ALIGNED SUB-HIERARCHIES: A STRUCTURE-BASED APPROACH TO THE COVER SONG TASK

Katherine M. Kinnaird
Data Sciences Initiative and Division of Applied Mathematics
Brown University, USA
katherine_kinnaird@brown.edu

ABSTRACT

Extending previous structure-based approaches to the song comparison tasks such as the fingerprint and cover song tasks, this paper introduces the *aligned sub-hierarchies* (AsH) representation. Built by applying a post-processing technique to the aligned hierarchies of a song, the AsH representation is the set of unique aligned hierarchies for repeats (called AH^{R}) encoded in the original aligned hierarchies of the whole song. Effectively each AH^{R} within AsH is a section of the aligned hierarchies for the original song. Like aligned hierarchies, the AsH representation can be embedded into a classification space with a natural metric that makes inter-song comparisons based on sections of the songs. Experiments addressing a version of the cover song task on score-based data using AsH as the basis of inter-song comparison demonstrate potential of AsH-based approaches for MIR tasks.

1. INTRODUCTION

A common starting point in music information retrieval tasks is the creation of visualizations and representations for music-based data streams. One of the most influential representations is Foote's self-similarity matrix (SSM) [4], which continues to be one of the most recognizable images in MIR. Much of the work starting with an SSM or a self-dissimilarity matrix (SDM) like [1, 5, 7, 9–12] create post-processing techniques that seek to enhance certain properties. This paper also offers a new post-processing technique that can be applied to either the SSM or SDM and extends the work of [7].

Under the music comparison tasks like the fingerprint task and the cover song task, structure-based approaches like [5, 7] seek to compare songs via their whole song representations. While this type of approach has varied success, there can be obvious issues. For example, with more rigid comparisons such as [7], whole song comparisons may only be meaningful if the songs have the same number of time-steps. Similarly whole song comparisons can fall victim to large artistic choices. For example, [5] created a smoothed image of the thresholded and resampled SDM, which was then compared using the Euclidean distance. While effective, this approach stumbled comparing recordings of Mazurka Op. 68 No. 4 where Chopin neglected to include a *fine* marking, causing some pianists to play the piece twice, while others played the piece once [5].

The contrast to the whole song approach is using sections of a song. In [2, 3], audio shingles representing sections of recordings are compared to address the fingerprint, cover song, and remix tasks. In [17], the fingerprint task is tackled by comparing sections of recordings' constellation maps marking their spectrogram peaks. Both approaches require access to the original audio signal.

This work introduces the aligned sub-hierarchies (AsH), a structure-based representation that can be used to compare songs based on sections of the songs. This AsH representation exists between the section-based comparison approaches like [2, 3, 17] and the structure-based approaches in [5, 7]. Furthermore, this representation embeds into a classification space with a natural metric that observes the triangle inequality.

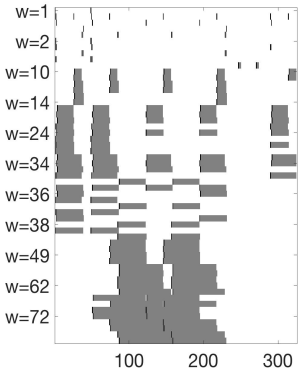
The paper is organized as follows. Section 2 motivates the necessity of the extension of aligned hierarchies into AsH representation in context of MIR tasks. In Section 3, we formalize the definition of the AsH representation, detail the construction of AsH, and describe embedding AsH into a classification space with a natural metric. In Section 4, we use AsH representations to perform experiments for a version of the cover song task on a set of Mazurka scores. We offer future directions for research in Section 5.

2. MOTIVATION AND BACKGROUND

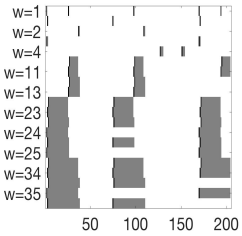
The aligned sub-hierarchies (AsH) representation is an extension of the aligned hierarchies from [7] that is motivated and inspired by making comparisons based on sections of songs or musical scores like those of [2,3,17]. This new representation seeks to combine the strengths of [5, 7] while addressing their limitations. Like their predecessor, AsH embeds into a classification space with a natural metric, but unlike in [7], the metric for AsH allows comparison between songs of differing lengths. Inspired particularly by the work in [5] stumbling on comparisons where some artists chose to repeat a song in its entirety, AsH seeks to note whether two songs share sections of unique structural

decompositions at the original scale of those sections. In other words, in contrast to [5], there is no resampling in the formulation nor in the comparison of AsH.

