SuPP & MaPP: ADAPTABLE STRUCTURE-BASED REPRESENTATIONS FOR MIR TASKS

Claire Savard[1] **Erin H. Bugbee**[2] **Melissa R. McGuirl**[3]
Katherine M. Kinnaird[4]
[1] Department of Physics, University of Colorado-Boulder, USA
[2] Department of Biostatistics, Brown University, USA
[3] Division of Applied Mathematics, Brown University, USA
[4] Department of Computer Science and Program in Statistical & Data Sciences, Smith College, USA
claire.savard@colorado.edu

ABSTRACT

Accurate and flexible representations of music data are paramount to addressing MIR tasks, yet many of the existing approaches are difficult to interpret or rigid in nature. This work introduces two new song representations for structure-based retrieval methods: *Surface Pattern Preservation (SuPP)*, a continuous song representation, and *Matrix Pattern Preservation (MaPP)*, SuPP's discrete counterpart. These representations come equipped with several user-defined parameters so that they are adaptable for a range of MIR tasks. Experimental results show MaPP as successful in addressing the cover song task on a set of Mazurka scores, with a mean precision of 0.965 and recall of 0.776. SuPP and MaPP also show promise in other MIR applications, such as novel-segment detection and genre classification, the latter of which demonstrates their suitability as inputs for machine learning problems.

1. INTRODUCTION

This paper builds on the traditions of matrix representations of songs started by Foote [1]. Specifically, we are principally interested in the cover song identification task. Recent content-based approaches to this task include building objects that compare one recording against a second one [2,3], comparing slices of recordings to each other [4], creating a graph of songs on which to perform clustering [5], and using deep learning [6].

Structure-based approaches to the cover song task begin by creating representations encoding songs' structural information. These structural representations do not always explicitly encode where repetitions occur (for example, [7]). In contrast, the aligned hierarchies encode every possible repeated structure's placement within a song [8]. Between these approaches is recent work by McGuirl et al. [9] which develops start-end (SE) and start(normalized)-length (S_NL) diagrams. These structure-based musical representations seek to balance the amount of structural information provided by leveraging ideas from Topological Data Analysis (TDA), a field of applied mathematics.

We address issues of SE and S_NL diagrams, discussed in Section 2.2, by transforming S_NL diagrams into surface and matrix representations, called Surface Pattern Preservation (SuPP) and Matrix Pattern Preservation (MaPP). SuPP and MaPP are analogous to TDA persistence surfaces and persistence images [10], respectively. These novel representations can be thought of as two versions of the same concept, with SuPP as the complete representation containing all possible structural information, and MaPP as its down-sampled computationally friendly extension. MaPP can be embedded into Euclidean space [1] making calculations straightforward using distance functions. Thus, MaPPs are more usable for machine learning algorithms than their predecessors.

Additionally, SuPP and MaPP are both adaptable to varying MIR tasks, such as the cover song task, novel-segment detection, and genre classification, whereas the aligned hierarchies, SE diagrams, and S_NL diagrams were created specifically for the cover song task. We present ways in which SuPP and MaPP may be adapted for these different tasks with a larger focus on the cover song task and how these new methods compare with results from S_NL diagrams and another extension of the aligned hierarchies.

2. BACKGROUND

Continuing the work begun with the aligned hierarchies [8] and extended in S_NL diagrams [9], SuPP and MaPP are consecutive representations that are smoothings of S_NL diagrams. Additionally, MaPP combines the strengths of the aligned hierarchies and S_NL diagrams to build a representation that can be used in standard machine learning algorithms such as k-means clustering or support vector machines. In this section, we briefly review aligned hierarchies and S_NL diagrams while highlighting their limitations to motivate the novel representations proposed in this work.

[1] In fact, MaPP can be embedded into any inner product space.

2.1 Aligned Hierarchies

The aligned hierarchies representation encodes all possible hierarchical structure decompositions of a song on a single common time axis [8]. While this representation clearly shows the length of each repeated section, it does not visually emphasize the differences between the lengths of repeats using white space. Also, the distance measure for the aligned hierarchies is inefficient to compute and is quite coarse [8]. First, the distance measure notes only when two repeats of the same size line up exactly in terms of placement within the song. Second, comparisons between songs under this distance measure require both songs to be the same number of beats. This last limitation makes using the aligned hierarchies impractical for cover song detection.

2.2 SE and S_NL Diagrams

Motivated by TDA theory, SE and S_NL diagrams extend the aligned hierarchies and overcome the rigidity in comparing songs with aligned hierarchies. While SE and S_NL diagrams are more flexible and computationally efficient than their predecessor, they were built with the cover song task in mind and are difficult to adapt for other MIR tasks. Furthermore, they are not suitable to use with machine learning algorithms.

