6.897	68	0.101
8 Minor	11.106	106.067	0.105
9 Major	19.822	119.067	0.166
9 Minor	16.396	101.867	0.161

Table 4: Onset Standard Deviations

Piece	Standard Deviation	Mean	Coefficient of Variation
0 Major	0.886	4.852	0.183
0 Minor	1.048	6.353	0.165
10 Major	0.763	6.914	0.110
10 Minor	0.857	2.328	0.368
11 Major	1.009	4.879	0.207
11 Minor	1.032	3.203	0.322
1 Major	1.054	6.459	0.163
1 Minor	0.751	3.288	0.228
2 Major	0.886	6.97	0.127
2 Minor	0.7	6.174	0.113
3 Major	1.049	5.008	0.209
3 Minor	0.61	2.094	0.291
4 Major	0.711	4.123	0.172
4 Minor	1.172	3.796	0.309
5 Major	0.643	6.348	0.101
5 Minor	0.931	3.82	0.244
6 Major	0.791	4.181	0.189
6 Minor	0.719	5.675	0.127
7 Major	1.116	6.536	0.171
7 Minor	0.84	4.453	0.189
8 Major	0.585	3.917	0.149
8 Minor	0.694	4.082	0.170
9 Major	0.818	4.734	0.173
9 Minor	0.919	5.08	0.181

Table 5: Onset Rate Standard Deviations

Piece	Standard Deviation	Mean	Coefficient of Variation
0 Major	0.121	0.072	1.681
0 Minor	0.141	-0.096	-1.469
10 Major	0.116	0.079	1.468
10 Minor	0.103	-0.124	-0.831
11 Major	0.108	-0.027	-4.000
11 Minor	0.133	0.044	3.023
1 Major	0.071	0.2	0.355
1 Minor	0.09	-0.138	-0.652
2 Major	0.088	0.108	0.815
2 Minor	0.138	0.096	1.438
3 Major	0.127	0.095	1.337
3 Minor	0.124	0.057	2.175
4 Major	0.116	0.093	1.247
4 Minor	0.142	0.009	15.778
5 Major	0.133	-0.096	-1.385
5 Minor	0.122	-0.036	-3.389
6 Major	0.073	0.196	0.372
6 Minor	0.123	0.129	0.953
7 Major	0.093	0.132	0.705
7 Minor	0.079	-0.01	-7.900
8 Major	0.073	0.258	0.283
8 Minor	0.121	0.038	3.184
9 Major	0.141	0.163	0.865
9 Minor	0.128	-0.162	-0.790