Before detailing the AsH representation, we will present a summary of the aligned hierarchies introduced in [7]. This structure-based visualization catalogues all meaningful repeats present in music-based data streams, showing all possible structural hierarchies aligned on one common time axis. Each aligned hierarchies representation H comprises of three components: an onset matrix B_H, a length vector w_H, and an annotation vector α_H. There are as many rows in these three components as there are kinds of repetitions present in a music-based data stream, with each row being tied to a particularly type of repeat. The binary matrix B_H encodes when each repeat begins with a entry of 1 at each starting time step. The vectors w_H and α_H together act as a key for B_H, while the former encodes the lengths of each type of repeat, and where the latter assigns annotations for the types of repeats so that types of the same length have different annotations. We can visualize the information contained in $H = (B_H, w_H, \alpha_H)$ as shown in Figures 1 and 3.1(b).



(a) Aligned hierarchies visualization for the version of the score observing Chopin's repeat signs. One segmentation of this version is AABABACA.



(b) Aligned hierarchies visualization for the version of the score ignoring the repeat signs as if Chopin never wrote them. One segmentation of this version is ABACA.

Figure 1. Aligned hierarchies for versions of Chopin's Mazurka Op. 6, No. 1 score, under threshold $T = 0.02$, with shingle size 12. Grey blocks denote repeats and rows denote types of repeats. Vertical labels are a subset of repeat lengths, while the horizontal ones mark the time steps.

The aligned hierarchies also provide an approach to the fingerprint task as they can be embedded into a classification space. However, nuanced comparisons can only be made between two aligned hierarchies with the exact same number of time steps [7]. This rigidity makes aligned hierarchies-based comparisons for the cover song task inappropriate since two cover songs are unlikely to be the same length. For certain MIR tasks, like the cover song task, we instead would like to compare parts of the songs' aligned hierarchies to each other, making comparisons based on these smaller aligned hierarchies a more suitable choice. This is the motivation and the inspiration for AsH. Take for example Figure 1 showing two versions of Chopin's Mazurka Op. 6, No. 1 score: Figure 1(a) shows the aligned hierarchies for the score as Chopin intended and Figure 1(b) shows the aligned hierarchies for the score with Chopin's repeat signs ignored. Under one version of the cover song task, we would like to identify these two versions as being based off the same score, and just comparing these two aligned hierarchies, we notice similarities between the sections of the shown structural hierarchies.

3. ALIGNED SUB-HIERARCHIES

This section introduces both the aligned sub-hierarchies (AsH) representation and the classification space that AsH representations embed into. Starting with the aligned hierarchies for a song or a musical score, we isolate different repeated patterns and find the individual aligned hierarchies for each isolated pattern. The collection of the unique aligned hierarchies for sections of music-based data stream is called the *aligned sub-hierarchies*, abbreviated to *AsH*.

AsH is the result of a post-processing technique on aligned hierarchies that is similar to the result of treating sections of a song as songs in their own right and then finding each section's aligned hierarchies. Like other comparison methods like [2, 3, 17] that compare sections of songs to each other, the AsH finds all possible structural hierarchies for sections of each song. By leveraging structural information already encoded in aligned hierarchies for the whole song, we have already found all the repeated sections with smaller repeats within them and potentially keep additional structure information that would have been hidden had we build an aligned hierarchies directly from the song's section.

3.1 Defining AsH

In this section, we will formally define the AsH representation and consider a motivating example. The below definition for the aligned hierarchies of a given repeat R_i^k, particularly the third condition, ensures that we capture all possible song sections with their own aligned hierarchies.

Definition 3.1. Consider a song and let H be the aligned hierarchies for the song. Let R_i^k be a repeat of length k beginning at time step i, encoded in H. Then the set of repeats meeting the following three conditions form the *aligned hierarchies* of R_i^k or $\text{AH}^{R_i^k}$:

1. Encoded in H that are of length less than k,
2. Contained in the set of time steps $[i, (i+k-1)] \cap \mathbb{N}$,
3. Have at least one corresponding repeat that is also contained within the time steps $[i, (i+k-1)] \cap \mathbb{N}$.

