TISMIR

Odekerken, D., et al. (2021). Improving Audio Chord Estimation by Alignment and Integration of Crowd-Sourced Symbolic Music. *Transactions of the International Society for Music Information Retrieval,* 4(1), pp. 141–155. DOI: https://doi.org/10.5334/tismir.81

RESEARCH ARTICLE

# Improving Audio Chord Estimation by Alignment and Integration of Crowd-Sourced Symbolic Music

Daphne Odekerken*, Hendrik Vincent Koops*,† and Anja Volk*

Automatic Chord Estimation (ACE) is a fundamental task in Music Information Retrieval (MIR) and has applications in both music performance and MIR research. The task consists of segmenting a music recording or score and assigning a chord label to each segment. Although it has been a task in the annual benchmarking evaluation MIREX for over 10 years, ACE is not yet a solved problem, since performance has stagnated and modern systems have started to tune themselves to subjective training data. We propose DECIBEL, a new ACE system that exploits heterogeneous musical representations, specifically MIDI and tab files, to improve audio-based ACE methods. From an audio file and a set of MIDI and tab files corresponding to the same popular music song, DECIBEL first estimates chord sequences. For audio, state-of-the-art audio ACE methods are used. MIDI files are aligned to the audio, followed by a MIDI chord estimation step. Tab files are transformed into untimed chord sequences and then aligned to the audio. Next, DECIBEL uses data fusion to integrate all estimated chord sequences into one final output sequence. DECIBEL improves all tested state-of-the-art ACE methods by 0.5 to 13.6 percentage points. This result shows that the integration of crowd-sourced annotations from heterogeneous symbolic music representations using data fusion is a suitable strategy for addressing challenging MIR tasks such as ACE.

**Keywords:** Automatic chord estimation; data fusion; music alignment

## 1. Introduction

Automatic Chord Estimation (ACE) is an important task in Music Information Retrieval (MIR), with the goal of automatically estimating chords in audio recordings or symbolic music representations. ACE segments a musical piece so that the segment boundaries represent chord changes and each segment has a chord label. This is typically represented by a sequence of ⟨start time, end time, chord label⟩ triples.

The estimation of chords in a musical piece is used in various MIR tasks, such as cover song identification, key detection, genre classification, lyrics interpretation and audio-to-lyrics alignment (McVicar et al., 2014). Furthermore, ACE has direct applications in online learning platforms, such as Chordify,[1] that automatically extract chord sequences from any audio file, so that users can play along with their favourite songs.

ACE has been a task in the annual benchmarking evaluation Music Information Retrieval Evaluation eXchange (MIREX) since 2008. The main evaluation measure for ACE is (Weighted) Chord Symbol Recall (WCSR/CSR). CSR reflects the proportion of correctly labelled chords in a song. WCSR weights the CSR of songs in a data set by their length (Harte, 2010). State-of-the-art ACE methods yield WCSRs of around 75–87%, given a chord vocabulary of major and minor chords.[2]

However, a study on MIREX results from 2010 to 2015 reveals a stagnation in ACE performance (Scholz et al., 2016). Besides, various studies (Ni et al., 2013; Humphrey and Bello 2015; Koops et al., 2019) throw light on another issue of ACE: chord annotations are inherently subjective, which can result in multiple, heterogeneous chord annotations. Recently, Koops et al. (2019) introduced a 50-song data set of popular music, annotated by 4 professional musicians, and found only 73% overlap on average for the traditional major-minor vocabulary. The currently common practice to evaluate ACE by comparing the results to a single reference annotation is disputed by Humphrey and Bello (2015); Ni et al. (2013) and Koops et al. (2019). Ni et al. (2013) and Koops et al. (2019) even claim that modern ACE systems have started to overfit on standard ACE data sets, effectively mimicking the specific aspects of MIREX's reference annotations. This shows the need to invest research into exploiting chord sequence heterogeneity and subjectivity to improve ACE, both in representation and evaluation of chord sequences. In this paper, we focus on the former. Heterogeneity in chord representations allows for the exploitation of different analytical approaches to harmony (e.g. modelling chords both in audio and symbolic music representations), as well

* Department of Information and Computing Sciences, Utrecht University, NL

† RTL Netherlands, NL

Corresponding author: Daphne Odekerken (d.odekerken@uu.nl)

as the consideration of different subjective interpretations of the same musical piece (e.g. chord interpretations encoded in tab files).

Heterogeneity in representations is not unique for chord sequences. In fact, for a given musical work, often multiple different versions in the symbolic (e.g., MIDI or chord label representations), audio, and visual (sheet music) domain exist. Each of these domains have their own domain-specific variability regarding particular aspects. For example, audio recordings may differ in performance parameters such as tempo, expressive timing and dynamics. Symbolic and visual representations of a musical work might differ because of different music editing conventions, historical distance from the original composing process, or publications for a commercial aim rather than for scholarly accuracy (Preston et al., 2012).

The idea of exploiting multiple versions, from different domains or different manifestations within a domain, is often referred to as *cross-version analysis* (CVA) (Ewert et al., 2012). The main idea of CVA is that after alignment, different analytical results from the same piece can be analysed and compared. As such, it presents an opportunity to compare methods across different versions or to create methods that exploit the domain-specific strengths while attenuating their weaknesses in order to create higher quality analyses (e.g. Konz and Müller, 2012; Koops et al., 2016).

In this light of CVA, we propose DECIBEL (DEtection of Chords Improved By Exploiting Linking symbolic formats), a novel system that exploits heterogeneous symbolic music representations, specifically MIDI and tab files, for improving ACE on popular music. MIDI and tab files can be considered as crowd-sourced note and chord transcriptions and are widely available for this genre on websites like Ultimate Guitar[3] and MIDI World.[4] Since DECIBEL only relies to a small extent on training on reference annotations, it prohibits further overfitting to existing data sets.

To evaluate DECIBEL, we compare its performance to state-of-the art ACE systems submitted in the MIREX competitions of 2017 to 2020, as well as a commercial ACE method. Using the existing one-to-one MIREX metrics, we find that DECIBEL improves each of the twelve tested ACE systems. This shows that DECIBEL effectively exploits heterogeneous chord sequences when evaluated using these metrics. We consider this as a promising step towards exploiting heterogeneous chord sequences for multiple chord sequence evaluations in the future, once appropriate evaluation metrics for a simultaneous evaluation of multiple chord sequences have been established.

**Contributions:** Our work extends earlier research in two ways. First, we are the first to propose a framework to create a rich harmonic representation from three heterogeneous formats, exploiting the combination of audio, MIDI and tab files. Second, we show that we can use symbolic music representations for improving audio ACE, using twelve recent state-of-the-art audio-based ACE methods as baseline. For this purpose, we (re-) implemented existing and new methods for tab scraping, tab parsing, tab-audio alignment, MIDI chord estimation, MIDI-audio alignment and data fusion in Python. We made our implementations available to the research community.[5]

**Synopsis:** The remainder of this paper is structured as follows. In Section 2, we discuss related work. Section 3 gives an overview of the DECIBEL system, introduces our data set and describes DECIBEL's audio subsystem. In Sections 4 and 5 we focus on DECIBEL's MIDI and tab subsystems. We show how we combine harmonic information from audio, MIDI and tab files in Section 6, present our results in Section 7 and conclude in Section 8.

## 2. Related Work

In this section, we first describe existing methods for ACE. Next, we relate DECIBEL to earlier approaches that integrate symbolic music and audio in the chord estimation task.

### *2.1 Existing audio-based ACE methods*

For a detailed overview of existing audio-based ACE methods we refer to McVicar et al. (2014) and Pauwels et al. (2019). Traditionally, most ACE methods used the following pipeline: first, audio data is partitioned into a training set and a test set and features are calculated on both partitions. Typically, features are either (variations on) the traditional chromagram (Wakefield, 1999) or features trained using deep neural networks (Korzeniowski and Widmer, 2016a). Subsequently, the features from the training set are used to train the parameters of a model. There is a wide variety of models used in earlier work, for example Hidden Markov Models (HMMs) (Sheh and Ellis, 2003), Dynamic Bayesian Networks (Mauch, 2010), Conditional Random Fields (Burgoyne et al., 2007) and Deep Recurrent Neural Networks (Sigtia et al., 2015). As a next step, the chord labels for the test set are estimated using this trained model. Finally, the performance of the system is evaluated by comparing the labels calculated by the model to the reference (ground truth) chord labels. In recent research on ACE, this modular approach is replaced by a deep learning approach, in which systems are trained more in an end-to-end fashion, for example the method by Chen and Su (2019).

### *2.2 Previous approaches of audio-symbolic integrated ACE*

Our work builds on previous approaches to integrate symbolic music and audio in the chord estimation task.

#### 2.2.1 Using MIDI

Ewert et al. (2012) introduce a CVA framework for comparing harmonic analysis results from different musical domains. After collecting a MIDI file for each audio file in a 112-song data set, they use two chord recognition methods for MIDI data and align each MIDI file to the corresponding audio recording. This way, they create a harmonic representation for each of the songs, which contains three chord label sequences: the ground-truth labels and the re-aligned outputs that were obtained by the two MIDI chord recognition systems. By visualising

this harmonic representation, they demonstrate how it can be used for qualitative error analysis of automatically generated chord labels, which contributes to the understanding of an ACE algorithm's behaviour and the properties of the underlying music material. This study lays the foundation for the work proposed in our paper, in which we expand the harmonic representation with chord labels from multiple MIDI and tab files for each audio recording. Furthermore, we show how this enriched harmonic representation can be used to improve ACE.