2.2.1 Topological Data Analysis Inspiration

Broadly, TDA is concerned with extracting quantifiable shape features from large, complex datasets [11–14]. Though not extracting topological information typically sought by TDA methods, SE and S_NL diagrams draw their inspiration from persistence diagrams [11, 12] which track how topological features persist across an increasing sequence of spatial scales. Just as persistence diagrams are a collection of 2-D points whose x- and y-coordinates represent the length scales at which the topological features appear and disappear, SE and S_NL diagrams are collections of 2-D points whose x- and y-coordinates represent the start and end (or length) times, respectively, of repetitive sections in a song. An advantage of the correspondence between persistence diagrams and SE and S_NL diagrams is that rich mathematical theory from TDA can be extended to the musical representations for computational tasks.

2.2.2 Structure of SE and S_NL

SE and S_NL diagrams extend the aligned hierarchies by transforming them into a representation consisting of a finite collection of points [9]. Specifically, the SE diagram for a song is defined as $\{(s_i, e_i)\}_{i=1}^N \subset \mathbb{R}^2_+$, where s_i and e_i are the start and end times, respectively, of the i^{th} repeated structure. Similarly, the S_NL diagram for a song is defined as $\{(\alpha(s_i/M), e_i - s_i)\}_{i=1}^N = \{(\bar{s}_i, \ell_i)\}_{i=1}^N$, where $\alpha > 0$ is a scaling factor and M is a normalization term related to the length of the song, with s_i and e_i as above. Figure 1 shows the transition from the aligned hierarchies to the S_NL diagram for Chopin's Mazurka Op. 6, No. 1. Both the SE and S_NL diagrams add visual emphasis for the differences between the lengths of the repetitions, but lose the visual width of each repetition.

2.2.3 Shortcomings of Previous Work

Similar to persistence diagrams, the complex structure of SE and S_NL diagrams makes them unsuitable inputs for most statistical analyses and machine learning tasks. For example, these diagrams do not live in an inner product space and simply computing averages in the space of persistence diagrams, and therefore in the spaces of SE and S_NL diagrams, remains a challenge. The notion of an average persistence diagram is defined through the Fréchet mean, which views the space of persistence diagrams as a probability space [15]. The Fréchet mean is computed as the solution of a minimization problem and is not guaranteed to be unique. Moreover, it is non-trivial to compute.

Without a unique and easy-to-compute mean or an inner product structure, the utility of SE and S_NL diagrams for MIR tasks is limited. In particular, SE and S_NL diagrams cannot be used as inputs for most classification and regression models, such as support vector machines, which would otherwise be useful for tasks like genre classification.

2.3 SuPP and MaPP Inspiration

This work continues to leverage TDA theory in the creation of two new structural representations, SuPP and MaPP. Just as persistence diagrams are transformed into persistence surfaces and persistence images [10] through a weighted sum of Gaussian functions centered at each point in a given persistence diagram, here we transform SE and S_NL into SuPP and MaPP. The mathematical details of this transformation are provided in Section 3. Like persistence images, MaPPs can be embedded into an inner product space, such as Euclidean space, and can therefore be used as inputs for machine learning algorithms.

3. METHODS

In this section, we define Surface Pattern Preservation (SuPP) and Matrix Pattern Preservation (MaPP). SuPP is a surface representation of S_NL diagrams [9] that allows similar repeated sections to be viewed as a single structure. Since comparing surfaces computationally is difficult, we introduce MaPP, the discrete matrix version of SuPP. As a matrix, MaPP allows for pairwise comparisons using common distances such as the Euclidean and Frobenius metrics. The procedure [2] for creating SuPP and MaPP from S_NL diagrams is outlined below and summarized in Algorithm 1.

3.1 Surface Pattern Preservation (SuPP)

SuPP is a smoothing of a song's S_NL diagram that maintains structural information of the song. This smoothing is achieved by placing a 2-D Gaussian at each point in a song's S_NL diagram and aggregating overlapping Gaussians.