We encode $\text{AH}^{R_i^k}$ as $h_{R_i^k} = \left(B_H|_{R_i^k}, w_H|_{R_i^k}, \alpha_H|_{R_i^k}\right)$, where each component of $h_{R_i^k}$ is defined similarly to those in $H = (B_H, w_H, \alpha_H)$. Here, the onset matrix $B_H|_{R_i^k}$ has k columns, encoding repeats in $h_{R_i^k}$ by by their relative position to the start of R_i^k. For a general repeat (without a specified start or length), we say *aligned hierarchies of a repeat* and shorten to AH^{R}.

Example 3.1. Consider a song with the thresholded distance matrix $\mathcal{T}$ shown in Figure 3.1(a). This song has three

kinds of structure: A (which occurs four times), B (which occurs five times), and C (which occurs only once). The aligned hierarchies for the song is shown in Figure 3.1(b).

We can build $\text{AH}^{R_1^{30}}$ from the aligned hierarchies by considering the blocks contained within the beats 1-30. In this case, those structures are the (ABA) structure on beats 1-25; the (BAB) structure on beats 11-30; the (AB) structure on beats 1-15 and 16-30; the (BA) structure on beats 11-25; the B structure on beats 11-15 and 26-30; and the two A structures on beats 1-10 and 16-25. Given that the A, B, and (AB) structures each repeat within beats 1-30, then by Definition 3.1, $\text{AH}^{R_1^{30}}$, encoded into $h_{R_1^{30}}$, contains these structures. A visualization for $h_{R_1^{30}}$ is shown in Figure 3.1(c).

Since the notion of time for $\text{AH}^{R_i^k}$ is relative to beat i, then for the Example 3.1, we have that $\text{AH}^{R_{36}^{30}}$ will encode the same information as $\text{AH}^{R_1^{30}}$. So $h_{R_1^{30}} = h_{R_{36}^{30}}$.

Definition 3.2. Let H be the aligned hierarchies for a song. The unordered set of *unique* AH^{R} representations denoted $\{h\} = \{h_1, h_2, \ldots, h_m\}$ where each $h_i \in \{h\}$ is the AH^{R} for at least one repeat encoded in H is the *aligned sub-hierarchies* (or AsH) of the song.

Example 3.2. Consider Example 3.1 shown in Figure 3.1. The repeats (BAB), (ABA), $(ABAB)$ each have an AH^{R}. Although each of these occur twice, the AsH representation is comprised of only *unique* AH^{R} representations. So $\{h\} = \left\{h_{(BAB)}, h_{(ABA)}, h_{(ABAB)}\right\}$.

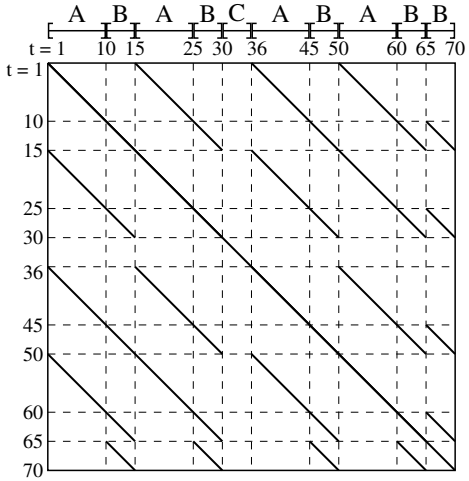
3.2 Building AsH from Aligned Hierarchies

In this section, we explain crafting $\{h\}$, the AsH representation of a song, from aligned hierarchies. First we detail constructing $\text{AH}^{R_i^k}$ for each repeat encoded in H and then explain when $\text{AH}^{R_i^k}$ is added to the AsH representation.

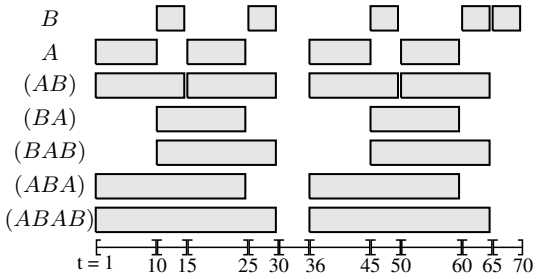
For each repeat in H, we note the starting time step i and the length of the repeat k, and then form $\text{AH}^{R_i^k}$ as follows. We first isolate the rows of $B_H \in H$ that correspond to repeats smaller than width k (that is the rows of B_H associated to the entries of $w_H < k$), and we further restrict this matrix to only the columns i through $(i + k - 1)$. We call the resulting sub-matrix $B_H|_{R_i^k}$. We also form the associated vectors $w_H|_{R_i^k}$ and $\alpha_H|_{R_i^k}$ as the entries of w_H and α_H that correspond to the rows of $B_H|_{R_i^k}$.