#### 2.2.2 Using tabs

The integration of tab and audio files with respect to chord estimation was earlier researched by McVicar et al. (2011). They show that an HMM-based system for audio ACE can be improved significantly by incorporation of external information from guitar tabs. For this purpose, they introduce four variations on the traditional Viterbi algorithm. The most promising variation is Jump Alignment, which aligns the chords that are extracted from tabs to the audio file, thereby allowing jumps from the end of any line to the beginning of any line in the tab file. We implemented Jump Alignment as part of DECIBEL (see Section 5.2).

#### 2.2.3 Data fusion

Exploring data fusion methods within machine learning for exploiting different web resources has been applied in several domains, such as for the modelling of book, restaurant, sport, flight, and stock data (Li et al., 2012). In our approach, we use different web sources of MIDI and tab files to improve audio ACE. The integration of heterogeneous output of multiple ACE algorithms in the context of MIR was first proposed by Koops et al. (2016). The authors experiment with three different techniques to combine chord sequence estimates from the MIREX 2013 ACE submissions into one final output sequence for each song. They show that their data fusion method yields the best results in terms of WCSR. In our study, we broaden the definition from Koops et al. (2016) by also including chord sequences from other sources than audio. We use a similar data fusion method to combine the chord labels obtained not just from audio, but also from MIDI and tab files.

## 3. The DECIBEL Framework

In this section we describe how DECIBEL integrates heterogeneous symbolic music representations for improving ACE. This is illustrated in **Figure 1**.

DECIBEL uses a data set of audio, MIDI and tab files. For each song, each of the three representations (audio, MIDI and tab) is mapped to an audio-timed chord sequence, which is a sequence of chord events, consisting of a start time, end time and chord label. The possible chord labels are specified by the chosen chord vocabulary; in this study, we choose a vocabulary of all 24 major and minor chords and the no-chord symbol. The method for this chord estimation step depends on the representation: we used three different methods for audio, MIDI and tab representations. At this point, we have a harmonic representation, consisting of potential chord sequences for the song, obtained from symbolic and audio representations. As a final step, DECIBEL uses data fusion to combine these chord sequences into one final chord sequence.

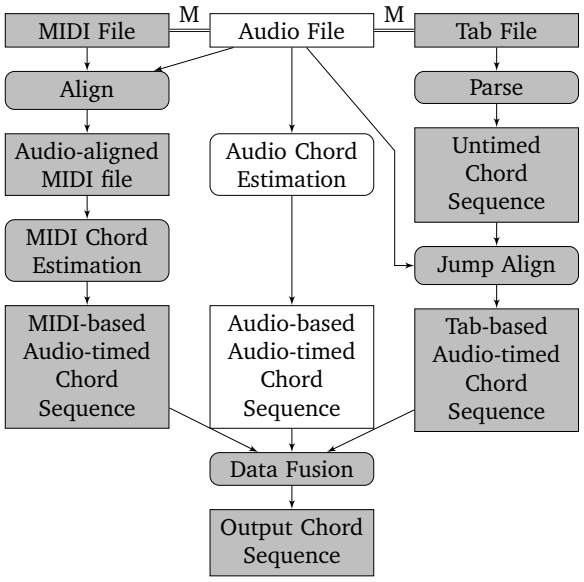


**Figure 1:** Diagram of DECIBEL's framework. The M represents the matching between different representations of the same song. Data formats are depicted by rectangles; procedures are represented as rounded rectangles. The grey elements show how DECIBEL extends existing audio ACE methods.

### *3.1 Data set*

DECIBEL's data set of audio, MIDI and tab files is based on a subset of the Isophonics Reference Annotations (Mauch et al., 2009). The Isophonics data set contains chord annotations for 180 Beatles songs, 20 songs by Queen, 7 songs by Carole King and 18 songs by Zweieck. We only use the songs by the The Beatles and Queen, as there were no MIDI or tabs for Zweieck available and we observed some inconsistencies in the Carole King annotations.

After acquiring the audio files and annotations, we collected MIDI and tab files and matched them to the songs in our data set. First, we downloaded MIDI files of the aforementioned 200 songs from 9 websites. This way, we found 770 MIDI files with unique MD5-checksums, hence multiple MIDI files (3.85 on average) map to a single audio file. We matched the MIDI and audio files by hand, based on the MIDI file name. Next, we obtained tabs from Ultimate Guitar by scraping all tabs from The Beatles and Queen from Ultimate Guitar's website and manually matching the tabs to the audio files, based on song title. Tabs from songs that were not in the data set, were discarded. This resulted in 1668 matched tabs, consisting of 974 chord sheets and 694 guitar tablature files.

MIDI and tab files are easy to obtain, but their qualities are extremely diverse. We did not do any manual selection of these files, because that would influence the results. Instead, we designed DECIBEL in such a way that

high-quality files are more likely to be integrated than low-quality files, as we will describe in the next sections.

### *3.2 ACE on audio*

We evaluate DECIBEL by considering various audio ACE systems. For each audio ACE system, we use the chord sequences that it estimated on our data set as input for DECIBEL. Subsequently, we compare DECIBEL's chord sequences to the sequences estimated by the audio ACE system only. We experiment with twelve state-of-the-art audio ACE systems: the submissions for the MIREX ACE competitions from 2017–2020 and the Chordify algorithm (based on Koops et al. (2017), version of 2017). Their performance in terms of WCSR is shown in the column "Audio WCSR" of **Table 4**. Note that the results slightly differ from the MIREX evaluation reported on the Isophonics data set, as we use a subset consisting of The Beatles and Queen songs.

Since we use state-of-the-art audio ACE systems, it is not trivial to improve them by integrating information from MIDI files and tabs.

## 4. ACE on MIDI

In this section, we discuss the subsystem of DECIBEL that detects an audio-timed chord sequence based on a MIDI file, illustrated in **Figure 2**. In order to receive audio-timed chord labels for each MIDI file, DECIBEL first finds an optimal alignment from the MIDI file to the audio file, realigns the MIDI file using this alignment (Section 4.1) and then uses a MIDI chord recogniser to estimate the chord labels on the realigned MIDI file (Section 4.2). Finally, output of the alignment and chord estimation methods is used to select the estimated best MIDI file for each song (Section 4.3).

### *4.1 MIDI-to-audio alignment*

Music alignment is the procedure which, given any position in one representation of a piece of music, determines the corresponding position within another representation. Alignment methods are typically based on either statistical approaches, e.g. HMMs (Cuvillier and Cont, 2014), or Dynamic Time Warping (DTW) (Raffel and Ellis, 2016).

For alignment between MIDI files and audio recordings, DECIBEL uses a DTW algorithm by Raffel and Ellis (2016). We selected this algorithm for three reasons. First, it calculates an easily interpretable alignment error score, which gives a good indication of the alignment quality, without requiring ground truth data. Second, the algorithm is conceptually simple and easy to implement. Third, Raffel and Ellis (2016) optimised the parameter settings for a synthetically-created data set of 1000 MIDI files that is similar in size and genre to ours. We use these parameters without modification.

The output of the DTW system is an alignment path and its alignment error score. The alignment path specifies which time point in the MIDI file is aligned to which time point in the audio file. We utilise this path to adapt the timing in the MIDI file, using the pretty_midi package (Raffel and Ellis, 2014). The result of this is an audio-aligned MIDI file.

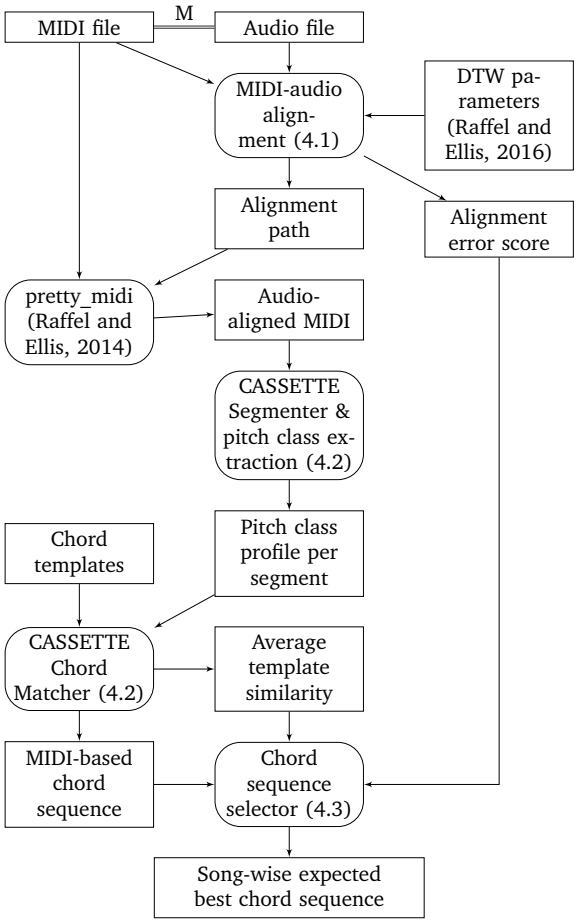


**Figure 2:** Diagram of DECIBEL's MIDI subsystem. The M represents the matching between different representations of the same song. Data formats are depicted by rectangles; procedures are represented as rounded rectangles.

