A METHOD FOR ANALYSIS OF SHARED STRUCTURE IN LARGE MUSIC COLLECTIONS USING TECHNIQUES FROM GENETIC SEQUENCING AND GRAPH THEORY

Florian Thalmann[1] **Kazuyoshi Yoshii**[1] **Thomas Wilmering**[2]
Geraint A.Wiggins[3] **Mark B. Sandler**[2]
[1] Speech and Audio Processing Laboratory, Kyoto University
[2] Centre for Digital Music, Queen Mary University of London
[3] Artificial Intelligence Lab, Vrije Universiteit Brussel
thalmann.florian.2x@kyoto-u.ac.jp

ABSTRACT

While common approaches to automatic structural analysis of music typically focus on individual audio files, our approach collates audio features of large sets of related files in order to find a shared musical temporal structure. The content of each individual file and the differences between them can then be described in relation to this shared structure. We first construct a large similarity graph of temporal segments, such as beats or bars, based on self-alignments and selected pair-wise alignments between the given input files. Part of this graph is then partitioned into groups of corresponding segments using multiple sequence alignment. This partitioned graph is searched for recurring sections which can be organized hierarchically based on their co-occurrence. We apply our approach to discover shared harmonic structure in a dataset containing a large number of different live performances of a number of songs. Our evaluation shows that using the joint information from a number of files has the advantage of evening out the noisiness or inaccuracy of the underlying feature data and leads to a robust estimate of shared musical material.

1. INTRODUCTION

Automatic analysis of musical structure from audio is one of the more challenging tasks in music information retrieval (MIR) [1–3]. Reasons for this are the relatively high-level nature of the problem and its dependence on lower-level audio descriptors, which have a tendency to be noisy, as well as the restricted availability of annotated collections in a limited number of musical genres, which are necessary for tackling the problem with solutions based on machine learning. However, the growing number of large public and online music collections, which are usually annotated with user-curated metadata (song titles, artists, recording information, or dates), can potentially be used for unsupervised structural analysis which may be of considerable musicological value.

This paper introduces an approach for the detection of temporal structure in large collections of musical audio recordings where information from a number of related recordings is combined to improve the quality of results. From a given set of input audio recordings, e.g. different performances of the same song, our method identifies the most commonly occurring sequential structures, relates them to each other and organizes them hierarchically. The individual files can then be described, compared and aligned with each other by referencing this shared structure. Inspired by techniques used in genetic sequencing, we combine the use of different alignment methods, including dynamic programming (DP) and multiple sequence alignment (MSA) with graph representations and search methods. We evaluate our method on a subset of the Live Music Archive (LMA) of the Internet Archive and analyze the harmonic content of a large number of performances of a selection of songs. We compare the harmonic essence thus obtained with existing lead sheets and illustrate the differences between individual performances in a few qualitative comparisons. Although the examples in this paper focus on formal structure determined by harmonic progressions, the method can easily be used for structure determined by other musical aspects.

2. RELATED WORK

With the omnipresence of large digital audio collections of music, the automatic analysis of large corpora is becoming an increasingly central task in music information retrieval. Recent research in this area has mainly focused on large-scale statistical analysis of audio features [4–6], as well as making these accessible using interactive browsing tools [7–9]. While the analysis of temporal structure, including the identification of musical patterns, motifs, harmonic progressions, or form across corpora, are relatively well established for collections of symbolic music [10–12], only a few have attempted to use similar analysis methods for audio collections [13–15]. Potential reasons for this are often discussed and may be due to common problems with audio collections, including mislabeling, duplication, differing recording quality, or noisy audio features [4, 6, 16, 17].

Many of the methods used jointly in this paper have previously been applied in different musical contexts. Dynamic programming is often used to align audio recordings of different performances of the same material, either audio-to-score or audio-to-audio [18–20], sometimes considering musical variations and structural differences [21]. While immediate uses include score following, automatic accompaniment, and computer-assisted production, some have used alignment methods for classification tasks such as cover song identification [22] and plagiarism [23].

While the alignments methods most commonly used in MIR are pairwise, i.e. applied to align pairs of sequences, such as the Smith-Waterman or the Needleman-Wunsch algorithm [24], there are many methods for directly aligning multiple sequences commonly used in bioinformatics, but only a few have so far been applied to musical data. [25] used *multiple sequence alignment (MSA)* to eliminate conflicts and typos in song lyrics retrieved from the Web. [20] used their own progressive MSA method as well as *Profile HMMs* (Hidden Markov Models) to align different recordings of performances of classical music and found the two methods to lead to comparable results. In [26, 27], which comes closest to the present work, different MSA libraries were used along with pattern mining to detect harmonic patterns in symbolic transcriptions of a set of 138 songs. The authors were able to identify cover songs as well as genre clusters with their most characteristic progressions.