3.1.1 Defining Repeats as Gaussians

Defining the Gaussians that represent the repeated structures of the piece is the heart of the transformation from

[2] Code is available on GitHub: `https://github.com/cgsavard/ICERM_compare_songs`

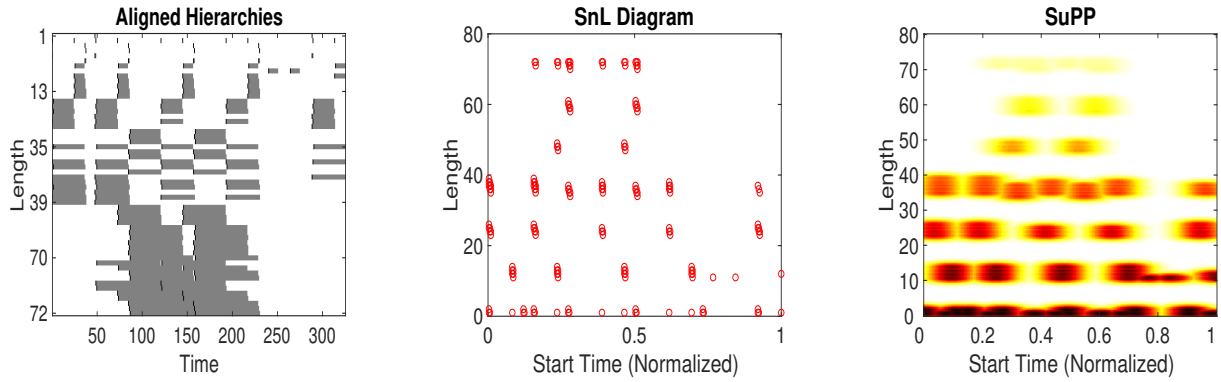

Figure 1: From left to right, the aligned hierarchies, S_NL diagram, and SuPP corresponding to the score of Mazurka Op. 6, No. 1 by Chopin. The weights applied to SuPP are the weights used for the cover song task found in Section 4.2.1.

the S_NL diagram to SuPP. In general, given a S_NL diagram $\{(\bar{s}_i, \ell_i)\}_{i=1}^N$, we first place a Gaussian (g) over each repeated structure so that $(\bar{s}_i, \ell_i) \to g_i(\bar{s}, \ell)$ where

$$g_i(\bar{s}, \ell) = \frac{1}{2\pi\sigma_s\sigma_\ell} e^{-\left(\frac{(\bar{s}-T_s(\bar{s}_i))^2}{2\sigma_s^2} + \frac{(\ell-T_\ell(\ell_i))^2}{2\sigma_\ell^2}\right)}, \quad (1)$$

$T = (T_s, T_\ell) : \mathbb{R}^2 \to \mathbb{R}^2$ defines the center of each of the Gaussians, σ_s is the standard deviation in the start direction, and σ_ℓ is the standard deviation in the length direction.

The Gaussians can either be centered at the beginning of a repeated structure, so that $T(\bar{s}_i, \ell_i) = (T_s(\bar{s}_i), T_\ell(\ell_i)) = (\bar{s}_i, \ell_i)$ is the identity, or in the middle of a repeated structure, so that $T(\bar{s}_i, \ell_i) = (T_s(\bar{s}_i), T_\ell(\ell_i)) = (\bar{s}_i + \frac{\ell_i}{2}, \ell_i)$. This choice is based on the task at hand as well as user preference. Midpoints correspond to the central beat in each repeated section, and are therefore a good indicator for where these structures are occurring, on average, in the song. For our cover song task experiment, we adjust the S_NL diagrams by choosing to center each Gaussian about the repeated structures' midpoints rather than their start times (i.e., $T(\bar{s}_i, \ell_i) = (\bar{s}_i + \frac{\ell_i}{2}, \ell_i)$). Figure 2 illuminates how using the start or midpoint of the repeated section for the Gaussians compares visually for the repetitions in aligned hierarchies. Other distributions aside from a Gaussian can also be used and may be preferable for certain MIR tasks.

After determining the appropriate placement of the Gaussians, the next step is to set the two standard deviation parameters that govern the shape of these Gaussians. The standard deviation σ_s determines the width of the Gaussians with respect to the start (or time) axis; that is, the axis representing where each repeat in the song - now represented as a Gaussian - exists. The standard deviation σ_ℓ determines the width of the Gaussians with respect to the length axis. These standard deviations control the extent to which nearby Gaussians will overlap.