The alignment error score is the mean distance between aligned pairs of frames on the optimal path divided by the mean distance in the entire submatrix containing the aligned portion of both feature sequences (Raffel and Ellis, 2015). This score gives an indication of the alignment quality. A qualitative evaluation on 500 real word MIDI/audio pairs by Raffel and Ellis (2016) showed that the alignment error score is a reliable measure for the quality of the alignment in most cases: in general, MIDI/audio pairs with an alignment error score below 85% are aligned well. In order to verify whether these results are applicable to our data set, we evaluated the performance of the DTW system on a small random sample of 25 MIDI files. For each MIDI file, we synthesised the realigned MIDI version and played it simultaneously with the original audio file, listening to the realigned MIDI file on the left earphone and the original audio on the right earphone. In this listening test, we classified each MIDI file into one of three alignment quality categories: bad alignments; alignments with minor issues; or good alignments. Our evaluation confirmed Raffel and Ellis's observation that alignments with a low alignment error score are good, while alignments with a high alignment error score (above 85%) have major issues, e.g. correspond

to MIDI files that were transposed to another key or badly transcribed. We conclude that the alignment error score gives a good indication of the alignment quality. This turns out to be very useful for selecting the best realigned MIDI file, as we will see in Section 4.3.

### *4.2 Chord estimation on MIDI*

In this section, we introduce our chord estimation algorithm CASSETTE, which extracts chord sequences from MIDI files that have been aligned to the audio recordings as described in the previous section.

Chord estimation from MIDI files is the task of dividing a MIDI file into segments, in such a way that each segment boundary corresponds to a chord change, and assigning a chord label to each segment. In contrast to audio chord estimation, only few methods have been proposed that extract chords from symbolic representations like MIDI. Early works (Winograd, 1968; Maxwell, 1992; Temperley and Sleator, 1999) are grammar- or rule-based approaches to perform automatic tonal (Roman numeral) analysis. Other systems combine segmentation and template matching, applying tie-breaking rules (Pardo and Birmingham, 2002; Scholz and Ramalho, 2008); use probabilistic models to perform functional harmonic analysis on MIDI data (Raphael and Stoddard, 2004); or perform chord estimation using an HMPerceptron model, in which the domain knowledge is modelled in Boolean features (Radicioni and Esposito, 2010). In more recent research by Masada and Bunescu (2019), a model based on semi-Markov Conditional Random Fields (semi-CRFs) performs a joint segmentation and labelling of symbolic music. Each of these algorithms would require modification and/or labelled training data in order to be used in DECIBEL: (1) rule-based systems (Winograd, 1968; Maxwell, 1992; Temperley and Sleator, 1999) are designed for a specific music genre (usually classical music); (2) some algorithms (Winograd, 1968; Maxwell, 1992; Temperley and Sleator, 1999; Raphael and Stoddard, 2004) recognise functional harmony instead of chord labels; and (3) other systems (Radicioni and Esposito, 2010; Masada and Bunescu, 2019) are based on machine learning, which requires a lot of labelled training data that is not available for MIDI files of popular music. Therefore, we designed CASSETTE (Chord estimation Applied to Symbolic music by Segmentation, Extraction and Tie-breaking TEmplate matching). CASSETTE is a template-matching based algorithm for MIDI chord recognition that is easy to implement and understand and does not require any training. It is based on the method by Pardo and Birmingham (2002), but we adapted the segmentation method and use alternative tie breaking rules.

CASSETTE first segments the MIDI both on the bar and on the beat level. In many cases, segmentation on the bar level is sufficient for popular music, as chord changes in this genre are often placed on the downbeat, i.e. the first beat of a bar. An advantage of segmentation on the bar level is that non-harmony notes, which tend to be short, are less problematic in the template matching step as they have relatively lower weights than the (typically longer) harmony notes. On the other hand, segmentation on the bar level does not work well for songs that have chord changes at other positions than the start of a bar. Therefore CASSETTE also segments the MIDI file on the beat level, which typically results in a chord sequence with more chord changes. For segmentation, CASSETTE relies on the get_downbeats and get_beats functions of pretty_midi (Raffel and Ellis, 2014).

As a second step, CASSETTE extracts a weighted pitch class profile for each of the segments of the MIDI file. For each given segment, CASSETTE (1) extracts the notes sounding between its start and end time; (2) computes for each note the product of the MIDI velocity and the proportion of the segment during which this note sounds; and (3) sums this product over all notes in the same pitch class. Consider for example a $\frac{4}{4}$ bar that only contains a quarter note C with a MIDI velocity of 100. For bar segmentation, the weighted pitch class profile of this bar would be [25, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. For beat segmentation, the weighted pitch class profile corresponding to the first beat would be [100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]; for the next three beats, the weighted pitch class profile would be [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. In the resulting vector, louder notes (with higher velocity) and longer notes (with greater duration) in the MIDI file are more important than softer or shorter notes. CASSETTE normalises the weighted pitch class profile by dividing each element by the total sum of all its elements (provided that this sum is larger than 0). This makes the feature invariant to the total loudness and duration of the notes in the segment. For bar segmentation, the resulting feature is [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. For beat segmentation, the feature for the first beat is [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] as well; for the next three beats, the feature is [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].

Third, CASSETTE finds the best matching chord for each segment, by assigning the chord template that is most similar to the normalised weighted pitch class profile of the segment. In this study, we use a vocabulary of 25 chords, consisting of all 24 major and minor chords and the no-chord symbol. A chord template is a 12-dimensional vector, in which an entry is 1 if the corresponding note occurs in that chord and 0 otherwise. For example: the D minor chord template is [0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0].

The similarity between a segment's weighted pitch class profile and a template is measured by the template similarity score. This score is based on the score in Pardo and Birmingham (2002) and is calculated with the formula $S = P - (N + M)$. $P$ is the positive evidence: the sum of the weights of the pitch classes of the bar that match a template element. $N$ is the negative evidence: the sum of the weights of the pitch classes of the bar that do not match a template element. $M$ stands for misses: the count of template elements not matched by any note. High scores correspond to well-matched templates. For each segment, CASSETTE assigns the chord with the highest template similarity score. If the score is −3 or lower, the algorithm assigns a no-chord. If multiple templates have the same template similarity score, CASSETTE selects the template whose root pitch has the greatest weight in the segment's pitch class profile.

In order to evaluate CASSETTE, we selected the 50 MIDI files with the lowest alignment error score from our data set and tested them against the Isophonics annotations.

With alignment error scores ranging from 0.458 to 0.603, these MIDI files are probably well aligned to the audio. Then, we ran the CASSETTE algorithm on the audio-aligned versions of these 50 MIDI files and calculated the WCSR for each of the resulting chord label sequences, as shown in **Table 1**. Note that particularly the beat-based MIDI chord recognition method performs quite well in terms of WCSR with a score of 80.0%. For beat segmentation, the CSR scores of the 50 best-aligned MIDI files range from 66.1% to 95.8%, with a standard deviation of 0.072; for bar segmentation, the CSR scores range from 37.9% to 95.8% with a standard deviation of 0.158. The reason for the very low CSR of 37.9% is that the corresponding lab file is strongly undersegmented. In general, the bar-based method is undersegmenting, whereas beat-based chord recognition has minor oversegmentation issues. This also explains the lower WCSR score of 71.0% of the bar-based method. We conclude that CASSETTE performs reasonably well on our data set with popular music and a limited chord vocabulary.

Another interesting property of CASSETTE is that it computes template similarity scores. A high score indicates a well-fitting chord label while a low-score segment probably has a less suitable chord label. The average of the scores over all segments is an approximate score for the quality of the estimated chord sequence. We call this score the average template similarity and we will discover its advantages in the next section.

### *4.3 MIDI file selection*

In the previous section we showed that CASSETTE was quite successful for the 50 best-aligned MIDI files. Yet extending the aforementioned experiment to the full data set showed a considerably worse performance: when comparing the chord sequences found by the MIDI file alignment and chord recognition system on all MIDI files, and averaging the performance across all MIDI files, we found a WCSR of just 65.2% for beat-based chord recognition and 62.3% for bar-based chord recognition. For the songs that have at least one well-aligned MIDI file, these numbers are 66.5% and 63.4% respectively. These poor results are partly due to low-quality MIDI files. **Table 2** shows that MIDI quality highly influences the ACE performance. To make sure that the rows in this table are mutually comparable, we only considered the songs for which at least one well-aligned MIDI file was available. If the worst MIDI file for each song is chosen, then the WCSR for our data set is 46.3% (beat) and 44.9% (bar); if the best MIDI file for each song is chosen, then the WCSR for our data set is 79.7% (beat) and 75.6% (bar).

**Table 1:** Results of MIDI chord recognition for the 50 MIDI files with the lowest alignment error score, in terms of WCSR, oversegmentation (OvS), undersegmentation (UnS) and segmentation (Seg) as defined by Harte (2010).

| Segmentation | WCSR | OvS | UnS | Seg |
|---|---|---|---|---|
| **Beat** | 80.0% | 83.0% | 89.1% | 80.8% |
| **Bar** | 71.0% | 95.4% | 67.2% | 67.1% |

In this section, we describe a method to select the estimated best MIDI file for each song. Since we cannot calculate the CSR for unlabelled data, we use a proxy measure based on the alignment error score and average template similarity.

As reported in Section 4.1, some MIDI files are badly aligned. Accordingly, these files will typically not yield good chord labels. Therefore, we discarded all MIDI files with an alignment error score higher than 85%. This leads to a great shift in performance: the WCSR on all MIDI files that are sufficiently aligned, is 73.8% (beat) and 70.1% (bar). Note that these WCSRs are measured over all songs for which there was at least one well-aligned MIDI file; we had to exclude one song for which all MIDI files had an alignment error score of over 85% from the calculation.