To satisfy Definition 3.1 as we build $\text{AH}^{R_i^k}$, we remove the rows of $B_H|_{R_i^k}$ that contain fewer than two repeats and then remove the corresponding entries in both $w_H|_{R_i^k}$ and $\alpha_H|_{R_i^k}$. Removing entries in $\alpha_H|_{R_i^k}$ may require adjusting the resulting values in $\alpha_H|_{R_i^k}$ so that for each repeat length k, the clusters of repeats of length k stored in $B_H|_{R_i^k}$ are identified with integers 1 through κ (that is, the number of clusters of repeats of length k stored in $B_H|_{R_i^k}$).

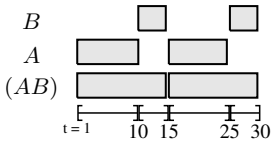
The resulting triple $h_{R_i^k} = \left(B_H|_{R_i^k}, w_H|_{R_i^k}, \alpha_H|_{R_i^k}\right)$ is the $\text{AH}^{R_i^k}$ representation. We next check if $h_{R_i^k}$ is already in $\{h\}$, the unordered list of unique $\text{AH}^{R_i^k}$. If $h_{R_i^k} \notin \{h\}$, then $h_{R_i^k}$ is added to $\{h\}$.



(a) $\mathcal{T}$ for the toy song with sections marked.



(b) Visualization for H of the toy song.



(c) Visualization for $h_{(ABAB)}$, which can be either denoted as $h_{R_1^{30}}$ or $h_{R_{36}^{30}}$, as they are equivalent.

Figure 2. Visualizations for toy song example with segmentation ABABCABABB.

By construction, a song's AsH can have AH^{R} of differing widths. If $h_i \in \{h\}$ is the AH^{R} for repeat R_i^k, then h_i is of width k.

3.3 Method for Comparing AsH

Comparing two AsH representations to each other requires finding the best alignments between collections of AH^{R}. This comparison must also respect that the AsH is an unordered set of AH^{R} representations and thus needs to be invariant to shifts in the ordering of the AH^{R}s. The AsH representation can be embedded into a space with a natural notion of distance, and this embedding leverages the embedding of the aligned hierarchies from Section 3 of [7].

3.3.1 Embedding AsH

As each AH^{R} is an aligned hierarchies representation, we start by embedding each AH^{R} into $(\mathcal{S}^*)^n$, the classification space for aligned hierarchies. To do this, a sequence of

binary matrices is created, where the k^{th} matrix is the rows of the aligned hierarchies associated to repeats of length k. The space $\mathcal{S}^*$ is defined as $\mathcal{S}/\sim$, where $\mathcal{S}$ is the space of $(m \times t)-$binary matrices and where $\sim$ is the equivalence relationship encoding that two matrices are equivalent if they are row permutations of each other.

In the case of AsH, we have a collection of AH$^{\text{R}}$s each of which can be represented as an element of $(\mathcal{S}^*)^n$. By treating each $h_i \in \{h\}$ as a column of information, we consider p copies of $(\mathcal{S}^*)^n$. In this sense, we have a product space comprised of $(n \times p)$-copies of the space $\mathcal{S}^*$, arranged into n rows and p columns, exactly like the entries of a $(n \times p)$-matrix, with each element of AsH occupying a column of this matrix-like layout.

Since AsH is an unordered collection of AH$^{\text{R}}$s, we need to define a space that is invariant to the ordering of the elements of AsH. We define the relationship $\sim_*$ on $(\mathcal{S}^*)^{(n\times p)}$ such that two AsH representations are equivalent under $\sim_*$ if they are different orderings of the same set of AH$^{\text{R}}$ representations. We can show that $\sim_*$ is an equivalence relation on $(\mathcal{S}^*)^{(n\times p)}$, which allows us to define the following:

Definition 3.3. Let $\mathcal{P}$ be the quotient space $(\mathcal{S}^*)^{(n\times p)}/\sim_*$. For $\{A\} \in \mathcal{P}$, the i-th column is A_i. We write $A_i \in \{A\}$ and call A_i a *column* of $\{A\}$.