Algorithm 1 Algorithm for constructing SuPP and MaPP

Input: Song's S_NL diagram $\{(\bar{s}_i, \ell_i)\}_{i \in I}$
for $i \in I$ **do**
- Replace $(\bar{s}_i, \ell_i)$ with 2-D Gaussian defined by $\mu = (\bar{s}_i, \ell_i)$ and $\sigma = (\sigma_s, \sigma_\ell)$

end for
- Aggregate all Gaussians to create a surface by using the maximum function
- Apply weighting function to surface to create SuPP
- Discretize SuPP to create gridded surface
- Integrate the area under each gridded unit and store resulting values in a matrix to create MaPP

Outputs: SuPP, MaPP

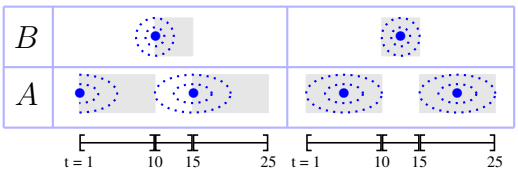

Figure 2: How 2-D Gaussians relate to repeated structures of aligned hierarchies when using the S_NL diagram (left) or the altered S_NL diagram with midpoints (right).

3.1.2 *Creating the Surface*

The next step in the SuPP creation is to aggregate the collection of Gaussians to create a surface. When two or more Gaussians intersect, which occurs when points in the S_NL diagram are close together, the SuPP value at the intersection is set to the maximum of the Gaussians rather than the sum of the Gaussians as is done for persistence surfaces [10]. That is, for any S_NL diagram $\{(\bar{s}_i, \ell_i)\}_{i=1}^N$, we define SuPP as the surface SuPP : $\mathbb{R}^2 \to \mathbb{R}$, where

$$\text{SuPP}(\bar{s}, \ell) = F(\bar{s}, \ell) * \max_{i \in [1,N]} g_i(\bar{s}, \ell), \quad (2)$$

for some weighting function $F(\bar{s}, \ell) : \mathbb{R}^2 \to \mathbb{R}$ and $g_i(\bar{s}, \ell)$ defined in Eqn. 1. Combining the Gaussians over all repeats allows repetitive structures that are similar in length and start time to be perceived as the same repeated section. We use the maximum instead of other aggregators because we want these similar structures to be treated as one section. We do not want sections to be more highly weighted if there are many similar structures in that section. The choices of σ_s and σ_ℓ determine how close points in S_NL need to be for their Gaussians to substantially blur together.

The weighting function $F(\bar{s}, \ell)$ governs which type of repeated structures are emphasized in the resulting surface. This weight controls the heights of the Gaussians in SuPP, lifting important sections of the song and suppressing other sections. The surface weight can be varied by the user based on the MIR task at hand. In Section 4, we provide an example of how one may want to set the weighting function.

3.2 Matrix Pattern Preservation (MaPP)

SuPP is a continuous representation carrying all information found in $S_N L$ diagrams. Beyond this, user-specified parameters can be set to emphasize various parts of a song. However, using a continuous surface representation is computationally complex and thus not feasible for many computing tasks. To address this challenge, a transformation of SuPP into a discrete representation with a natural embedding and metric for comparison, such as MaPP, is necessary.

To create MaPP, SuPP is first sectioned by placing a grid over the $\mathbb{R}^2$ plane over which SuPP is defined. This discretization is based on a user-defined resolution. Then, the volume beneath each grid unit of SuPP is computed using numerical integration. These volumes are recorded as entries in a matrix with the same dimensions as the gridded SuPP, resulting in MaPP.

Namely, for a $S_N L$ diagram $\{(\bar{s}_i, \ell_i)\}_{i=1}^N$, the corresponding MaPP is a $P \times P$ matrix such that:

$$\text{MaPP(SuPP)}_{jk} = \int_{\beta_k}^{\beta_{k+1}} \int_{\alpha_j}^{\alpha_{j+1}} \text{SuPP}(\bar{s}, \ell) d\bar{s} d\ell \quad (3)$$

where $\alpha_j = j\frac{R_{\bar{s}}}{P}$ and $\beta_k = k\frac{R_\ell}{P}$ are the individual grid widths and heights given by the user-defined resolution P and the ranges $R_{\bar{s}}$ and R_ℓ of the respective axes in the SuPP.

3.3 Embedding MaPP Representations

Since the MaPP representations are matrices, they can be embedded into various metric spaces. There are many pre-existing metrics with strong mathematical theory that can be applied to compute distances between matrices. We use the Frobenius norm. The Frobenius distance between two MaPPs A and B is defined as:

$$d_F(A, B) = \sqrt{\text{Tr}((A - B) * (A - B)^T)}, \quad (4)$$

where Tr indicates the trace [16]. The Frobenius norm measures the "average" value within the difference matrix $A - B$.

With any suitable embedding, we inherently define a classification space for MaPP representations and the songs that they represent. Thus, we can employ various computational techniques to compare songs. We also note that not all MIR tasks rely on song comparisons, and MaPP can also be used for exploration within a song.

4. APPLICATION TO COVER SONG TASK

SuPP and MaPP can be used in structure-based retrieval methods for a variety of MIR tasks. In this section, we show how these representations can address the cover song task, and compare our methods with some previous methods.