There are generally three common approaches to automatically discovering musical structure in sequences of feature vectors from individual recordings, via repetition, novelty, or homogeneity [1]. The first identifies repeating subsequences whereas the other two identify abrupt changes or comparatively stable areas. Sections often reappear in slightly varied forms, which has been addressed in [28]. Recent methods often use combined approaches using both harmonic and timbre features in order to improve results, e.g. [29]. While music is inarguably organized hierarchically [2], few approaches enable the detection of hierarchical structure [30].

Our approach is purely repetition-based, however, not only in individual recordings, but in the entire collection of interest. This allows the identification of sections occurring only once in a given piece and leads to more robust descriptions of the found sections. Similar to our approach, [31] used automatically detected musical structure to improve chord label quality for individual pieces.

3. METHOD

Our method consists of a five-step process. Given a set of recordings, we create an alignment graph based on a number of self-alignments and pairwise alignments. Then, we use an MSA to partition the alignment graph and create a structure graph. We then search the structure graph for commonly occurring sections and classify them into section types. These section types are then grouped into a hierarchical structure, based on their co-occurrence. Finally, we complete the structure graph with material from the individual recordings that was left out by the MSA and annotate the recordings with section types. Figure 1 shows an overview of the process.

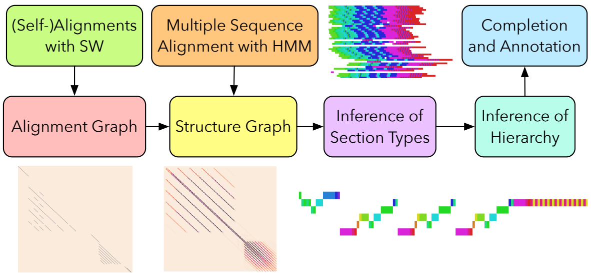
(Self-)Alignments with SW
Multiple Sequence Alignment with HMM
Completion and Annotation
Alignment Graph
Structure Graph
Inference of Section Types
Inference of Hierarchy

Figure 1. Overview of the five-step process.

3.1 Alignment Graph

Given a collection A of K related audio recordings we obtain a *feature sequence* $A^k = (a^k_1, ..., a^k_{N_k})$ of length N_k for each recording $k = 1, ..., K$. Each element a^k_i represents a time segment, such as a beat, bar, or onset, with corresponding feature information, such as chroma vectors or chord labels.

A *local alignment* between two such sequences A^s, A^t is commonly defined as a sequence of pairs $p = (p_1, ..., p_L)$ with $p_l = (s_l, t_l) \in [1, ..., N_s] \times [1, ..., N_t]$ with a monotonicity constraint $1 \leq s_1 \leq \ldots \leq s_{N_s} \leq N_s$ and $1 \leq t_1 \leq \ldots \leq t_{N_t} \leq N_t$ and a step condition $p_{l+1} - p_l \in \{(0, 1), (1, 0), (1, 1)\}$. A local *self-alignment* can be defined accordingly with $A^s = A^t$.

In our situation it is advantageous to only consider *diagonal local alignments* in order to reduce ambiguity and noise in the alignment graph. We can achieve this with a more strict step condition $p_{l+1} - p_l = (1, 1), \forall l = 1, \ldots, L$-1.

Due to repetition and variation in the given musical material, many sensible local alignments may exist between each pair of recordings. For example, if in one recording the first of a pair of musical sections is expanded or another is inserted between the two sections, two independent local alignments are still able to capture the commonality between the two recordings. The same is true for self-alignments, which are able to characterize repetition at different temporal intervals within recordings.

From a large number of such alignments and self-alignments we can then create an *alignment graph* $G_A = (N_A, E_A)$ for collection A with a node for each segment a^k_i and an edge between each aligned segment pair p_l. Due to the alignments being local, not every node in the graph is necessarily connected, and some nodes may have many incident edges.

Due to the large size of many audio collections of interest and the time complexity of alignment algorithms, it may not be feasible to calculate the alignments between every possible pair of recordings. However, for our method it has proven to be sufficient to select a small subset of all possible pairings, e.g. n random pairings per recording ($n * K$ alignments) plus all K self-alignments.