The AsH representation $\{h\}$ of a song can be represented as an element of $\mathcal{P}$ where $h_i \in \{h\}$, the i-th AH$^{\text{R}}$, gets placed in the i-th column and so $\{h\} \in \mathcal{P}$. This space $\mathcal{P}$ encodes the invariance of the ordering of the AH$^{\text{R}}$s in an AsH representation, due to the equivalence relation $\sim_*$, meaning that the AH$^{\text{R}}$ (for a given AsH representation) can be placed into $\mathcal{P}$ in any order.

3.3.2 Metric on $\mathcal{P}$

To compare two songs via AsH, we find the pairs of the AH$^{\text{R}}$ from the first song with those from the second song that minimize the sum of the distances between the pairs. We first consider the AH$^{\text{R}}$s within the AsH representations that are of a fixed length k. Then we add all the distances from the identified matchings across the possible values of k. The resulting sum encodes the total dissimilarity between the repeated patterns of all sizes present in all of AH$^{\text{R}}$s contained within the two AsH representations.[1]

To first compare AH$^{\text{R}}$s of the same length, consider two AsH representations $\{A\} = \{A_1, A_2, \ldots, A_q\}$ and $\{B\} = \{B_1, B_2, \ldots, B_r\}$, with all AH$^{\text{R}}$s of length k. Assuming that $q, r \in \mathbb{Z}_{\geq 0}$ and that $q \geq r$, and ensuring that we are comparing lists of the same lengths, we append $(q-r)$ empty AH$^{\text{R}}$s to $\{B\}$, each of which is a row-vector of k zeros. We recall that $d_H : (\mathcal{S}^*)^n \times (\mathcal{S}^*)^n \to \mathbb{R}$ is the distance between two aligned hierarchies encoding the total dissimilarity between them.

Below, we define f_L that permutes the elements of $\{B\}$ to find the optimal matching of AH$^{\text{R}}$s in $\{B\}$ to those in $\{A\}$. This sum of the distances between the pairs of AH$^{\text{R}}$s is the minimum across all possible matchings.

[1] The proofs for the material in this section can be found in the author's doctoral thesis [6].

Proposition 3.1. *Let $\{A\}, \{B\} \in \mathcal{P}$. Let S_p be the symmetric group of degree p. Define $f_L : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ as*

$$f_L(\{A\}, \{B\}) = \min_{\sigma \in S_p} \sum_{i=1}^{p} d_H\left(A_i, B_{\sigma(i)}\right)$$

Then the function $f_L : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ is a distance function.

Proof Sketch: Leveraging properties of the symmetric group and the fact that d_H is a distance function, it is a straight forward check of the four requirements of a distance function: non-negativity, observance that the distance between two objects is zero if they are equivalent, reflectivity, and obeying the triangle inequality.[2] □

While f_L is a notion of total dissimilarity between two sets of AH$^{\text{R}}$s all of the same fixed length, a song's AsH representation likely contains AH$^{\text{R}}$s of differing lengths. To find the total dissimilarity across all possible lengths of AH$^{\text{R}}$s within AsH representations we add up the f_L distances found across all values of k. For clarity, we use the following definitions and notation:

Definition 3.4. Let $\{A\} \in \mathcal{P}$ such that the AH$^{\text{R}}$s of $\{A\}$ are not necessarily the same width. Define $\{A^k\}$ to be the AH$^{\text{R}}$s of $\{A\}$ that are of width k.

Corollary 1. *Let $\{A\}, \{B\} \in \mathcal{P}$. Let M be the largest width of the AH$^{\text{R}}$s in $\{A\}$ or $\{B\}$. Let $d_P : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ be given by:*

$$d_P(\{A\}, \{B\}) = \sum_{i=1}^{M} f_L\left(\{A^i\}, \{B^i\}\right).$$

Then d_P is also a distance function.

Proof Sketch: Using that f_L is a distance function, check that d_P satisfies the definition of a distance function. □

We note that the comparison of two songs using AsH requires that both songs have AsH representations. Due to the fact that we require each AH$^{\text{R}}$ to be the aligned hierarchies for a section of the song, it is possible that there exists aligned hierarchies for a song, but not one AH$^{\text{R}}$. This would happen when all sections within a song do not have smaller repeated structures that repeat within that section. In this case, the song has an *empty AsH representation* and note that no comparisons can be made between a song with an empty AsH representation and any other song.