4.1 Dataset

To test the efficacy of SuPP and MaPP, we use a collection of Chopin's Mazurka scores as `**kern` files from the KernScore online database [3] [17]. There are 52 scores in the Mazurka dataset, and we extract two versions for each score. The first version of each piece, referred to as the "expanded" score, plays each repeated section as marked in the score. The second version, referred to as the "non-expanded" score, does not respect these repetition markers and plays marked repeated sections once. As a result, the data consists of 104 songs with pairs of expanded and non-expanded versions for each Mazurka piece.

Each score is initially represented as a thresholded self-dissimilarity matrix (SDM). Following the procedure from [8, 18], we first extract a chroma vector for each beat in the score using the Python library `music21`. We then build audio shingles [4] for each beat [19–21] by concatenating S number of consecutive chroma vectors, encoding local information for each beat. We set the shingle width to $S = \{6, 12\}$ and then compute the cosine-dissimilarity measure between each pair of audio shingles. We finally create the SDM by thresholding the matrices using $\mathcal{T} = \{0.01, 0.02, 0.03, 0.04, 0.05\}$. This SDM is converted to the aligned hierarchies [8], and then to $S_N L$ diagrams [9]. From the $S_N L$ diagram, we create a SuPP which is then discretized to become a MaPP, as described in Section 3.2.

4.2 Experiment

The cover song identification task, or version detection task, aims to identify recordings that are performances of the same piece of music. We address this task by creating SuPP and MaPP representations for each song and calculating pairwise Frobenius distances between MaPPs. The distance between two MaPPs is a measurement of structure dissimilarity, which is used alongside a clustering technique to deem whether songs are covers of the same work.

4.2.1 Setting SuPP Parameters

As discussed in Section 3.1, to create the SuPP representation we start by defining where the Gaussians representing each repetition are centered. For addressing the cover song task, the $S_N L$ diagram is adjusted so the horizontal axis reflects the midpoints of the repeated structures instead of their start times. [5] We next define σ_s, σ_ℓ, and the weighting function to determine the size, shape, and height of the 2-D Gaussians placed over each point in the $S_N L$ diagram.

For these experiments, we use a normalized constant standard deviation for σ_s, which determines the width of the Gaussians on the time axis. Recall that the time axis is normalized in the $S_N L$ diagrams so that all songs are placed within the same range. We set $\sigma_s = \frac{1}{M}$ beats for each song, where M is the normalization constant, or the total

[3] `http://kern.humdrum.org/search?s=t\&keyword=Chopin`

[4] While we are not using audio data, we still refer to these objects as audio shingles, as we are using a technique from [19–21] that uses this name.

[5] Results of our experiment were comparable between using start or midpoint times on the horizontal axis of $S_N L$ diagrams.

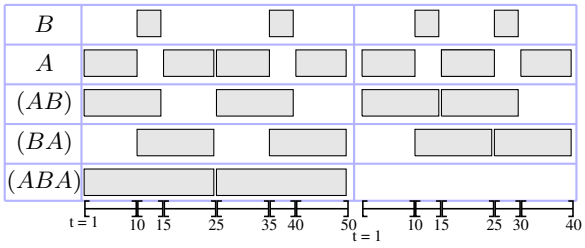

Figure 3: These aligned hierarchies represent songs of structure ABAABA (left) and ABABA (right). By omitting a middle "A" section in ABAABA, the longer repetitive sections break down while the shorter ones are preserved.

song length, inherited from the $S_N L$ diagram for a given song. This means that repeats of the same length that start within two beats ($2\sigma_s$) of each other will overlap, and thus combine to form a single structure in the SuPP.

The second parameter is σ_ℓ, which determines the width of the Gaussians in the length direction. We use a constant value for this parameter, setting $\sigma_\ell = 1$. Since the songs are not normalized along this dimension in the $S_N L$ diagrams, no normalization term is needed here. This means that there will be overlap between repeats centered at the same beat and those whose lengths differ by up to two beats.