After this MIDI file selection step based on alignment quality, we have 592 out of our initial 770 MIDI files left. We do an additional selection step to obtain the expected best MIDI file for each song, using the average template similarity score calculated by CASSETTE. The CSR correlates with the average template similarity: the Pearson correlation coefficient is 0.542 for beat segmentation and 0.587 for bar segmentation. For most songs, the CSR of the MIDI file with the highest average template similarity is (almost) as good as the CSR of the actual best file, as shown in **Figure 3**. The plot shows that there are very few songs for which there is a large difference between CSR of the best and CSR of the estimated best MIDI file: there are only three songs for which the difference is greater than 0.5. In two of these songs, the estimated best MIDI was a semitone transposed, compared to the original audio. In the third song, the estimated best MIDI is a transcription of only a part of the song. However, in the vast majority of songs, the difference in CSR between the estimated and actual best MIDI files is small: most points are close to the diagonal line. We conclude that the average template similarity is a suitable measure to select the estimated best MIDI file for each song. This is also reflected in the performance shown in **Table 2**: the chord sequences from MIDI files selected with our proposed selection method (Estimated best) have WCSRs of 76.3% and 72.2% and thereby outperform the performance of all well-aligned MIDI files by over two percentage points.

**Figure 4** shows the CSR distribution for all MIDI files and all estimated best MIDI files for each song. It shows that MIDI files corresponding to the poorest chord estimates (blue peaks on the left) are typically not selected.

In this section, we introduced and evaluated the MIDI subsystem of DECIBEL. This subsystem extracts chord sequences from MIDI files by first aligning the MIDI file to the audio file and then running CASSETTE, a simple chord estimation method, on the re-aligned MIDI file. We also showed how we can select "good" (and in many cases the best) MIDI files for each song by (1) ignoring files with a high alignment error score and (2) selection of the file with the highest average template similarity. This resulted in WCSRs of 76.3% for the beat segmentation and 72.2% for bar segmentation for songs with at least one well-aligned MIDI file.

**Table 2:** Performance comparison of five MIDI file selection methods on the songs for which there was at least one well-aligned MIDI file in terms of WCSR, oversegmentation, undersegmentation and segmentation as defined by Harte (2010).

| | WCSR | | OvSeg | | UnSeg | | Seg | |
|---|---|---|---|---|---|---|---|---|
| | **Beat** | **Bar** | **Beat** | **Bar** | **Beat** | **Bar** | **Beat** | **Bar** |
| **Min CSR** | 46.3% | 44.9% | 76.4% | 89.7% | 75.5% | 62.0% | 66.6% | 61.1% |
| **All (averaged)** | 66.5% | 63.4% | 79.2% | 91.8% | 83.3% | 68.6% | 73.9% | 67.7% |
| **Well-aligned (averaged)** | 73.8% | 70.1% | 80.2% | 93.0% | 86.8% | 71.4% | 76.8% | 70.5% |
| **Estimated best** | **76.3%** | **72.2%** | **81.5%** | **93.2%** | **87.5%** | **73.5%** | **78.3%** | **72.7%** |
| **Max CSR** | 79.7% | 75.6% | 82.8% | 93.6% | 87.9% | 73.4% | 79.6% | 72.6% |

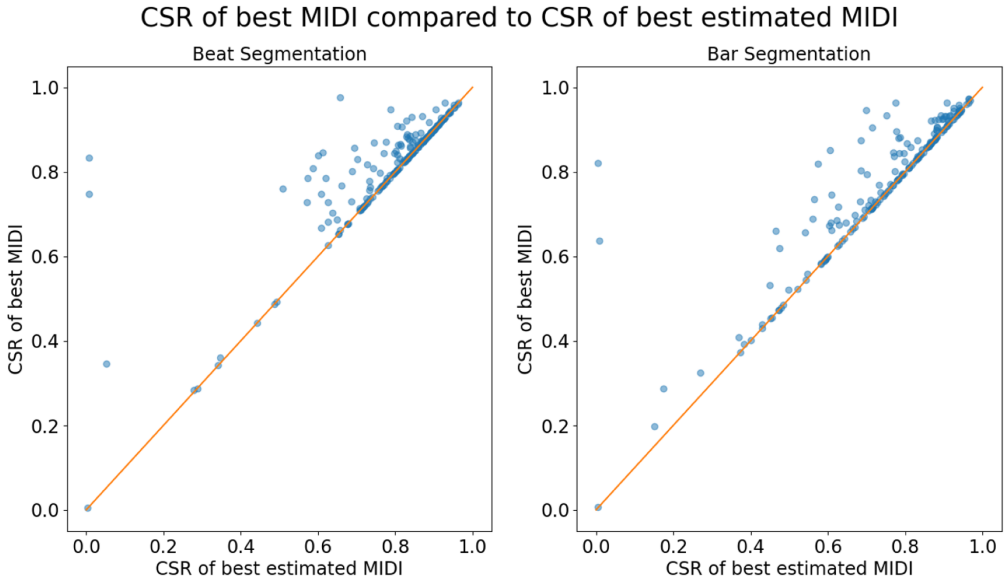


**Figure 3:** CSR of the real best MIDI file compared to the CSR of the estimated best MIDI file for both beat and bar segmentation. Points on the diagonal line (i.e. $x = y$) correspond to songs for which the best MIDI file was estimated correctly. The vertical distance between each point and the line is the difference between the CSR of the best MIDI file and the CSR of the estimated best MIDI file.

## 5. ACE on Tabs

Next, we describe the subsystem of DECIBEL that uses tab files for estimating chord labels. First, the tabs are parsed and the chord information is extracted (Section 5.1). Then, as described in Section 5.2, DECIBEL aligns the chord information to the audio file. This results in multiple chord estimates. Finally, DECIBEL selects the expected best tab file for each song as described in Section 5.3.

### *5.1 Tab parsing*

Before aligning the tab files to the audio, DECIBEL first needs to parse them and extract the chord information. DECIBEL parses the tabs in a similar way to the parser proposed by Macrae (2012). First, it classifies each line in the tab file to a line type. Then, it segments the tab by splitting on empty lines. As a next step, all systems in each segment are identified. A system is a set of subsequent lines that belong to each other. For example: a tab system consists of exactly six subsequent tab lines. In chord sheets, a common system is the alternation between chord lines and lyrics lines. From these systems, DECIBEL then extracts the chord labels, retaining line information (i.e. the line of the chord in the text file).

### *5.2 Tab-to-audio alignment*

Next, DECIBEL needs to align the chord labels to the audio file. To the best of our knowledge, there exist only four algorithms that use tabs in audio ACE, proposed by McVicar et al. (2011). The most promising of these four algorithms, which we use in DECIBEL's tab subsystem, is Jump Alignment, an HMM-based approach. Following McVicar et al. (2011), we refer to the sequences of chords parsed from tab files as Untimed Chord Sequences (UCSs). We only use tabs with a UCS of at least five chords: if less than five subsequent chords were found in the tab parsing step, it is very unlikely that this tab will contribute to a better chord label estimation. For the large majority of tab files (1438 out of 1668), at least five chords were identified. There was just one song in our data set for which no suitable tab file remained.

Given an audio recording and a tab file from the same song, the hidden states in the HMM are the ordered indices

in the UCS of the tab file and the observed states are the audio feature vectors. The HMM's transition probability distribution is defined in such a way that the transition from a state $e_i$ (i.e. the $i$th chord in the UCS) to a state $e_j$ is only valid if:

1. $i = j$ (i.e. the chord is repeated) or $i + 1 = j$ (i.e. we move to the next chord in the UCS);
2. $i < j$ and $i$ is the end and $j$ is the beginning of a line (i.e. we jump forward to a later line); or
3. $i > j$ and $i$ is the end and $j$ is the beginning of a line (i.e. we jump backward to an earlier line).

The probability of forward and backward jumps is dependent on the parameters $p_f$ and $p_b$, as defined by McVicar et al. (2011). We use the parameter setting $p_f = 0.05$ and $p_b = 0.05$.

The observed states of the HMM contain beat-synchronised chroma features from audio. For feature extraction, we use the Python package librosa (McFee et al., 2018). First, we convert the audio file to mono and use the HPSS function to separate the harmonic and percussive elements of the audio. Subsequently, we apply a constant-Q transform with a sampling rate of 22050 Hz and a hop length of 256 samples on the harmonic part. This step yields chroma features for each sample. Then, we beat-synchronise the features by running the beat-extraction function on the percussive part of the audio and averaging the chroma features between the consecutive beat positions. We beat-synchronise the chord annotations as well, by taking the most prevalent chord label between beats. Each mean feature vector with the corresponding beat-synchronised chord label is regarded as one frame. Now we have the feature vectors $X$ and chord labels $y$ for each song, which we use to train and test our HMM (using 10-fold cross-validation). Note that this is the only part of DECIBEL's MIDI and tab subsystems that requires training on the Isophonics annotations.

Tabs are often notated in a transposed key compared to the original audio file, because some keys are easier to play on the guitar than others. In order to correct for this, Jump Alignment considers the tab files in all 12 transpositions and selects for each tab file the transposition with the highest log-likelihood.