3.1.1 Alignment and Self-Alignment Methods Used

Common approaches to alignment are usually designed to find a single global or local alignment for a pair of given sequences. For the reasons outlined above we are more interested in finding multiple local diagonal alignments of reasonable length. Many common approaches can be modified for this purpose. Here we discuss the *Smith-Waterman* algorithm, which we use in our experiments.

Smith-Waterman is a dynamic programming algorithm developed in the context of genetic sequence alignment [32]. For two given input sequences A^s, A^t, the simplest variant with a linear gap penalty generates a scoring matrix $H_{i,j}$ with dimension $N_s + 1 \times N_t + 1$ and with $H_{i,0} = H_{0,j} = 0$. Each $H_{i,j}$ with $i, j > 0$ is then determined as follows:

$$H_{ij} = \max \begin{cases} H_{i-1,j-1} + sim(a_j^s, a_j^t), \\ H_{i-1,j} - P_G, \\ H_{i,j-1} - P_G, \\ 0 \end{cases} \qquad (1)$$

where sim is a similarity function between feature vectors and P_G is a gap penalty.[1] Depending on the nature of the feature vectors, one may choose sim to be a simple cosine similarity, or a function that returns a match score for identical vectors and else a lower mismatch score.

Starting from the highest score in the matrix, the algorithm then finds the most likely alignment path by tracing back the origins of the score and ending at a position with a score of 0. Our modification of the algorithm finds diagonal paths by limiting trace-back to matching pairs containing a maximum number of $\gamma \geq 0$ subsequent diagonal gaps (mismatches). With one iteration of Smith-Waterman, several of these paths may be extracted by gradually removing found paths from the matrix and setting their values to 0.

Furthermore, due to the fact that some potential paths can be covered up by higher-rated paths nearby, we propose an *iterative variant* of Smith-Waterman where every element in the neighborhood of a previously found alignment path p is set to zero, i.e. $\forall p_l \in p$ we set $H_{ij} = 0$ for $s_l - \delta \leq i \leq s_l + \delta$ and $t_l - \delta \leq j \leq t_l + \delta$. The parameter $\delta \geq 0$ controls the minimum distance between alignments, which can be used for limiting the number of results.[2]

Additional ways of improving the performance of the algorithm and the quality of the resulting alignments that have proven useful are limiting the number of iterations, setting a score threshold below which alignments are no longer considered, or setting a minimum segment length.

Many of the existing methods can also be modified for *self-alignment*. With Smith-Waterman, we simply treat the trivial diagonal alignment as a previously found path p. Figure 2 shows an example diagonal self-alignment.

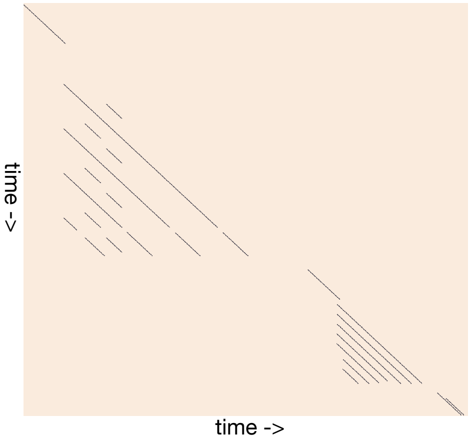

Figure 2. The matrix resulting from a diagonal self-alignment of a recording of *China Doll* from the dataset used in Section 4 (10 longest segments).

3.2 Structure Graph

The next step is to construct a *structure graph* that encapsulates the most common structural characteristics found in the given collection. The goal is to identify a large subgraph G'_A of G_A that can be partitioned into a sequence of partitions $P_1, \ldots, P_M$ of corresponding nodes in G'_A, i.e. $P_m \subset N_A$. Each partition contains at most one segment of each recording, i.e. $k \neq l$, $\forall a_i^k, a_j^l \in P_m$, and the partitions are strictly ordered temporally, i.e. $i < j$, $\forall a_i^k \in P_m$, $a_j^l \in P_{m+1}$. For example, if the nodes of G_A represent bars in A, each partition contains bars of different recordings that can be considered equivalent. Subsequent partitions may be thought of as recurring sequences, although there may be gaps in individual or all recordings between to adjacent partitions. Note that $\cup P_m$ may likely not include all nodes of G_A due to significant structural differences between the individual recordings, hence the definition of G'_A. Figure 3 shows the connection matrix of an example partition where the z-axis shows the number of connections between partition pairs.