Lastly, we choose a weighting function to apply to our surface that will give more importance to shorter structures than longer structures, due to longer repetitive structures having more variance between cover songs than shorter ones. An example of this phenomenon can be seen in Figure 3; a slight change in the overall song pattern, such as repeating the chorus one fewer time, breaks down the largest structures while maintaining the shorter repeats. To encode this importance on smaller-sized structures, we choose the following piece-wise linearly decreasing function:

$$F(\bar{s}, \ell) = \begin{cases} 1 & \ell \leq \ell_{min} \\ 1 - \frac{\ell - \ell_{min}}{\ell_{max} - \ell_{min}} & \ell_{min} \leq \ell \leq \ell_{max} \\ 0 & \ell \geq \ell_{max} \end{cases} \quad (5)$$

where ℓ_{min} and ℓ_{max} are user-specified bounds on the lengths of repetitive structures included in SuPP. We set $\ell_{min} = 0$ beats and $\ell_{max} = 80$ beats in order to include 98% of structures seen in our data. As the length increases from ℓ_{min} to ℓ_{max}, the weight decreases from one to zero.

4.2.2 Comparing MaPPs

For the cover song task, we set the MaPP resolution to $P = 200$, yielding a 200 x 200 matrix. This choice of resolution follows the work in [10, 22], which shows that this parameter value is robust, though other resolutions may be applied. This process is analogous to the resampling in [7] but is an aggregation within SuPP instead of a sampling.

After creating a MaPP for each song, we compute pairwise Frobenius distances and apply a clustering algorithm. Noting that MaPP encodes a notion of musical structure, the Frobenius distance offers a measure of the dissimilarity between the musical structures of two different pieces. For cover songs, we expect this distance to be low because songs that cover the same piece of music will likely have similar repeated musical structures.

For the Mazurka scores dataset, each song has exactly one match, namely the expanded version with repeat markers honored and the non-expanded version where the repetition markers are ignored. Therefore, we use mutual k-nearest neighbors with $k = 1$ to pair the songs. Under this technique, two songs are only labeled covers of the same piece if they both claim each other to be their closest "neighbor," corresponding to the smallest distance computed. If there is no mutual nearest neighbor, then that song is not matched to any other in the dataset.

4.2.3 Results

Ten experiments were performed with varying thresholds and shingle numbers applied to the SDMs. Overall, precision and recall values across these experiments were consistent (see Table 1) with mean precision of 0.965 and mean recall of 0.776. Figure 4 shows these results to be comparable to similar experiments on average using $S_N L$ diagrams [9] and aligned sub-hierarchies [6] (AsH) [18]. However, MaPP results have less variability among the ten experiments and thus show more stability. Additionally, MaPP is more flexible in its creation, allowing for more user creativity, and it is more computationally efficient to compare MaPPs than $S_N L$ or AsH representations, as the latter two include optimal matching steps in their comparisons.

We found that songs with few repetitive structures (and thus a scarce $S_N L$ diagram) make up the majority of the songs left without a match or improperly matched. Therefore, our method works best when analyzing songs with ample repetitive structures. An example of expanded and non-expanded versions of a score that were not matched together is shown in Figure 5, visibly due to scarce amounts of repeated structures in the non-expanded $S_N L$ diagram.

[6] AsH are extensions of aligned hierarchies that make aggregate comparisons using sections of songs instead of one cohesive structure representation as with aligned hierarchies.

Threshold ($\mathcal{T}$)	Shingle (S)	Precision	Recall
0.01	6	0.952	0.769
	12	0.975	0.778
0.02	6	0.974	0.731
	12	0.964	0.786
0.03	6	0.952	0.769
	12	0.976	0.789
0.04	6	0.952	0.769
	12	0.976	0.789
0.05	6	0.976	0.789
	12	0.954	0.789

Table 1: Experimental results for the cover song task using midpoint versions of $S_N L$ diagrams, normalized constant σ_s, constant σ_ℓ, and linearly decreasing weight along the length axis from Eqn. 5.

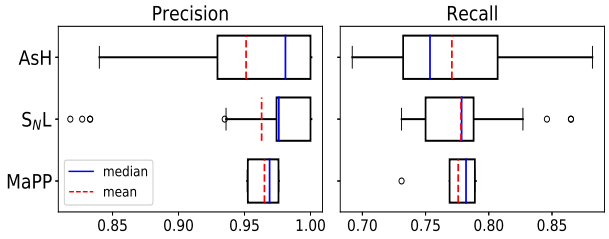

Figure 4: A comparison of precision and recall values for varying threshold and shingles for three different methods: MaPP, S_NL diagrams, and AsH. The AsH results do not include the songs which were left unmatched due to empty AsH representations (which varied between 14% and 65% of the dataset) as error, thus inflating the results compared to the other two methods.