4. COVER SONG EXPERIMENTS

To test the validity of the using AsH representation and its associated metric to approach MIR tasks, we apply the AsH-based comparison method to address a version of the cover song task for a score-based data set.[3] Each experiment follows the below procedure:

[2] We can further prove that if we first find and remove exact matches for the AH$^{\text{R}}$s in the two AsHs, then the distance between the remaining AH$^{\text{R}}$s in $\{A\}$ and $\{B\}$ is the same as the distance between the full AsHs $\{A\}$ and $\{B\}$. This fact adds efficiency to the computation of $f_L(\{A\}, \{B\})$. This proof proceeds by induction on the number of unmatched exact matches between $\{A\}$ and $\{B\}$.

[3] The code used for these experiments as well as those in [7] can be found at `https://github.com/kmkinnaird/ThesisCode/releases/tag/vT.final2`

1. Pre-process the songs by building audio shingles using s concatenated beat synchronous Chroma feature vectors, and then creating and thresholding an SDM using a global threshold T
2. Construct the aligned hierarchies as in [7]
3. Extract AsH representations from each aligned hierarchies representation
4. Compute pairwise distances between pairs of AsH representations under the d_P metric
5. Match two songs if they are mutual nearest neighbors of each other
6. Evaluate the resulting matchings compared to the ground truth using precision and recall scores

4.1 Score-Based Data Set

For the experiments in this work, we use a score-based data set comprised of 52 Mazurka musical scores by Chopin. For each score, we download two **kern files posted on the KernScore online database [4] (see [13]). The first file observes the repeated signs as marked by Chopin in the score, while the second ignores these repeat signs as if they are not written at all. If a score has no marked repeat signs, we download the single **kern file twice, marking one copy as observing the repeat signs and the second copy as having the repeat signs ignored. We refer to the versions of the scores as songs.

In this data set, each time step is in terms of beats with one time step being equivalent to one beat. We use the `music21` Python library [5] to extract the Chroma feature vectors for each beat in the song. For each time step, we encode local information by creating the audio shingles (like those in [2, 3]) that are s concatenated Chroma feature vectors, for a fixed integer s. As most Mazurkas have 3 beats per measure, in these experiments (like those in [7]), we set $s = 6$ or $s = 12$. This means that we encode two or four (three beats) bars into each audio shingle.

We then create $\mathcal{D}$, the SDM for each song, by computing cosine dissimilarity measure between all pairs of audio shingles. So for audio shingles a_i, a_j associated to time steps i and j respectively, we define

$$\mathcal{D}_{i,j} = \left(1 - \frac{\langle a_i, a_j \rangle}{||a_i||_2 ||a_j||_2}\right).$$

Then each SDM is thresholded based on the chosen global threshold T, which denotes how similar two audio shingles must be to be considered repetitions of each other. In these experiments, we choose global thresholds that are associated to very small differences between collections of 3 to 5 notes. To make this choice, we used the framework presented in [8] to connect our choice of T to the number of additions to the C-maj chord one can make and still be considered a repeat of the C-maj chord under the threshold T. This thresholding method differs from those in the literature that choose a threshold based on a fixed percentage of pairwise measures to be selected such as [1, 5, 11] or based on a fixed number of nearest neighbors as in [14–16].

We complete the processing of each song by extracting the associated aligned hierarchies from the thresholded SDM as done in [7]. To find the AsH representation for each song, we apply the post-processing steps outlined in subsection 3.2. The AsH is the basis for our inter-song comparison in the following experiments.

4.2 Experimental Setup

For this work, we define the cover song task as matching the score's **kern file with the repeat signs observed to the score's **kern file that ignore the repeat signs. After finding the AsH for each song, we compute the pairwise distances between the songs' representations using d_P, and store the results in a pairwise distance matrix $\mathbb{D}$. We then perform a mutual nearest neighbor matching, by treating each song as a query track. Therefore each experiment has a maximum of 104 possible matches as each song as another version of its score to match with.

For these experiments, the ground truth is the song list with their cover as given by the meta data of each **kern file. We compute precision and recall by comparing the experiment's resulting matches to the ground truth.