We can infer such a partition directly from the connections in G_A in an iterative manner. We experimented with various graph search approaches, e.g. searching for the most densely connected components with at most one node per recording and aligning these components temporally, or with beam search with various partition improvement and modification methods based on functions that rate the clarity of the connection matrix of the partition.

However, while these methods work well for relatively similar input sequences, we obtained better and faster results for more diverse input material with multiple sequence alignments using *Profile HMMs* [24], a method common in genetic sequencing where a reasonable simultaneous alignment is found for all sequences. A Profile HMM is a particular type of Hidden Markov Model with three types of states and a given length L. *Match states* M_j

[1] In genetic sequence alignment it is generally more appropriate to replace P_G with a gap penalty function that distinguishes between opening and closing a gap and decreases for longer gaps.

[2] A similar iterative ('recursive') variant was introduced in [19]. However, there each sequence element is involved in at most one pair.

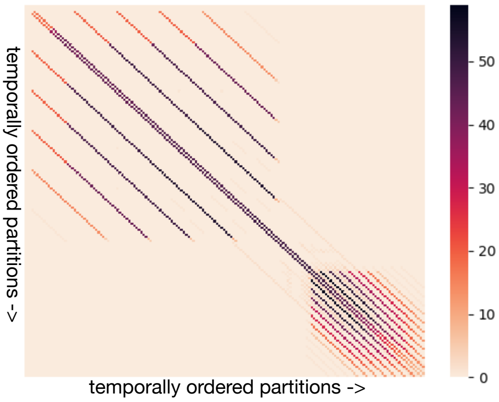

Figure 3. Connection matrix of the partitioned alignment graph of *China Doll* (65 recordings). The z-axis shows the number of connections between the nodes of each partition pair. The beginning of the song features a regularly repeating section (verses and solos), followed by a section that occurs only once but in most versions (interlude and chorus), followed by a short repeated ostinato (outro).

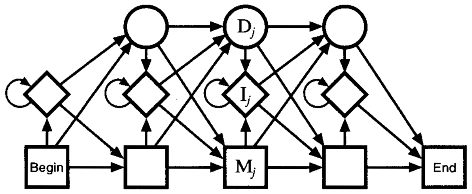

Figure 4. Transition structure of a Profile HMM [24].

represent segments shared between sequences, *insert states* I_j represent possibly multiple consecutive segments particular to an individual sequence, and *delete states* D_j represent segments missing in a particular sequence (the segment represented by the corresponding match state), where $j = 1, \ldots, L$ (see Figure 4).

There are L match and L delete states, as well as $L + 1$ insert states. The emission distributions for the match and insert states are chosen depending on the input feature sequences, e.g. 12-dimensional multivariate gaussian for chroma vectors, or multinomial discrete distributions for chord labels. The model is trained with a number of related sequences, in our case the A^k, using the expectation maximization variant of Baum-Welch. L is usually chosen based on the lengths of the input sequences, e.g. their maximum, median, mean, or minimum length. Individual state sequences for each input sequence can be decoded using the Viterbi algorithm, i.e. we obtain a state label for each a_i^k. We can then infer the M partitions from the segments associated with the L match states ($M \leq L$), for example by only keeping the match states that appear in a proportion of at least $0 \leq \lambda \leq 1$ of all the sequences.

We now define a *structure graph* $G''_A = (N''_A, E''_A)$ whose nodes correspond to the partitions P_m and whose edges are determined by the most common mutual connections in G'_A between the elements of different partitions. More precisely, we add an edge for each node pair $P_m, P_n \in N''_A$ where $P_m \in con(P_n, \mu)$ and $P_n \in con(P_m, \mu)$. The function $con(x, \mu)$ returns the set of $\mu > 0$ nodes in N''_A most strongly connected to x:

$$con(x, \mu) = \underset{y \in N'_A \setminus x \quad \mu}{\operatorname{argmax}} \; |\{e \in E_A \mid a, b \in e, a \in x, b \in y\}| \tag{2}$$

where $\operatorname{argmax}_\mu$ returns the set of μ arguments for which the function is maximized. The parameter μ should be relatively small for best results, e.g. $0 < \mu \leq 5$. For illustration, the resulting graph is a pruned simple-graph version of the multigraph of which a connection matrix is shown in Figure 3. The pruned graph contains only nodes representing significantly large partitions are kept and simple edges are established wherever the multigraph has many edges.