### *5.3 Tab file selection*

The results of Jump Alignment are shown in **Table 3**. Taking the average CSR of all tabs for a song (for all 199 songs for which there is at least one suitable tab file), the WCSR is 72.2%. If we select the best tab for each song, the WCSR is 78.5%. Since we do not know the CSR for unlabelled data, we select the expected best tab-file for each song based on the log-likelihood, following McVicar et al. (2011). By choosing the tab file with the highest log-likelihood for each song, the resulting WCSR improves to 75.0%. The distribution of CSRs of selected tab files compared to all tab files is also shown in **Figure 4**.

In this section, we have examined DECIBEL's tab subsystem. This subsystem consists of a parser that extracts UCSs from guitar tablature or chord sheets and Jump Alignment, which aligns these UCSs to the audio recording. When selecting the estimated best tabs for each song, the tab subsystem reaches a WCSR of 75.0%.

## 6. Integration Methods

DECIBEL estimates chord label sequences from different music representations, i.e. audio, MIDI and tab files, as we have seen in Sections 3.2, 4 and 5. This results in a set of chord label sequences for each song in our data set, which forms a rich harmonic representation. This representation is an interesting result already, since it can easily be visualised (see **Figure 5**). Such a visualisation contributes to a better understanding of an ACE algorithm's behaviour and the song's harmonic properties, as shown for the combination of MIDI and audio data by Ewert et al. (2012).

In order to test if DECIBEL improves current audio ACE systems, we need to combine these chord label sequences into one final sequence. DECIBEL achieves this using a

**Table 3:** WCSR of all songs, with different tab file selection methods.

| | **WCSR** |
|---|---|
| **Worst CSR of all tabs** | 59.0% |
| **Average CSR of all tabs** | 72.2% |
| **Best log-likelihood of all tabs** | **75.0%** |
| **Best CSR of all tabs** | 78.5% |

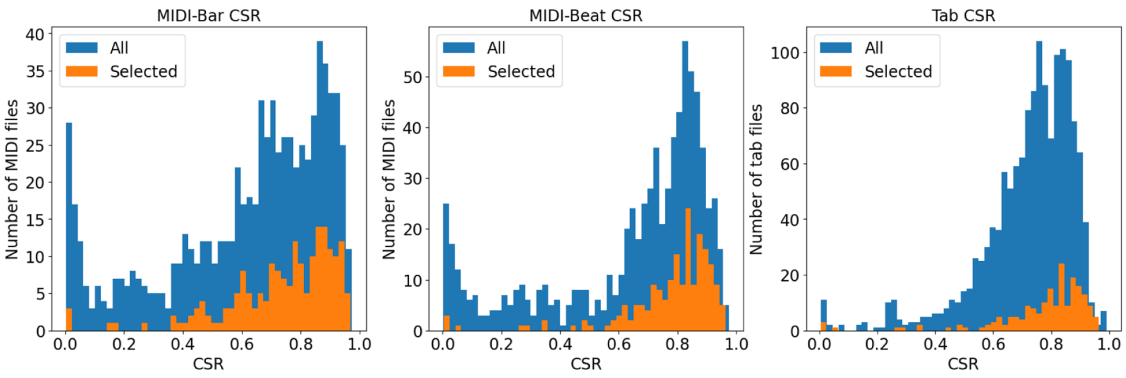


**Figure 4:** Histograms showing the distribution of CSR for: (Left) MIDI files with bar segmentation; (Centre) MIDI files with beat segmentation; and (Right) tab files.

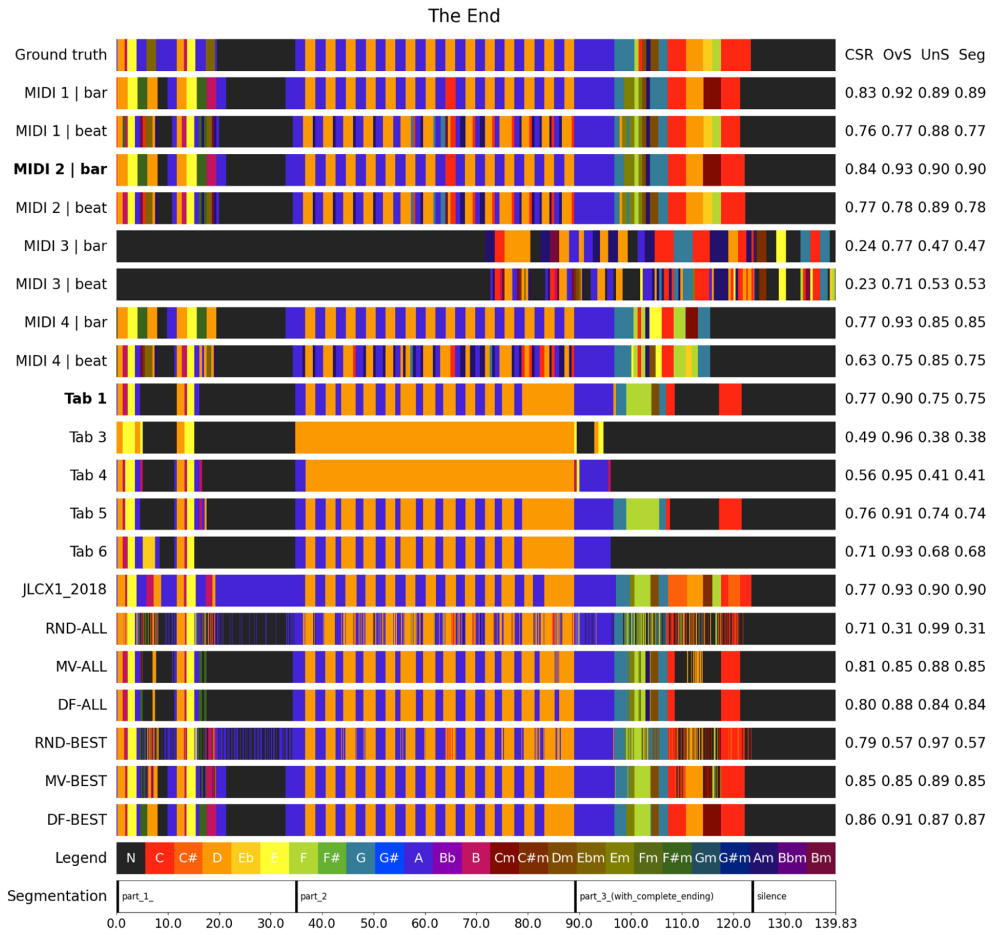


**Figure 5:** Visualisation of harmonic representation. The expected best MIDI file based on the average template similarity is MIDI 2 with bar segmentation (shown in boldface); the expected best tab file (based on log-likelihood) is Tab 1. In this song, the audio method (JLCX1) was unable to correctly classify the percussive section between 19.6 and 24.8s, whereas DF-BEST uses information from the MIDI and tab files to classify this as a no-chord. Also note that DF-BEST performs better than each of the individual sources (MIDI 2 bar, Tab 1 and JLCX1_2018).

data fusion step. In this section, we motivate the chosen data fusion method by an evaluation of various methods, based on Koops et al. (2016). In contrast to earlier work, DECIBEL needs to combine a varying number of sources per song. That is why we experimented with a selection strategy that selects only one source per representation. We compare two selection strategies in combination with three different integration methods. The two selection strategies are ALL and BEST: ALL takes the chord sequences of all tabs and MIDI files as sources. BEST only uses the sources of the expected best tab and MIDI file for each song, as described in Section 4.3 and 5.3. For one song, no MIDI file is selected, since all files were badly aligned; here, BEST only combines the audio and tab sequences. Similarly, BEST only combines the audio and MIDI sequence for the song without a suitable tab file.

The three integration methods are based on earlier work (Koops et al., 2016). We first divide each input chord sequence into 10 ms samples. Then we integrate the sources, selected using ALL/BEST, with either random picking (RND), majority voting (MV) or data fusion (DF). The implementations of RND and MV are unchanged compared to Koops et al. (2016): given a specific sample, RND takes the chord label of an arbitrary source, while MV assigns the chord label used by most of the sources.

Our implementation of the DF technique takes into account both the expected accuracy of sources and the probability of the labels provided by the sources. Let $\mathcal{X}$ be the set of samples, $V$ be the chord vocabulary, $S$ be the sources selected by the integration method and let $\mathcal{L}$: $\mathcal{S} \times \mathcal{X} \times \mathcal{V} \rightarrow \{0, 1\}$ be a labelling function such that $\mathcal{L}(s, x, v) = 1$ if source $s$ assigns chord $v$ to segment $x$ and 0 otherwise.

The source accuracy $A[s]$ is the probability that a source $s$ provides appropriate chords; the chord label probability $P[x, v]$ is the probability of chord label $v$ at segment $x$, given the chord labels of the sources. The chord label probability is defined based on the labelling $\mathcal{L}$ and the chord label vote count $VC[x,v]$. This value determines the influence of each source on the final label, based on its source accuracy. These values are iteratively computed as follows:

$$A[s] = \frac{\sum_{x \in \mathcal{X}} \sum_{v \in \mathcal{V}} P[x,v] \cdot \mathcal{L}(s,x,v)}{|\mathcal{X}|} \quad (1)$$

$$VC[x,v] = \sum_{s \in \mathcal{S}} \mathcal{L}(s,x,v) \cdot \ln \frac{(|\mathcal{V}|-1)A[s]}{1-A[s]} \quad (2)$$

$$P[x,v] = \frac{\exp(VC[x,v])}{\sum_{v' \in \mathcal{V}} \exp(VC[x,v'])} \quad (3)$$

$A[s]$, $VC[x,v]$ and $P[x,v]$ are defined in terms of each other. Therefore, we use an alternative equation for the initial computation of the chord label probability $P_0[x,v]$:

$$P_0[x,v] = \frac{\sum_{s \in \mathcal{S}} \mathcal{L}(s,x,v)}{|\mathcal{S}|} \quad (4)$$

The final chord labelling for DF is obtained by first computing Equation 4 and then updating Equations 1–3 five times. In the work of Koops et al. (2016) these computations are repeated until convergence; we empirically tested the number of required repetitions and fixed this number to five. Subsequently, we assign the chord label $v$ with the highest probability $P[x,v]$ to each segment $x$ to obtain the final DF chord sequence. See **Figure 6** for an example.