Table 6: mirmode Standard Deviations

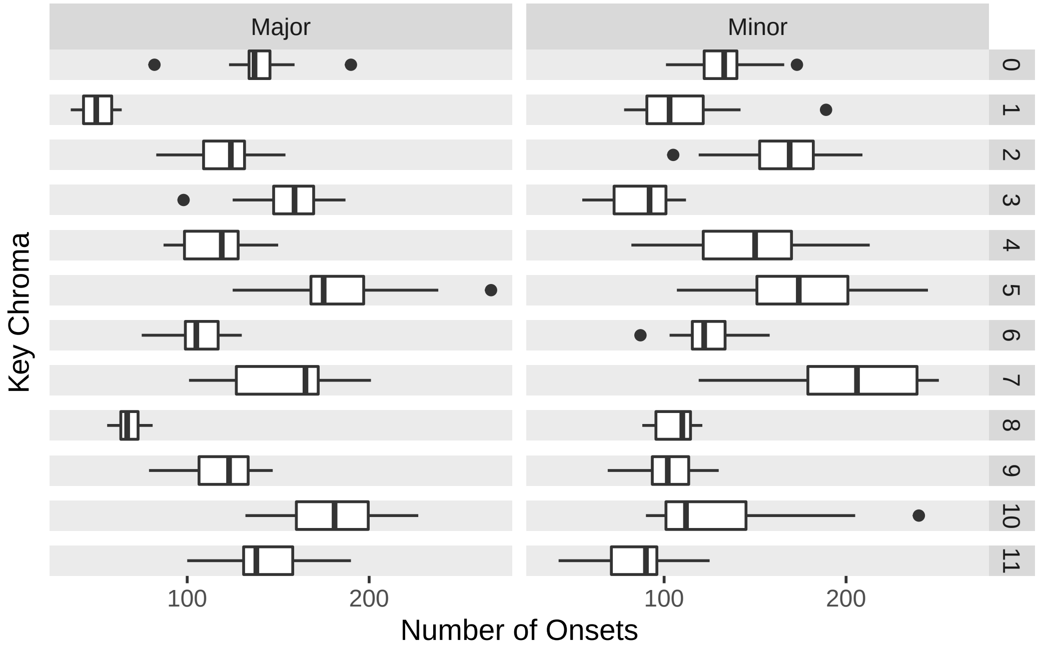

Figure 9: onsets

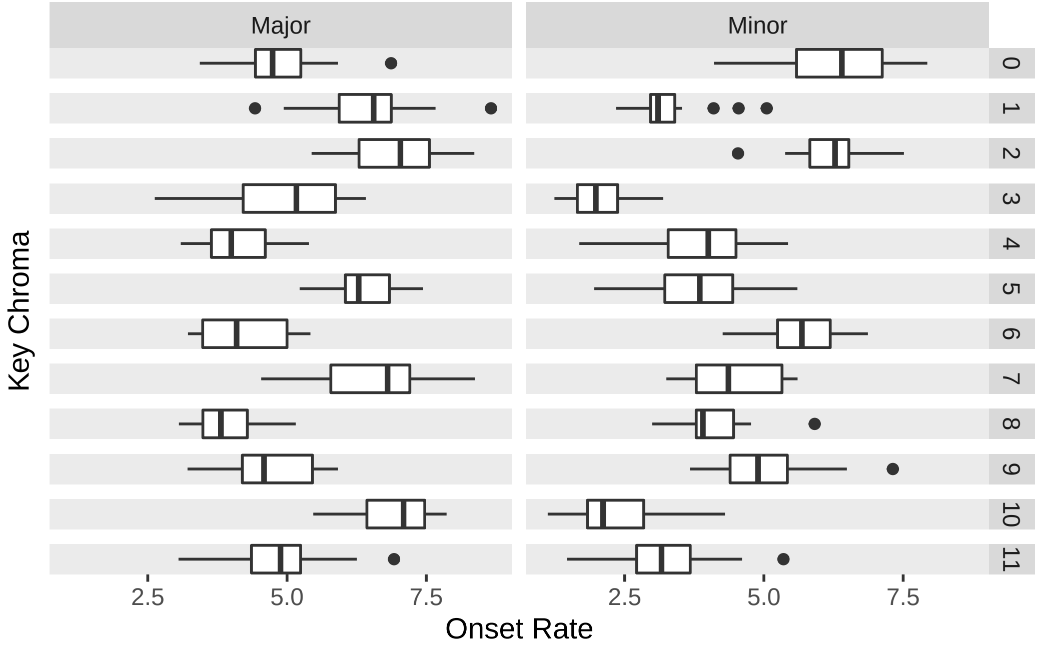

Figure 10: onsets

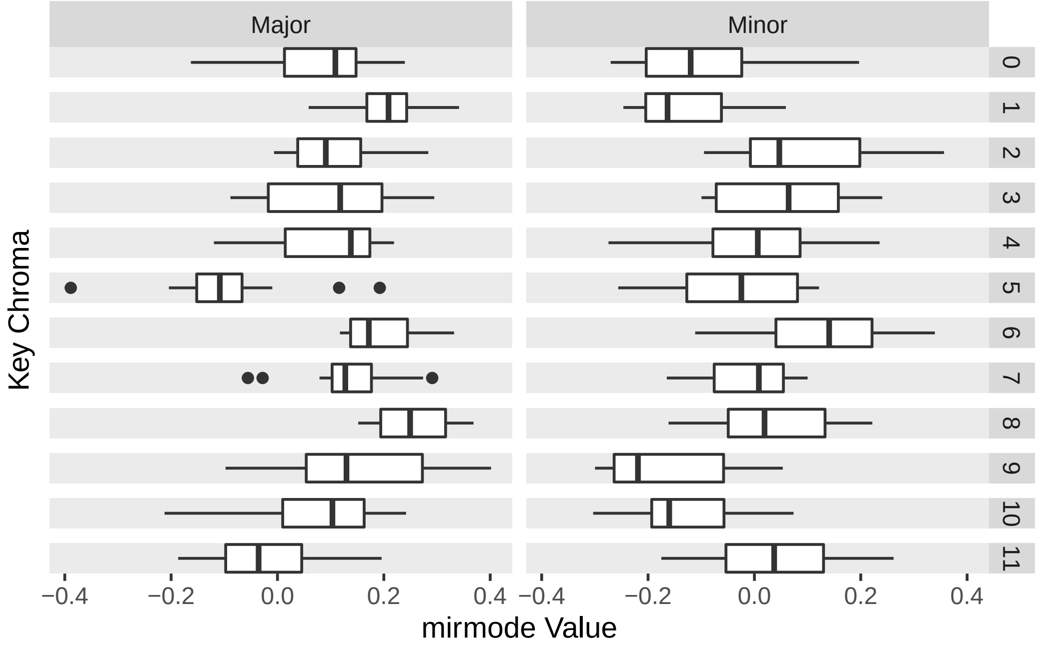

Figure 11: onsets