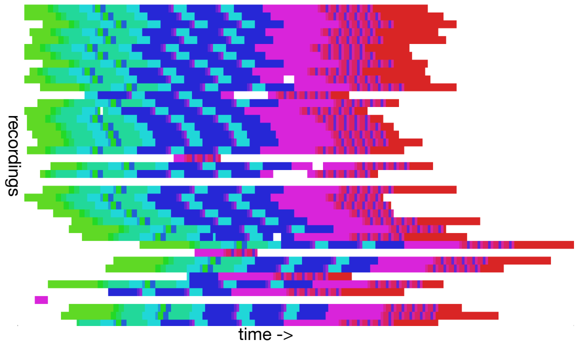

Figure 5. Juxtaposition of different recordings of *China Doll* where colors represent segment types. The different horizontal offsets illustrate varying lengths of introductory tuning or announcements. The empty lines are mislabeled recordings of different songs.

3.3 Inference of Section Types and Hierarchies

The structure graph can then automatically be decomposed in order to find *types of sections* recurring in the collection and within single recordings. The connected components C_j in G''_A represent sets of equivalent partitions of segments recurring at different points in time. We sort these components by their lowest partition index $\min_m(P_m \in C_j)$ and group temporally adjacent ones where $P_m \in C_j \iff P_{m+1} \in C_{j+1}$ into sequences. For each of these sequences we can retrieve the corresponding recurrent sections by simply transposing the two-dimensional arrays of indexes, i.e. $(C_{j1}, \ldots, C_{jJ})^T$ with $j1, \ldots, jJ$ being the sorted indexes of the components in a given group. Finally, we merge temporally adjacent groups of sections G, H if for each section in G there is a directly temporally adjacent section in H and vice versa. Figure 5 shows a visualization of the section types thus obtained for the partitioned graph shown in Figure 3.

For music with no recurring sections the above procedure may result in only a few or no section types. We therefore suggest an additional step of inferring boundaries between adjacent connected components C_j, C_{j+1}. Let $\Delta_{j,j+1} = avg_k(n_k - m_k)$ be the average difference in index over all recordings k appearing C_j, C_{j+1}, where m_k, n_k are the indexes of the segments of k, i.e. $a_{m_k}^k \in C_j, a_{n_k}^k \in C_{j+1}$. If $\Delta_{j,j+1} \geq \tau$ for a given thresh-

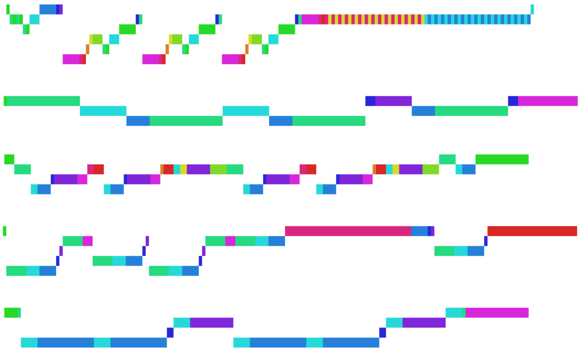

Figure 6. Visualization of the structural hierarchies of five different songs from the dataset from Section 4.1. The topmost is the unflattened hierarchy of *China Doll* (Figures 3 and 5), all others are flattened. The colors indicate section types, the vertical axis nesting level.

old $\tau > 1$, we introduce a section boundary between C_j and C_{j+1}. For example, for $\tau = 5$ we add a section boundary between any two subsequent components where there are on average 4 segments missing per recording.

Finally, we can simplify the structure by inferring a *hierarchy* from the obtained section types. This can be done using a simple recursive search method that identifies the most frequently recurring adjacent section types and either combines them into a new type or concatenates them if they always co-occur. This hierarchy can then be simplified by further merging adjacent types that always appear together, and finally flattening nested types if their parts only occur within them. Figure 6 shows a few visualizations of example hierarchical structures.

3.4 Annotation of Individual Structures

In a final step of the process, each individual recording in the collection can be annotated using the found shared structure. First, we label each segment in the structure graph with a corresponding section type identified as described in the previous section. Then, using the alignment graph G_A we can infer section types for segments that are not in the structure graph, which may for example be the case if some recordings contain additional repetitions of sections. We consider for each segment a_i^k in $G_A \setminus G'_A$ and check which partition P_m in G''_A it is most strongly connected to, i.e. which partition contains the most segments connected to a_i^k. Note that some segments, sections, or entire recordings may remain unlabeled, if they were not aligned or self-aligned in the first step of the process (Section 3.1), due to being entirely unrelated or their features being too noisy (see Figure 5 for some examples).

Finally, we can annotate each segment that received a section type with a *feature value* derived from the shared structure. We determine a value for each position of every section type by summarizing the original features of all segments associated with that position. For example, we may label the first beat of a section type with the chord label most frequently occurring among all associated beats. All corresponding segments in individual recordings can then be annotated with that label.