## 7. Results

We now compare the performance of each of the six combinations of integration methods and selection strategies (i.e. RND-ALL, RND-BEST, MV-ALL, MV-BEST, DF-ALL and DF-BEST), applied to each of our twelve audio ACE systems. Friedman tests (Friedman, 1937) for each of these twelve systems show that the integration methods and selection strategies give significantly different results in terms of CSR. Subsequently, we perform a Tukey's Honest Significant Difference post-hoc test (Tukey, 1949) to identify which integration-selection combinations are significantly different. The results for the Chordify audio system are visually represented in **Figure 7**. In this figure, differences are significant if the corresponding horizontal lines do not overlap. Both RND methods are significantly worse than the original audio ACE system (CHF). There is no significant difference between CHF and DF-ALL or MV-ALL. MV-BEST and DF-BEST perform significantly better than the original CHF system, with no significant difference between these two best-performing methods.

When examining the test results for all twelve audio ACE systems, we make four observations. *First*, the random picking integration method always performs worse than the original audio ACE system, regardless of

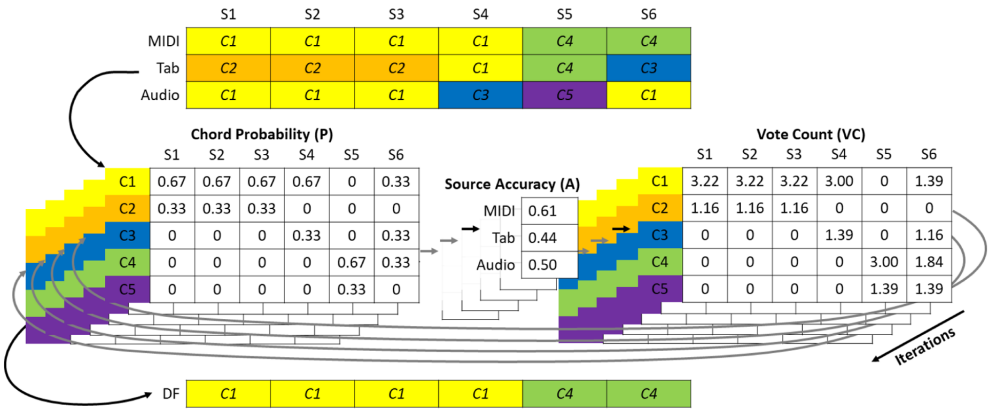


**Figure 6:** Toy example illustrating the data fusion procedure for a song consisting of six segments (S1 to S6). The input for the data fusion step consists of three sources: a MIDI, a tab and an audio file. In this example, we have a chord vocabulary of five chords (C1 to C5). First, the probability of each chord in each segment is computed. From this matrix, the source accuracies of the MIDI, tab and audio files are calculated. Then the computation of vote counts for each chord-segment pair is based on these source accuracies. After iterating these three steps, the result of data fusion is obtained by assigning the chord with the highest chord probability to each segment.

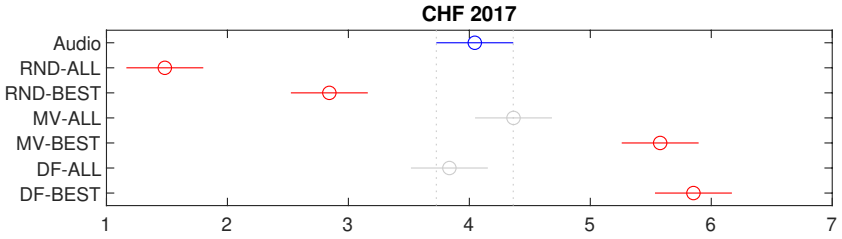


**Figure 7:** Visual representation of the differences in terms of Chord Symbol Recall between different data fusion methods, for the CHF audio ACE algorithm. For each pair of horizontal lines that do not overlap, the difference in CSR between the corresponding data fusion methods is significant. For example, DF-BEST is significantly better than DF-ALL, but the difference between DF-BEST and MV-BEST is not significant.

the chosen selection strategy. This is consistent with the findings of Koops et al. (2016). From this observation, we conclude that agreement between sources should be taken into account. *Second*, the BEST selection method always performs better than the ALL selection method. The explanation for this is as follows: given a poor MIDI or tab file, DECIBEL's MIDI or tab subsystem will find a poor chord label sequence. In the ALL selection method, this chord label sequence will, to a greater or lesser extent, be integrated in the final output sequence, which deteriorates the output sequence's quality, whereas the BEST selection method (hopefully) ignores this poor chord label sequence. *Third*, we observe that the difference between BEST and ALL is even larger for the DF integration method than for MV. A possible explanation for this is that chord label sequences from tabs are often undersegmented, for example because chord changes in instrumental parts or very short chords are often not detected. This is typical for tabs and therefore can occur for multiple (independent) tabs of the same song. Consequently, suboptimal chord sequences from tabs could get a high source accuracy in DF-ALL, and therefore substantially influence the final output sequence. DF-BEST only considers the expected best tab and MIDI file. If the expected best tab is undersegmented, it will probably get a low source accuracy, because chord sequences from MIDI and audio have less undersegmentation issues. The difference between MV-ALL and MV-BEST is smaller, as MV does not use source accuracies. Our *fourth* and final observation is that DF-BEST performs better than the original audio ACE system, yielding significant improvement for all tested audio methods except JLCX1 and JLCX2. In comparison, MV-BEST significantly improves all tested audio methods except FK2, KBK2, JLCX1 and JLCX2.

From these observations, we conclude that DF-BEST is the best selection-integration combination. **Table 4** compares the WCSR of each of the twelve audio ACE systems to DF-BEST applied to each system. Using DF-BEST improves ACE methods based on audio by 4.03% on average. The improvement is particularly clear for relatively weak methods, while ACE methods that are already strong are improved to a smaller extent. The columns DF-ALL MIDI WCSR, DF-ALL Tab WCSR, DF-BEST MIDI WCSR and DF-BEST Tab WCSR also show the results when combining the audio only with the (expected best) midi or tab file(s). From these columns it becomes clear that MIDI files contribute most to the improved performance, but adding tab files further lifts the performance. The final column of **Table 4** (DF-GT-BEST WCSR) shows the WCSR if DF is computed on the actual best MIDI and tab files. This gives an upper bound on the WCSR that can be obtained by improving MIDI/tab file selection methods.

Finally, we investigated the song-wise performance of DF-BEST compared to using only the original audio file. A comparison between the performances in terms of CSR distributions can be found in **Figure 8**. For the vast majority of songs in our data set, the chord label sequence found by DF-BEST was similar to or better than the sequence found by the original audio ACE system. However, we

**Table 4:** WCSR of audio ACE systems and DF-BEST. Note that two of the 2017 systems were resubmitted in MIREX 2018 and one system was also resubmitted in 2019. The Improvement column shows the improvement from DF-BEST compared to the audio-only method, where significant improvements are shown in boldface. Using DF-BEST improves ACE WCSR on average by 4.03%.

| Audio ACE | MIREX | Audio WCSR | DF-ALL MIDI WCSR | DF-ALL Tab WCSR | DF-ALL WCSR | DF-BEST MIDI WCSR | DF-BEST Tab WCSR | DF-BEST WCSR | Improvement | DF-GT-BEST WCSR |
|---|---|---|---|---|---|---|---|---|---|---|
| **CHF** | – | 82.0% | 81.3% | 76.2% | 80.3% | 83.5% | 78.2% | 84.6% | **2.6%** | 87.0% |
| **CM2/CM1** | '17–'19 | 75.7% | 80.4% | 75.9% | 79.7% | 80.9% | 76.6% | 81.6% | **5.9%** | 85.0% |
| **JLW1** | '17 | 79.0% | 80.6% | 76.0% | 79.7% | 82.4% | 77.5% | 83.0% | **4.0%** | 85.5% |
| **JLW2** | '17 | 78.5% | 80.6% | 76.0% | 79.7% | 82.2% | 77.3% | 82.7% | **4.2%** | 85.3% |
| **KBK1** | '17 | 82.8% | 81.5% | 76.5% | 80.5% | 84.1% | 78.5% | 85.3% | **2.4%** | 86.7% |
| **KBK2/FK2** | '17, '18 | 87.3% | 81.9% | 76.6% | 80.9% | 86.6% | 81.0% | 87.9% | **0.5%** | 89.2% |
| **WL1** | '17 | 79.9% | 80.8% | 75.9% | 79.8% | 82.5% | 77.3% | 83.4% | **3.6%** | 85.6% |
| **JLCX1** | '18 | 86.3% | 81.4% | 76.1% | 80.3% | 85.8% | 80.7% | 87.2% | 0.9% | 89.2% |
| **JLCX2** | '18 | 86.5% | 81.4% | 76.1% | 80.4% | 85.8% | 80.7% | 87.1% | 0.6% | 89.2% |
| **SG1** | '18 | 79.5% | 80.9% | 76.2% | 80.2% | 82.1% | 76.5% | 83.8% | **4.3%** | 86.1% |
| **CLSYJ1** | '19 | 77.3% | 80.3% | 75.6% | 79.6% | 81.6% | 75.6% | 83.0% | **5.7%** | 85.6% |
| **HL2** | '20 | 67.2% | 79.9% | 75.9% | 79.8% | 76.0% | 67.3% | 80.8% | **13.6%** | 84.2% |