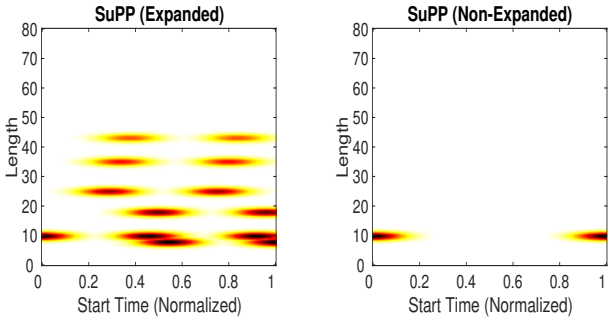

Figure 5: Mazurka Op. 68 No. 4 expanded (left) and non-expanded (right) do not match using k-mutual nearest neighbors with $k = 1$ on their corresponding MaPPs due to scarce repeated structures in the non-expanded S_NL diagram.

5. OTHER APPLICATIONS

Following are two additional examples of how SuPP and MaPP can be used to address other MIR tasks: novel-segment detection and genre classification. Preliminary experimental results show promise with both tasks, and future work will include a full analysis of these experiments.

5.1 Novel-Segment Detection

We define novel-segment detection to be finding the boundaries between repeated segments and novelty sections. This is a combination of both the novelty detection and segmentation tasks in MIR [23–25]. These boundaries often distinguish between typical segments within a musical piece, like a verse or a coda, making the segmentation task a natural application of such detection [23, 24].

For this task, we extend the analysis of MaPP from Section 4 by transforming the matrix into a vector; that is, given a MaPP, we create a 1-D vector by taking the sum of each column. This projection yields a time-dependent vector whose entry at a given time step corresponds to the sum of the structure activity measured by MaPP at that time. Global and local minima (as specified by user-specific constraints) of this projection correspond to regions between large amounts of structure. Outliers within the collection of minima correspond to novelty sections, where no repetitive structure is present for a long period of time. Preliminary work using this methodology of locating the minima of the summation projection of MaPP shows promising results, highlighting the flexibility of MaPP in other MIR tasks.

5.2 Genre Classification

In genre classification, we seek to classify songs by the genres assigned to them by their recording company. We use the collection of 104 Chopin's Mazurkas along with a selection of 676 Jazz lead sheets [26] from the *iRb Corpus* in the `**jazz` format to have two genres.

Given that MaPP representations embed into inner product spaces, we can use machine learning algorithms for MIR tasks. MaPP matrix elements become the features for each song with the number of features set by the resolution parameter. Various machine learning algorithms, implemented in sklearn, [7] are applied to our set of MaPPs as constructed in Section 4, representing 104 Mazurka scores and 676 jazz lead sheets. Logistic regression, a Gaussian kernel support vector machine (SVM), and a polynomial kernel SVM distinguish between the Mazurka and jazz pieces with high accuracy of above 94% for each classifier.

6. CONCLUSION

In this paper, we introduce SuPP and MaPP, two musical representations influenced by TDA theory. We describe how SuPP and MaPP are built and give intuition into how parameters may be chosen when applying these representations to the cover song task. Our accuracies using MaPP for this task are comparable to previous studies on average but indicate greater stability among various SDM thresholds and shingles. Preliminary experiments applying SuPP and MaPP to novel-segment detection and genre classification are plausible, demonstrating the adaptability of these representations for distinct MIR tasks.

We discuss how SuPP and MaPP overcome limitations of the aligned hierarchies [8] and S_NL diagrams [9], and how they are adaptable with user-specified parameters allowing for task-specific representations. Unlike its predecessors, MaPP is well suited for machine learning. Future work will demonstrate this through various MIR tasks, such as genre classification, and by applying similar methods to additional datasets including audio data, such as the DaTACOS dataset [27]. A drawback of SuPP and MaPP is the necessary manual selection of parameters, requiring a deep understanding of the task at hand.

SuPP and MaPP, alongside SE and S_NL diagrams [9], offer inspiring insights and open up a realm of opportunities in the intersection of TDA and MIR. These methods are both grounded in mathematical theory and have practical applications to the field of MIR, as seen by the effective use of MaPP for the cover song task. The experiments in this paper further highlight the utility of TDA-based methods and explore new opportunities for future experimentation in the intersection of TDA and MIR.

[7] `https://scikit-learn.org/stable/`

7. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. DMS-1439786 while the authors were in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the Summer@ICERM2017 and Collaborate@ICERM programs. The second author was supported by The Karen T. Romer Undergraduate Teaching and Research Awards. The third author was supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1644760. The last author is the Clare Boothe Luce Assistant Professor of Computer Science and Statistical & Data Sciences at Smith College and as such, is supported by Henry Luce Foundation's Clare Boothe Luce Program.

8. REFERENCES

[1] J. Foote, "Visualizing music and audio using self-similarity," *Proc. of ACM Multimedia 1999*, pp. 77–80, 1999.