4. EXPERIMENTS

We tested our approach on material from the Grateful Dead collection of the Live Music Archive, [3] which holds more than 13,000 recordings of over 2,000 shows spanning the years 1965 to 1995. The large number of recordings of individual songs and the improvised nature of Grateful Dead's performances make this collection particularly interesting for our work.

4.1 Dataset and Preparations

We created a *dataset* [4] with all performed versions of 15 songs from this collection, selected based on the criteria that a large number of versions exist and that a corresponding studio recording by the Grateful Dead is available that could potentially be used as a reference in the future. The fact that these recordings are live recordings poses additional challenges to the ones outlined in Section 2. Many of them contain a considerable amount of crowd noise which may lead to noisy audio features, and most of these recordings were made by amateurs using their analog tape equipment, which means that many of them are out of tune due to varying tape speed. We addressed the second of these problems and *resampled* the audio files after comparing their rotated chroma features with the ones of the respective studio version. For a ground truth, we *transcribed* the chord progressions beat-by-beat and grouped them into bars and sections for each of the songs with the help of existing lead sheets. [5] This level of granularity is particularly important due to the fact that many of these songs are based on odd meters (e.g. 7/4 in *Estimated Prophet*) or contain metrical changes (e.g. abbreviated 2/4 bars in *China Cat Sunflower*). Tuning ratios, a script for downloading and resampling, and transcriptions are published with the dataset.

The experiments described here [6] are based on a *subset* of the dataset with at most 100 versions of every song. We extracted triadic chord features using [33] (root notes and one of the four qualities major, minor, diminished, and augmented) and summarized them to beats extracted using Madmom. [7] The summarization process is based on the statistical mode of the chords in each temporal segment, i.e. for each segment the chord that was played for the longest. In order to be comparable with the features, we simplified the transcribed chords to triads as well.

We used our own implementation of a Profile HMM and initialized it with $L =$ median input length and with uniform distributions and transition probabilities, except match-match 0.999 and delete-insert 0.01. Our Smith Waterman implementation led to the quickest and best results

[3] https://archive.org/details/GratefulDead

[4] https://github.com/grateful-dead-live/fifteen-songs-dataset

[5] e.g. at http://jdarks.com/GDTab.html

[6] Code available at https://github.com/florianthalmann/ismir2020-shared-structure

[7] https://madmom.readthedocs.io

	(a) baseline	(b) annotated	(c) shared
p_G	0.691	0.779	0.825
p_O	0.411	0.461	0.482

Table 1. Proportion of matched chords in groundtruth and output for baseline (extracted chords), annotated recordings, and shared harmonic structure (averaged over all recordings in the case of baseline and annotated).

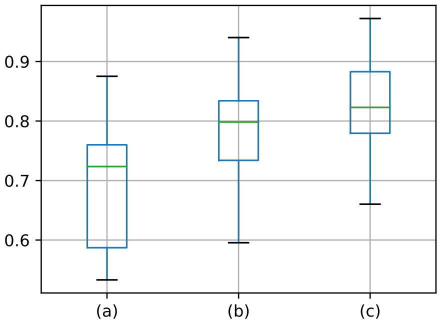

Figure 7. Distributions of average proportions of matches in ground truth p_G per song. (a) baseline, (b) annotated versions, (c) shared structure.

with the following parameter settings: a single iteration, 10 longest alignment segments, minimum segment length 16, $\gamma = 4$, $\delta = 4$, $\lambda = .1$, $\mu = 1$, $\tau = 2$.

4.2 Results

Due to the fact that there is no previous work with which to compare our method, we chose to perform an evaluation similar to [31] where structural information is used to improve chord prediction accuracy. [8] We used Smith Waterman to align the following sets of sequences with the ground truth transcriptions: (a) the original extracted chord sequences as a baseline (b) the sequences annotated by our method according to Section 3.4 and (c) the shared harmonic structure identified by our method according to Section 3.3. We then calculated two measures for each of the sets of sequences: the proportion of correctly matched ground truth segments p_G as well as the proportion of correctly matched segments in the output p_O, i.e.

$$p_G = \frac{\text{matches in alignment}}{\text{length of groundtruth}}, \; p_O = \frac{\text{matches in alignment}}{\text{length of output}} \quad (3)$$

Table 1 shows the overall values and Figure 7 shows distributions of p_G per song. p_O is lower than p_G due to for example additional repetitions of sections in performances or the high degree of variation and improvisation in many of the songs, i.e. there are deviations from the 'lead sheet' in individual recordings. However, the fact that on average the shared harmonic structure matched with the lead sheet content with an average probability of 82.5% is promising.