4.3 Results

Table 1 reports experimental results for $s \in \{6, 12\}$ and $T \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$. The 10 experiments have high precision rates but more modest recall rates. Since not all songs have an AsH representation for each value of s and T, the number of possible matches varies.

For each experiment, we note the number of feature vectors per audio shingle, s, and the threshold value T that determines when two time steps are said to be repeats of each other. The choice of s and T affects the number of non-zero entries in the thresholded SDM, which determines whether or not a song has an AsH representation. If a song does not have an AsH associated to it, then we remove the row and column in $\mathbb{D}$ associated to that song from consideration as well as remove that song from the ground truth listing. Our computations for precision and recall are based the adjusted ground truth list. In addition to reporting the precision and recall values for each experiment, we also report the number of possible matches that could be made based on the choices of s and T.

4.4 Discussion

The above results demonstrate the usability of the AsH representation in approaching MIR tasks. What is more, these results expose both strengths and weaknesses of using AsH as the basis for inter-song comparisons.

In considering the above results, we note that this version of the cover song task differs from the typical presentation of the cover song task. Most cover songs follow the original composer's intended structure fairly closely. In these experiments, half of our data set closely follows the composer's intentions while the other half blatantly makes

[4] `http://kern.humdrum.org/search?s=t&keyword=Chopin`
[5] `http://web.mit.edu/music21/`

s	T	Possible Matches	Precision Rate	Recall Rate	Empty AsHs
6	0.01	62	1	0.774	35
	0.02	66	0.962	0.757	33
	0.03	74	0.931	0.730	26
	0.04	78	1	0.692	24
	0.05	88	1	0.727	15
12	0.01	34	1	0.882	68
	0.02	44	1	0.818	57
	0.03	46	0.85	0.739	54
	0.04	56	0.84	0.75	45
	0.05	62	0.929	0.839	39

Table 1. Results for AsH for score data set on 10 experiments varying s, the width of the audio shingles, and T, the threshold used on the SDM for each song

large changes to the structure of the pieces. It is uncommon (though not unheard of as discussed in [5]) to see such large structural changes between two recordings of the same piece, which is what makes this version of the cover song task more challenging. Under this version of the cover song task, the AsH-based method was able to achieve strong experimental results.

The data in these experiments differs from the typical data set for the cover song task. For this score data, there are only two natural versions of each score: one with the repeat signs observed and the second with the repeat signs ignored. This means that every song has exactly one cover song to match with, contrasting from the typical data set for cover song retrieval that has an unknown number of covers for each query song. While a mutual nearest neighbors matching condition makes sense for this score-based data set (and other similar collections), it does mean that the choice of what each query track matches to is not independent from each other. An adjustment to the matching condition would need to be made for a data set of audio tracks with varying numbers of cover songs.

We also note that for the experiments in [7] nearly every song could be represented by aligned hierarchies, regardless of the shingle size s or threshold value T. In contrast, for the same data set under the same values for s and T, we find several songs without an AsH representationdata set, meaning that those songs cannot be represented by an AsH representation or compared to other songs' AsH representations. Songs will lack an AsH representation if none of the repeats in the aligned hierarchies have smaller repeated structures within them. We can add flexibility to our definition of what it means for two sections to be repetitions of each other by increasing the value of T. Understandably, as the value of T rises, so do the number of songs with AsH representations, meaning that an appropriate choice of T is crucial to comparison methods based on AsH representations. Even with this caveat, the results from the above experiments provide evidence in favor of the usability of AsH as a low-dimensional representation for high-dimensional sequential data with lots of repeated structure.

Finally, the AsH comparison method is based on an accumulation of structure-based comparisons between structure decompositions of sections of the query song (represented by the collection AH^R for the query) and the structure decompositions of sections in every other song in the data set (also via their collections of AH^R). As with the aligned hierarchies, the AsH-based comparisons are based on hierarchical structure decompositions of sections of songs and are more than just one level or one size of structure. What is more, each AH^R, like the aligned hierarchies, encode not one possible structure hierarchy, but all structure hierarchies that exist within that section of the song. Matchings via AsH will occur when two songs have several sections that share hierarchical structure decompositions. This is a far more nuanced matching than just matching based on one segmentation. This AsH-based comparison method is a starkly different approach than [17] which compares just one section of the query track to the other songs in the data set. This approach is reminiscent of the work in [3] that takes a truncated sum of the distances between pairs of audio shingles. The crucial difference between [3] and this work is that the former is based directly on the audio frequencies within a section of a song, while the latter is based on the lengths and positioning of repeats within sections of the song.