CHF: Koops et al. (2017), CM2/CM1: Cannam et al. (2018), JLW1 and JLW2: Jiang et al. (2017), KBK1 and KBK2/FK2: Korzeniowski and Widmer (2016b), WL1: Wu et al. (2017), JLCX1 and JLCX2: Jiang et al. (2018), SG1: Gasser and Strasser (2018), CLSYJ1: Lee et al. (2019), HL2: Ku and Lee (2020).[6]

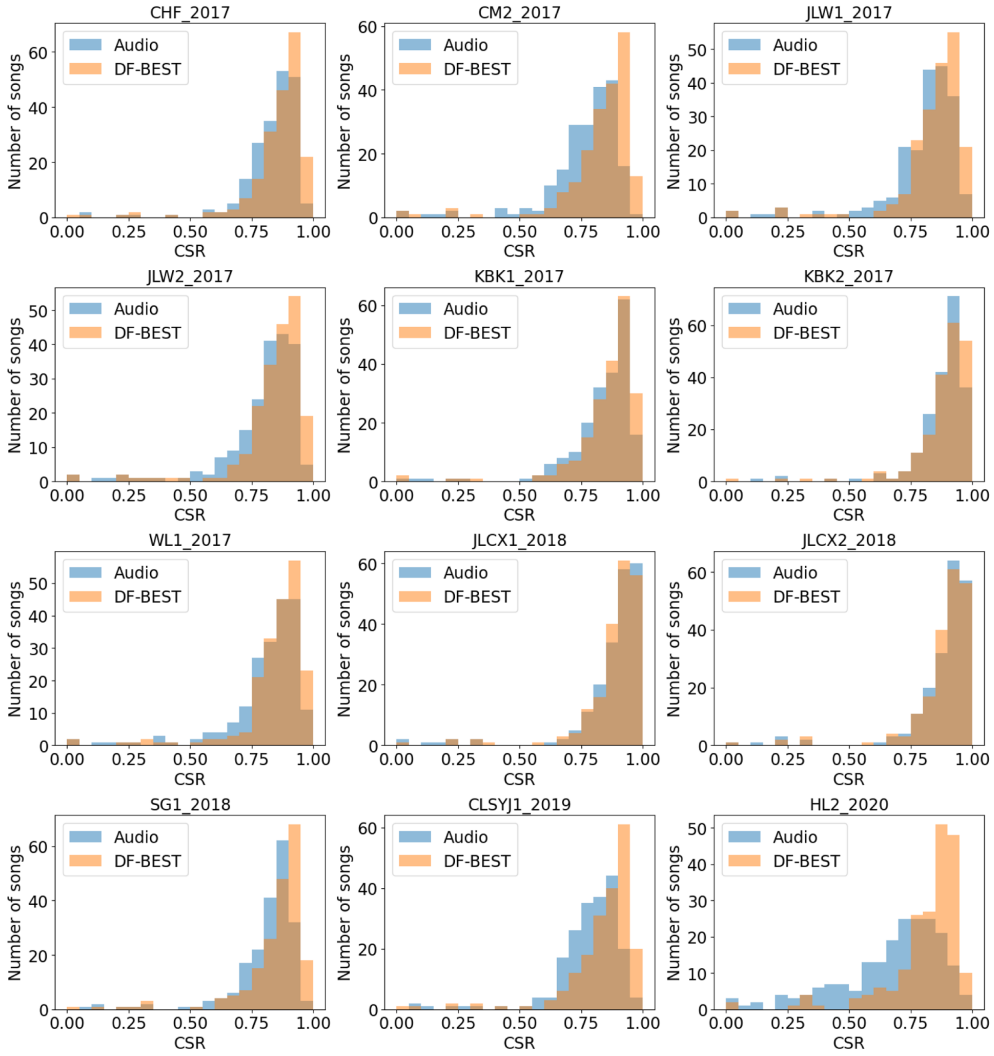


**Figure 8:** Distribution of Audio CSR and DF-BEST WCSR. For the audio algorithms CHF, CM2, JLW1, JLW2, KBK1, WL1, SG1, CLSYJ1 and HL2 the improvement is evident. No significant improvement was found for the algorithms JLCX1 and JLCX2. For KBK2, the improvement is mainly in the songs that already have a high CSR.

observed that DF-BEST performed consistently worse for the song *Let It Be* by The Beatles. For this song, our audio file was shifted compared to the audio file used in the MIREX competition. Consequently, the chord label sequence found by DF-BEST is shifted and therefore is not consistent with the Isophonics annotations. Another problematic song is *We Will Rock You* by Queen: this song starts with a lot of percussion and little harmonic content. Therefore, there are multiple different views on the chord labels for this song and the view selected by DF differs from the view adopted in the Isophonics annotations.

In this section we showed that DF-BEST (or MV-BEST) is the best combination of the tested selection strategies and integration methods. DF-BEST consistently performs better than the original audio algorithm in terms of WCSR.

## 8. Conclusion

We proposed DECIBEL, a novel method to improve ACE by aligning MIDI and tab files to audio recordings, using representation-specific chord estimation techniques to estimate chord sequences for each file. This way, we obtain a rich harmonic representation that reveals different views on the harmonic content of a song. Furthermore, DECIBEL integrates this harmonic representation into one final output sequence, combining representation-specific selection strategies (Section 4.3 and 5.3) with a representation-independent data fusion technique (Section 6).

DECIBEL uses relatively simple techniques, such as our basic MIDI chord estimation system CASSETTE. Apart from the audio-tab alignment method, these methods do not require any training and therefore do not overfit to ground truth data. Still, DECIBEL improves twelve state-of-the-art audio ACE systems in terms of WCSR, increasing the performance with 0.5 to 13.6 percentage points. We therefore conclude that the integration of multiple symbolic formats and audio is an interesting research direction for the improvement of ACE.

A first suggestion for future work would be to test DECIBEL's performance on a larger data set or on songs of another genre. Given that the performance of audio-only

algorithms on other (MIREX) data sets is typically lower than on the Isophonics data, we expect an even higher potential for DECIBEL to improve results for these data sets. Second, as Koops et al. (2016) observed that the performance of data fusion increases with larger chord vocabularies, we recommend extending DECIBEL with a larger chord vocabulary, in order to test how much the integration of symbolic formats helps in recognizing these complex chords. Third, we suggest experimenting with various alternative techniques for DECIBEL's subtasks, such as the consideration of alternative methods to extract chords from MIDI files. Following the argument of inherent heterogeneity and subjectivity of chord sequences by Koops et al. (2019); Ni et al. (2013), this paper addresses heterogeneity and subjectivity in chord representations, while it can be argued that the current one-to-one (e.g. MIREX) evaluation methods are not optimal for evaluating multiple reference annotations. Our fourth suggestion would therefore be to establish evaluation methods that take into account multiple reference annotations.

Symbolic formats offer a wealth of information. We hope to encourage the research community to exploit these valuable sources for ACE and other MIR tasks by making our implementation and documentation of DECIBEL available.

## Notes

[1] https://chordify.net/
[2] https://www.music-ir.org/mirex/
[3] https://www.ultimate-guitar.com/
[4] https://www.midiworld.com/
[5] https://github.com/DaphneO/DECIBEL
[6] https://github.com/ismir-mirex/ace-output/tree/master/2020/Isophonics2009/HL2

## Reproducibility

In order to stimulate further research on this interesting topic and enable reproducibility, we made our implementation of DECIBEL available on GitHub at https://github.com/DaphneO/DECIBEL; DOI: 10.5281/zenodo.5090853

## Acknowledgements

We thank Johanna Devaney as the action editor and three anonymous reviewers for their insightful comments.

## Competing Interests

Anja Volk is a co-Editor-in-Chief of the Transactions of the International Society for Music Information Retrieval. She was removed completely from all editorial processing. The authors have no other competing interests to declare.

## Author Contributions

Daphne Odekerken contributed to the overall design and execution of the study and the writing of the article. Hendrik Vincent Koops and Anja Volk supervised this work and contributed to the writing of the article. All authors have read and agreed to the published version of the manuscript.

## References

**Burgoyne, J. A., Pugin, L., Kereliuk, C.,** and **Fujinaga, I.** (2007). A cross-validated study of modelling strategies for automatic chord recognition in audio. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 251–254.

**Cannam, C., Benetos, E., Davies, M. E., Dixon, S., Landone, C., Levy, M., Mauch, M., Noland, K.,** and **Stowell, D.** (2018). MIREX 2018: VAMP plugins from the Centre for Digital Music. *Proceedings of the Music Information Retrieval Evaluation eXchange*.

**Chen, T.-P.** and **Su, L.** (2019). Harmony Transformer: Incorporating chord segmentation into harmony recognition. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 259–267.