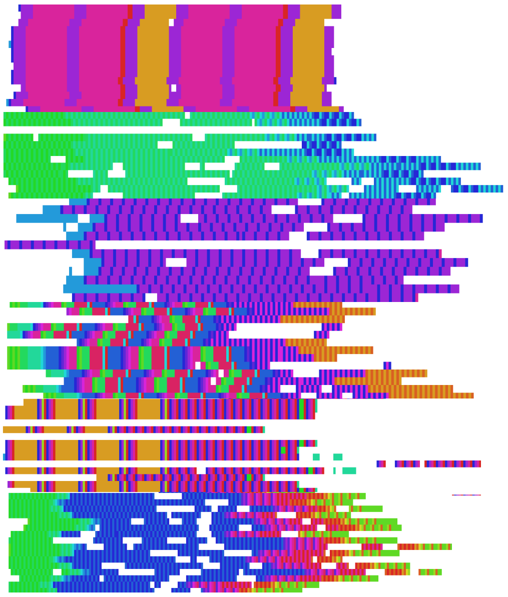

Figure 8. Visualisation of the segment types for 6 songs (*Box of Rain*, *Eyes of the World*, *Franklin's Tower*, *Sugar Magnolia*, *Casey Jones*, and *China Cat Sunflower*), around ten recordings each.

4.3 Application

As a more qualitative investigation of the potential of our approach we created a simple Web application for the interactive exploration of annotations and alignments along with the underlying recordings. Users can hear the corresponding segments of the recordings by clicking on the colored blocks. Figure 8 compiles six screenshots for different songs that illustrate different shared structures and individual deviations from them. Whereas *Box of Rain* has a very even and simple $AABAAB$ structure with tiny insertions, other songs, such as *Eyes of the World* or *Franklin's Tower*, feature longer more open-ended yet highly repetitive sections. *Casey Jones* is a combination of both with four verse/chorus repetitions followed by an extended jam over part of the chorus.

5. CONCLUSION

We have presented a new method for the extraction of shared temporal structure from a number of related audio recordings and shown with both quantitative and qualitative results how such a method could be useful. For example, it could provide musicologists a way to systematically study and explore larger archives of related recordings, or yield more reliable estimates of audio features for noisy live music recordings. Besides a more extensive evaluation and application, future work could include an expansion of the method for joint use of different kinds of feature vectors, either to improve the inference of sections and hierarchies for less repetition-based music (analogous to other approaches to structure inference) or for a more multidimensional analysis of the musical material.

[8] Note that instead of evaluating the hierarchical structures, which necessitates a non-trivial generalization of the method suggested in [34] and will be done in future work, we evaluate the flattened annotations, which nevertheless result from the process described in sections 3.1 through 3.4.

6. ACKNOWLEDGEMENTS

This work is supported by JST ACCEL No. JPMJAC1602, JSPS KAKENHI Nos. 16H01744 and 19H04137, as well as EPSRC Grant EP/ L019981/1.

7. REFERENCES

[1] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis." in *ISMIR*. Utrecht, 2010, pp. 625–636.

[2] M. Müller, "Music structure analysis," in *Fundamentals of Music Processing*. Springer, 2015, pp. 167–236.

[3] M. Giraud, R. Groult, and F. Levé, "Computational analysis of musical form," in *Computational Music Analysis*. Springer, 2016, pp. 113–136.

[4] N. Collins, "The ubuweb electronic music corpus: an mir investigation of a historical database," *Organised Sound*, vol. 20, no. 1, pp. 122–134, 2015.

[5] M. Mauch, R. M. MacCallum, M. Levy, and A. M. Leroi, "The evolution of popular music: Usa 1960–2010," *Royal Society open science*, vol. 2, no. 5, p. 150081, 2015.

[6] K. R. Page, S. Bechhofer, G. Fazekas, D. M. Weigl, and T. Wilmering, "Realising a layered digital library: exploration and analysis of the live music archive through linked data," in *Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries*. IEEE Press, 2017, pp. 89–98.

[7] A. Porter, M. Sordo, and X. Serra, "Dunya: A system for browsing audio music collections exploiting cultural context," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, 2013.