5. CONCLUSION

In this paper, we introduce the aligned sub-hierarchies (AsH) representation, an extension of the aligned hierarchies in [7] that allows for structure-based comparisons between sections of songs. This representation seeks to address limitations of the approach in [5] to the cover song task by creating a collection of unique structure representations for sections of each song within a data set. There is a mathematical framework underpinning AsH as shown by embedding AsH representation into a classification space with a natural metric. Finally, we address a version of the cover song task using AsH-based pairwise song comparisons on a score-based data set. These experiments provide a proof of concept for using the AsH representation for highly repetitive, sequential data and offer new insights into structure-based approaches to comparison tasks based on sections of songs.

By existing between music comparisons based on whole song representations like [5, 7] and those based on partial song representations like [17], the AsH representation opens several new avenues of research. In future work, we plan to explore the impact of relaxing the third condition in Definition 3.1, both from the theory angle of creating an appropriate metric, like the one in subsection 3.3 and from the practical angle of being able to efficiently address MIR tasks on large data sets. Further exploration of the impact of T and s on AsH is also needed. As the experiments presented here were limited to score-based data, we also plan to apply AsH-based comparisons to the cover song task on collections of audio recordings.

Acknowledgements

Part of this work is a portion of the author's doctoral thesis [6], which was partially funded by the GK-12 Program at Dartmouth College (NSF award #0947790). The author also thanks Scott Pauls, Michael Casey, Dan Ellis, Jessica Thompson, and Brian McFee for their feedback on the earlier versions of this work.

6. REFERENCES

[1] J. Bello. Measuring structural similarity in music. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2013–2025, 2011.

[2] M. Casey, C. Rhodes, and M. Slaney. Analysis of minimum distances in high-dimensional musical spaces. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(5):1015 – 1028, 2008.

[3] M. Casey and M. Slaney. Fast recognition of remixed audio. *2007 IEEE International Conference on Audio, Speech and Signal Processing*, pages IV – 1425 – IV–1428, 2007.

[4] J. Foote. Visualizing music and audio using self-similarity. *Proc. ACM Multimedia 99*, pages 77–80, 1999.

[5] P. Grosche, J. Serrà, M. Müller, and J.Ll. Arcos. Structure-based audio fingerprinting for music retrieval. *Proc. of* 13^{th} *ISMIR Conference*, pages 55–60, 2012.

[6] K. M. Kinnaird. *Aligned Hierarchies for Sequential Data*. PhD thesis, Dartmouth College, 2014.

[7] K. M. Kinnaird. Aligned hierarchies: A multi-scale structure based representation for music-based data streams. *Proc. of* 17^{th} *ISMIR Conference*, 2016.

[8] K. M. Kinnaird. Examining musical meaning in similarity thresholds. *Proc. of* 18^{th} *ISMIR Conference*, 2017.

[9] B. McFee and D. P. W. Ellis. Analyzing song structure with spectral clustering. In *Proc. of* 15^{th} *ISMIR Conference*, 2014.

[10] M. Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007.

[11] M. Müller, P. Grosche, and N. Jiang. A segment-based fitness measure for capturing repetitive structures of music recordings. *Proc. of* 12^{th} *ISMIR Conference*, pages 615–620, 2011.

[12] M. Müller and F. Kurth. Enhancing similarity for music audio analysis. *Proc. of ICASSP*, 2006.

[13] C.S. Sapp. Online database of scores in the humdrum file format. *Proc. of* 6^{th} *ISMIR Conference*, pages 664–665, 2005.

[14] J. Serrà, M. Müller, P. Grosche, and J.Ll. Arcos. Unsupervised detection of music boundaries by time series structure features. *Proc. of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[15] J. Serrà, M. Müller, P. Grosche, and J.Ll. Arcos. Unsupervised music structure annotation by time series structure features and segment similarity. *IEEE Transactions on Multimedia*, 16(5), 2014.

[16] J. Serrà, X. Serra, and R.G. Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(093017), 2009.

[17] A. L. Wang. An industrial-strength audio search algorithm. In *Proc. of* 4^{th} *ISMIR Conference*, 2003.