[2] J. Serrà, X. Serra, and R. Andrzejak, "Cross recurrence quantification for cover song identification," *New Journal of Physics*, vol. 11, no. 093017, 2009.

[3] C. J. Tralie, "Early MFCC and HPCP Fusion for Robust Cover Song Identification." *Proc. of* 18^{th} *ISMIR Conference*, pp. 294–301, 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1417331

[4] D. F. Silva, F. V. Falcão, and N. Andrade, "Summarizing and comparing music data and its application on cover song identification," *Proc. of* 19^{th} *ISMIR Conference*, pp. 732–739, 2018.

[5] M. Sarfati, A. Hu, and J. Donier, "Community-based cover song detection," *Proc. of* 20^{th} *ISMIR Conference*, pp. 244–250, 2019. [Online]. Available: http://archives.ismir.net/ismir2019/paper/000028.pdf

[6] G. Doras and G. Peeters, "Cover detection using dominant melody embeddings," *Proc. of* 20^{th} *ISMIR Conference*, pp. 107–114, 2019.

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

[8] K. M. Kinnaird, "Aligned hierarchies: A multi-scale structure-based representation for music-based data streams," *Proc. of* 17^{th} *ISMIR Conference*, pp. 337–343, 2016.

[9] M. R. McGuirl, K. M. Kinnaird, C. Savard, and E. H. Bugbee, "SE and SNL diagrams: Flexible data structures for MIR," *Proc. of* 19^{th} *ISMIR Conference*, pp. 341–347, 2018.

[10] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier, "Persistence images: A stable vector representation of persistent homology," *Journal of Machine Learning Research*, vol. 18, no. 8, pp. 1–35, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-337.html

[11] G. Carlsson, A. Zomorodian, A. Collins, and L. J. Guibas, "Persistence barcodes for shapes," *International Journal of Shape Modeling*, vol. 11, no. 02, pp. 149–187, 2005.

[12] H. Edelsbrunner and J. L. Harer, *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[13] A. Zomorodian, *Topology for Computing*. Cambridge University Press, 2009.

[14] F. Chazal, V. de Silva, M. Glisse, and S. Oudot, *The Structure and Stability of Persistence Modules*, 1st ed. Springer International Publishing, 2016.

[15] Y. Mileyko, S. Mukherjee, and J. Harer, "Probability measures on the space of persistence diagrams," *Inverse Problems*, vol. 27, no. 12, p. 124007, November 2011.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

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

[18] K. M. Kinnaird, "Aligned sub-hierarchies: A structure-based approach to the cover song task," *Proc. of* 19^{th} *ISMIR Conference*, pp. 585–591, 2018.

[19] M. Casey, C. Rhodes, and M. Slaney, "Analysis of minimum distances in high-dimensional musical spaces," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 1015–1028, 2008.

[20] M. Casey and M. Slaney, "Song intersection by approximate nearest neighbor search," *Proc. of* 7^{th} *ISMIR Conference*, pp. 144–149, 2006.

[21] ——, "Fast recognition of remixed audio," *Proc. of 2007 IEEE International Conference on Audio, Speech and Signal Processing*, pp. IV–1425 – IV–1428, 2007.

[22] M. Zeppelzauer, B. Zielinski, M. Juda, and M. Seidl, "Topological descriptors for 3D surface analysis," in *CTIC*, 2016.

[23] M. Hartmann, O. Lartillot, and P. Toiviainen, "Musical feature and novelty curve characterizations as predictors of segmentation accuracy," *Proc. of the Sound and Music Computing Conference*, pp. 365–372, 2017.

[24] J. Foote, "Automatic audio segmentation using a measure of audio novelty," *Proc. of IEEE International Conference on Multi-Media and Expo*, pp. 452–455 vol.1, 2000.

[25] M. Müller, T. Prätzlich, and J. Driedger, "A cross-version approach for stabilizing tempo-based novelty detection," *Proc. of* 13*th ISMIR Conference*, pp. 427–432, 2012.

[26] Y. Broze and D. Shanahan, "The iRb Corpus in **jazz format," http://musiccog.ohio-state.edu/home/index.php/iRb_Jazz_Corpus, 2012, [Online; accessed 28-September-2016].

[27] F. Yesiler, C. Tralie, A. A. Correya, D. F. Silva, P. Tovstogan, E. Gómez, and X. Serra, "Da-TACOS: A dataset for cover song identification and understanding," *Proc. of* 20*th ISMIR Conference*, pp. 327–334, 2019. [Online]. Available: http://archives.ismir.net/ismir2019/paper/000038.pdf