[8] S. Abdallah, E. Benetos, N. Gold, S. Hargreaves, T. Weyde, and D. Wolff, "The digital music lab: A big data infrastructure for digital musicology," *Journal on Computing and Cultural Heritage (JOCCH) - Special Issue on Digital Infrastructure for Cultural Heritage, Part 1*, vol. 10, no. 1, 2017.

[9] A. Allik, F. Thalmann, and M. Sandler, "Musiclynx: Exploring music through artist similarity graphs," *WWW '18 Companion Proceedings of the The Web Conference 2018, Geneva, Switzerland*, 2018.

[10] O. Lartillot, "Efficient extraction of closed motivic patterns in multi-dimensional symbolic representations of music," in *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*. IEEE, 2005, pp. 229–235.

[11] D. Temperley and T. d. Clercq, "Statistical analysis of harmony and melody in rock music," *Journal of New Music Research*, vol. 42, no. 3, pp. 187–204, 2013.

[12] D. Meredith, "Analysing music with point-set compression algorithms," in *Computational Music Analysis*. Springer, 2016, pp. 335–366.

[13] T. Collins, S. Böck, F. Krebs, and G. Widmer, "Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

[14] M. Barthet, M. D. Plumbley, A. Kachkaev, J. Dykes, D. Wolff, and T. Weyde, "Big chord data extraction and mining," in *Proceedings of the 9th Conference on Interdisciplinary Musicology – CIM14*, 2014.

[15] J. Van Balen, J. A. Burgoyne, D. Bountouridis, D. Müllensiefen, and R. C. Veltkamp, "Corpus analysis tools for computational hook discovery," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[16] B. Sturm, "An analysis of the GTZAN music genre dataset." *Proceedings of the 2nd international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, 2012.

[17] T. Wilmering, G. Fazekas, S. Dixon, K. Page, and S. Bechhofer, "Towards high level feature extraction from large live music recording archives," *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML), 6-11 July, Lille, France*, 2015.

[18] S. Dixon and G. Widmer, "Match: A music alignment tool chest." in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, 2005, pp. 492–497.

[19] S. Ewert, M. Muller, V. Konz, D. Mullensiefen, and G. A. Wiggins, "Towards cross-version harmonic analysis of music," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 770–782, 2012.

[20] S. Wang, S. Ewert, and S. Dixon, "Robust and efficient joint alignment of multiple musical performances," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2132–2145, 2016.

[21] M. Grachten, M. Gasser, A. Arzt, and G. Widmer, "Automatic alignment of music performances with structural differences," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, 2013.

[22] J. Serra, E. Gómez, P. Herrera, and X. Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1138–1151, 2008.

[23] C. Dittmar, K. F. Hildebrand, D. Gärtner, M. Winges, F. Müller, and P. Aichroth, “Audio forensics meets music information retrieval—a toolbox for inspection of music plagiarism,” in *2012 Proceedings of the 20th European signal processing conference (EUSIPCO)*. IEEE, 2012, pp. 1249–1253.

[24] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[25] P. Knees, M. Schedl, and G. Widmer, “Multiple lyrics alignment: Automatic retrieval of song lyrics.” in *ISMIR*. Citeseer, 2005, pp. 564–569.

[26] P. Esling and M. G. Bergomi, “Molecular clock synthesis,” IRCAM, http://repmus.ircam.fr/esling/projet-atiam-2014.html, Tech. Rep., 2015.

[27] M. G. Bergomi, “Dynamical and topological tools for (modern) music analysis,” Ph.D. dissertation, Universià degli Studi di Milano; Universié Pierre et Marie Curie, 2015.

[28] M. Müller and F. Kurth, “Towards structural analysis of audio recordings in the presence of musical variations,” *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, p. 089686, 2006.

[29] G. Shibata, R. Nishikimi, E. Nakamura, and K. Yoshii, “Statistical music structure analysis based on a homogeneity-, repetitiveness-, and regularity-aware hierarchical hidden semi-markov model,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[30] B. McFee and D. Ellis, “Analyzing song structure with spectral clustering.” in *ISMIR*, 2014, pp. 405–410.

[31] M. Mauch, K. C. Noland, and S. Dixon, “Using musical structure to enhance automatic chord transcription.” in *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, 2009, pp. 231–236.

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

[33] Y. Wu, T. Carsault, and K. Yoshii, “Automatic chord estimation based on a frame-wise convolutional recurrent neural network with non-aligned annotations,” in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[34] B. McFee, O. Nieto, M. M. Farbood, and J. P. Bello, “Evaluating hierarchical structure in music annotations,” *Frontiers in psychology*, vol. 8, p. 1337, 2017.