**Cuvillier, P.** and **Cont, A.** (2014). Coherent time modeling of semi-Markov models with application to real-time audio-to-score alignment. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE. DOI: https://doi.org/10.1109/MLSP.2014.6958908

**Ewert, S., Müller, M., Konz, V., Müllensiefen, D.,** and **Wiggins, G. A.** (2012). Towards cross-version harmonic analysis of music. *IEEE Transactions on Multimedia*, 14(3):770–782. DOI: https://doi.org/10.1109/TMM.2012.2190047

**Friedman, M.** (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701. DOI: https://doi.org/10.1080/01621459.1937.10503522

**Gasser, S.** and **Strasser, F.** (2018). Multi objective chord estimation. *Proceedings of the Music Information Retrieval Evaluation eXchange*.

**Harte, C.** (2010). *Towards Automatic Extraction of Harmony Information from Music Signals*. PhD thesis, Department of Electronic Engineering, Queen Mary University of London.

**Humphrey, E. J.** and **Bello, J. P.** (2015). Four timely insights on automatic chord estimation. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, pages 673–679.

**Jiang, J., Chen, K., Li, W.,** and **Xia, G.** (2018). MIREX 2018 submission: A structural chord representation for automatic large-vocabulary chord transcription. *Proceedings of the Music Information Retrieval Evaluation eXchange*.

**Jiang, J., Li, W.,** and **Wu, Y.** (2017). Extended abstract for MIREX 2017 submission: Chord recognition using random forest model. *Proceedings of the Music Information Retrieval Evaluation eXchange*.

**Konz, V.** and **Müller, M.** (2012). A cross-version approach for harmonic analysis of music recordings. In Müller, M., Goto, M., and Schedl, M., editors, *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*, pages 53–72. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany.

**Koops, H. V., de Haas, W. B., Bountouridis, D.,** and **Volk, A.** (2016). Integration and quality assessment of

heterogeneous chord sequences using data fusion. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 178–184.

**Koops, H. V., de Haas, W. B., Bransen, J.,** and **Volk, A.** (2017). Chord label personalization through deep learning of integrated harmonic intervalbased representations. In *Proceedings of the First International Workshop on Deep Learning for Music*, pages 19–25.

**Koops, H. V., de Haas, W. B., Burgoyne, J. A., Bransen, J., Kent-Muller, A.,** and **Volk, A.** (2019). Annotator subjectivity in harmony annotations of popular music. *Journal of New Music Research*, 48(3):232–252. DOI: https://doi.org/10.1080/09298215.2019.1613436

**Korzeniowski, F.** and **Widmer, G.** (2016a). Feature learning for chord recognition: The Deep Chroma Extractor. In *Proceedings of 17th International Conference on Music Information Retrieval*.

**Korzeniowski, F.** and **Widmer, G.** (2016b). A fully convolutional deep auditory model for musical chord recognition. In *Proceedings of the 26th International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE. DOI: https://doi.org/10.1109/MLSP.2016.7738895

**Lee, S.-R., Chien, I., Yeh, T.-C.,** and **Jang, J.-S. R.** (2019). MIREX 2019 submission: Chord estimation. *Proceedings of the Music Information Retrieval Evaluation eXchange*.

**Li, X., Dong, X. L., Lyons, K., Meng, W.,** and **Srivastava, D.** (2012). Truth finding on the deep web: Is the problem solved? *Proceedings of the VLDB Endowment*, 6(2). DOI: https://doi.org/10.14778/2535568.2448943

**Macrae, R.** (2012). *Linking Music Metadata*. PhD thesis, Queen Mary University of London.

**Masada, K.** and **Bunescu, R.** (2019). Chord recognition in symbolic music: A segmental CRF model, segment-level features, and comparative evaluations on classical and popular music. *Transactions of the International Society for Music Information Retrieval*, 2(1). DOI: https://doi.org/10.5334/tismir.18

**Mauch, M.** (2010). *Automatic Chord Transcription from Audio Using Computational Models of Musical Context*. PhD thesis, Queen Mary University of London.

**Mauch, M., Cannam, C., Davies, M., Dixon, S., Harte, C., Kolozali, S., Tidhar, D.,** and **Sandler, M.** (2009). OMRAS2 metadata project 2009. In *Late-breaking Session at the 10th International Conference on Music Information Retrieval*.

**Maxwell, H. J.** (1992). An expert system for harmonic analysis of tonal music. In *Proceedings of the First Workshop on AI and Music*, pages 20–33. AAAI.

**McFee, B., McVicar, M., Balke, S., Thomé, C., Raffel, C., Lee, D., Nieto, O., Battenberg, E., Ellis, D., Yamamoto, R., Moore, J., Bittner, R., Choi, K., Friesch, P., Stöter, F.-R., Lostanlen, V., Kumar, S., Waloschek, S., Seth, Naktinis, R., Repetto, D., Hawthorne, C. F., Carr, C., Pimenta, W., Viktorin, P., Brossier, P., Santos, J. F., JackieWu, Erik, and Holovaty, A.** (2018). librosa/librosa: 0.6.1.

**McVicar, M., Ni, Y., Santos-Rodriguez, R.,** and **De Bie, T.** (2011). Using online chord databases to enhance chord recognition. *Journal of New Music Research*, 40(2):139–152. DOI: https://doi.org/10.1080/09298215.2011.573564

**McVicar, M., Santos-Rodríguez, R., Ni, Y.,** and **De Bie, T.** (2014). Automatic chord estimation from audio: A review of the state of the art. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(2):556–575. DOI: https://doi.org/10.1109/TASLP.2013.2294580

**Ni, Y., McVicar, M., Santos-Rodriguez, R.,** and **De Bie, T.** (2013). Understanding effects of subjectivity in measuring chord estimation accuracy. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(12):2607–2615. DOI: https://doi.org/10.1109/TASL.2013.2280218

**Pardo, B.** and **Birmingham, W. P.** (2002). Algorithms for chordal analysis. *Computer Music Journal*, 26(2):27–49. DOI: https://doi.org/10.1162/014892602760137167

**Pauwels, J., O'Hanlon, K., Gómez, E.,** and **Sandler, M.** (2019). 20 years of automatic chord recognition from audio. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 54–63.

**Preston, J. B., Pietras, M.,** and **Robinson, L.** (2012). Three views of the "musical work": Bibliographical control in the music domain. *Library Review*. DOI: https://doi.org/10.1108/00242531211292060

**Radicioni, D.** and **Esposito, R.** (2010). BREVE: An HMPerceptron-based chord recognition system. *Advances in Music Information Retrieval*, pages 143–164. DOI: https://doi.org/10.1007/978-3-642-11674-2_7

**Raffel, C.** and **Ellis, D. P.** (2014). Intuitive analysis, creation and manipulation of MIDI data with pretty_midi. In *15th International Society for Music Information Retrieval Conference Late Breaking and Demo Papers*, pages 84–93.

**Raffel, C.** and **Ellis, D. P.** (2015). Large-scale contentbased matching of MIDI and audio files. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, pages 234–240.

**Raffel, C.** and **Ellis, D. P.** (2016). Optimizing DTWbased audio-to-MIDI alignment and matching. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 81–85. IEEE. DOI: https://doi.org/10.1109/ICASSP.2016.7471641

**Raphael, C.** and **Stoddard, J.** (2004). Functional harmonic analysis using probabilistic models. *Computer Music Journal*, 28(3):45–52. DOI: https://doi.org/10.1162/0148926041790676

**Scholz, R.** and **Ramalho, G.** (2008). COCHONUT: Recognizing complex chords from MIDI guitar sequences. In *Proceedings of the 9th International Conference on Music Information Retrieval*, pages 27–32.

**Scholz, R., Ramalho, G.,** and **Cabral, G.** (2016). Cross task study on MIREX recent results: An index for evolution measurement and some stagnation hypotheses. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 372–378.

**Sheh, A.** and **Ellis, D. P.** (2003). Chord segmentation and recognition using EM-trained hidden Markov models. In *Proceedings of the 4th International Conference on Music Information Retrieval*, pages 183–189.

**Sigtia, S., Boulanger-Lewandowski, N.,** and **Dixon, S.** (2015). Audio chord recognition with a hybrid recurrent neural network. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, pages 127–133.

**Temperley, D.** and **Sleator, D.** (1999). Modeling meter and harmony: A preference-rule approach. *Computer Music Journal*, 23(1):10–27. DOI: https://doi.org/10.1162/014892699559616

**Tukey, J. W.** (1949). Comparing individual means in the analysis of variance. *Biometrics*, pages 99–114. DOI: https://doi.org/10.2307/3001913

**Wakefield, G. H.** (1999). Mathematical representation of joint time-chroma distributions. In *Proceedings of SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, volume 3807, pages 637–646. International Society for Optics and Photonics. DOI: https://doi.org/10.1117/12.367679

**Winograd, T.** (1968). Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory*, 12(1):2–49. DOI: https://doi.org/10.2307/842885

**Wu, Y., Feng, X.,** and **Li, W.** (2017). MIREX 2017 submission: Automatic audio chord recognition with MIDI-trained deep feature and BLSTM-CRF sequence decoding model. *Proceedings of the Music Information Retrieval Evaluation eXchange*.

**How to cite this article:** Odekerken, D., Koops, H. V., & Volk, A. (2021). Improving Audio Chord Estimation by Alignment and Integration of Crowd-Sourced Symbolic Music. *Transactions of the International Society for Music Information Retrieval*, 4(1), pp. 141–155. DOI: https://doi.org/10.5334/tismir.81

**Submitted:** 03 November 2020 **Accepted:** 05 July 2021 **Published:** 09 November 2021



]u[ *Transactions of the International Society for Music Information Retrieval* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS