IN SEARCH OF THE CONSENSUS AMONG MUSICAL PATTERN DISCOVERY ALGORITHMS

Iris Yuping Ren
Utrecht University
y.ren@uu.nl

Hendrik Vincent Koops
Utrecht University
h.v.koops@uu.nl

Anja Volk
Utrecht University
a.volk@uu.nl

Wouter Swierstra
Utrecht University
w.s.swierstra@uu.nl

ABSTRACT

Patterns are an essential part of music and there are many different algorithms that aim to discover them. Based on the improvements brought by using data fusion methods to find the consensus of algorithms on other MIR tasks, we hypothesize that fusing the output from musical pattern discovery algorithms will improve the pattern discovery results. In this paper, we explore two methods to combine the pattern output from ten state-of-the-art algorithms using two datasets. Both provide human-annotated patterns as ground truth. We show that finding the consensus among the output of different musical pattern discovery algorithms is challenging for two reasons: First, the number of patterns found by the algorithms exceeds patterns in human annotations by several orders of magnitude, with little agreement on what constitutes a pattern. Second, the algorithms perform inconsistently across different pieces. We show that algorithms lack a consensus with each other. Therefore, it is difficult to harness the collective wisdom of the algorithms to find ground truth patterns. The main contribution of this paper is a meta-analysis of the (dis)similarities among pattern discovery algorithms' output and using the output in two fusion methods. Furthermore, we discuss the implication of our results for the MIREX task.

1. INTRODUCTION

An important property of music is its recurring structures [18]. Musically meaningful repetitions in the form of musical patterns or musical motifs [29] provide one of the most intensely researched aspects both for analyzing individual musical pieces [24] and groups or collections of musical pieces for identifying musical style based on musical patterns [8, 23, 34]. Automatic pattern discovery is an active research area in Music Information Retrieval (MIR) that aims to discover these patterns automatically. Different pattern discovery methods have been introduced, such as string-based approaches [4, 7, 14, 16, 17, 25], geometric approaches [3, 6, 21, 31], data mining approaches [28], and machine learning approaches [26]. Musical pattern discovery algorithms have been used for different applications: for determining similarity between musical pieces [1], for automatic compositions [11], and for describing musical style characteristics [8].

Although many approaches have been developed over the recent decades (for a detailed overview see [12]), musical pattern discovery algorithms face a number of challenges. Music is inherently ambiguous: musicologists often do not agree on what the important musical patterns are in a given piece [5]. This makes it difficult to evaluate the quality of automatically extracted musical patterns. Furthermore, each algorithm is historically tested on unassociated datasets with disparate metrics [12]. One attempt to systematically evaluate the algorithms is the MIREX Discovery of Repeated Themes & Sections task initiated in 2014. In the task, a pattern is defined as a set of time-pitch pairs that occurs at least twice in a piece of music [10]. Although the state-of-the-art algorithms cannot reproduce the human-annotated patterns yet, they perform acceptably well according to the evaluation metrics in this task. However, the algorithms perform inconsistently across different pieces which makes it hard to determine whether there exists a single 'best' performing algorithm.

Another problem is that algorithms tend to find far more patterns than human annotators do [10]. Hence the challenge is to find which potential patterns are musically meaningful. The poor performance of automatically extracted patterns in the compression and classification task on the Dutch Song Database in [1] also shows that pattern discovery is far from being a solved problem in MIR and Computational Music Analysis.

Integrating different algorithms using data fusion has been shown to be a successful approach to improving overall performance in other areas dealing with ambiguous musical data, such as in Automatic Chord Estimation [15]. To address the challenges in musical pattern discovery, we hypothesize that integrating the output of state-of-the-art algorithms to find a consensus among these algorithms will help us to achieve an overall better pattern discovery result. To this end, we explore two fusion methods: a new algorithm, the Pattern Polling Algorithm (PPA), and the Time Indexed Novelty Algorithm (TINA), which is based on commonly used time indexed novelty scores. Using these two methods, we aim to integrate the patterns found by multiple pattern discovery algorithms to a consensus and therefore employ their collective wisdom.

Fusing the patterns produced by individual algorithms is challenging since there are different assumptions, datasets and methods behind the development of each algorithm. By exploring PPA and TINA using the MIREX dataset and the Annotated Corpus from the Dutch Song Database [32], we identify two problems with using these fusion methods. First, because the number of patterns taken as input of the fusion process is several orders of magnitude larger than the human-annotated ground truth patterns and they are disparate in terms of the pattern location, the pattern length, pattern overlap, and pattern coverage of the music pieces, it makes it difficult to find agreements among these patterns. The disagreement reflects the ambiguity of the pattern discovery task and a need for better definitions of musical patterns. Second, the individual algorithms perform inconsistently on different pieces of music. The lack of large musical pattern discovery data sets aggregates the issue of the inconsistency and prevents further improvements on using machine learning algorithms.

In this paper, we make two main **contributions**: First, we undertake a meta (dis)similarity comparison among the output of musical pattern discovery algorithms using two fusion methods, TINA and PPA (Section 2 and Section 3). Second, based on this research, we discuss issues of the MIREX Discovery of Repeated Themes & Sections task and suggest future directions for improving musical pattern discovery research (Section 4).

2. METHODS

In this section, we introduce the two fusion methods of PPA and TINA along with our evaluation methods. We use the MIREX monophonic version of Chopin's Mazurka Op. 24 No. 4 as an example to illustrate the algorithms. The code of the algorithms and supportive explanations can be found in https://github.com/irisyupingren/2017Pattern.

2.1 Algorithms Overview

The two new methods we use to explore musical pattern fusion have different goals. PPA focuses on using the gathered information to extract local pattern features (pattern boundaries), while TINA focuses on globally integrating the output patterns of individual algorithms to a probability distribution (pattern distribution).

We devise PPA based on the fact that all pattern discovery algorithms aim at finding the salient parts in musical compositions. We assume that each algorithm's output can be taken as a vote on whether or not a given time point participates in a salient part of the composition, e.g. is part of a musical pattern. Moreover, we define a salience degree of a time point which corresponds to the number of patterns that the time point participates in. In essence, the PPA is a voting system in which each algorithm votes on the salience degree of a time point based on the discovered patterns. The resulting *polling curve* is then taken as a base to detect pattern beginnings and endings.

TINA is devised based on taking the polling curve and the ground truth patterns and normalize them to a probability distribution. Along with the polling curve, we use the time indexed novelty score [9], which is produced by correlating a checkerboard kernel along the main diagonal of the similarity matrix of pattern votes. The time indexed novelty scores are then taken as a base to compare with the pattern distributions of individual algorithms, the polling curve, and the human-annotated patterns.

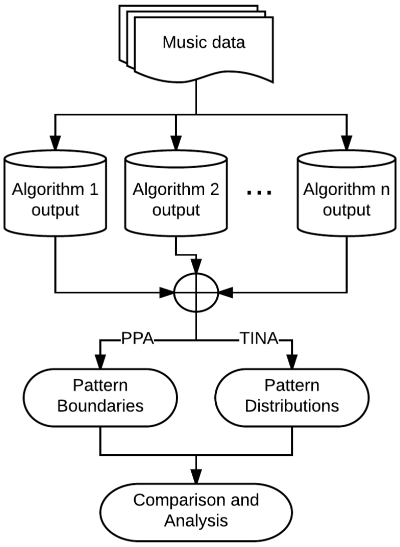

Figure 1. The pipeline of the fusion and evaluation. Same datasets and evaluation methods are used to compare two fusion methods (PPA and TINA) with individual algorithms.

The pipeline of the entire fusion and evaluation process can be found in Figure 1. For a set of music data and musical pattern discovery algorithms, we first determine the musical patterns discovered by each algorithm on each musical piece. Then we use PPA to extract pattern boundaries and use TINA to calculate the pattern distributions. Finally, we analyze the fusion results and the individual algorithms.

2.2 Pattern Polling Algorithm (PPA)

PPA starts with calculating a polling curve by taking into account all musical patterns output of all algorithms. After smoothing the polling curve, the algorithm takes the critical points (i.e. where the derivatives equal to zero) of the curve and the first derivative as the boundaries of the patterns (the beginnings and endings of the patterns). This is because changes in salience values could potentially reveal structural changes in music.

Polling Curve. The polling curve (PC) is created using the output from all individual algorithms. We let each algorithm vote at a given time point to decide whether it is a salient part of the music. To create the voting time points in the music, we use the resolution of one quarter note length. The time points where the algorithms vote are therefore in the vector $T := [0, 1, ..., n]$ with the unit of a quarter note.

The voting is realized by looking up discretized time points in the occurrences of output patterns: if there is an occurrence interval which covers the time point, we count that there is a valid vote. Finally, we add up the voting

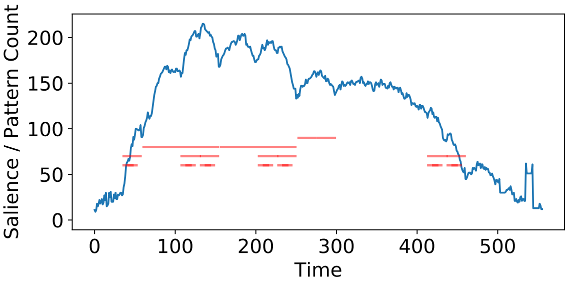

Figure 2. The polling curve of Chopin's Mazurka Op. 24 No. 4 using algorithms from the MIREX task (see Section 3). The horizontal bars show where the ground truth patterns are present. The x-axis represents time in the unit of quarter note and the y-axis represents the salience value, which is the number of pattern counts if each vote carries the same weight. We see promising correspondences between the polling curve and human annotations.

from all the algorithms and produce the polling curve $P(t)$, which is a time series consisting of the salience values at time points in T.

Since PPA uses a combination of algorithms, we should consider which algorithms we want to include or exclude. PPA could be extended if we have extra information on which algorithms should be trusted more and make a portfolio of algorithms as the input. The portfolio is essentially a way of assigning binary weights to the algorithms' votes: when the algorithm is included, its patterns have weight one, and when not included, weight zero. We can also generalize the weight to a continuous value.

To formalize the process of voting:

$$P(t) = \sum_{A}\sum_{P}\sum_{O} I_O^{A,P}(t) \tag{1}$$

where A stands for Algorithm, P stands for Pattern, O stands for occurrence, and $I_O^{A,P}(t)$ is the weighted indicator function of an occurrence in the pattern P in the algorithm A:

$$I_O^{A,P}(t) = \begin{cases} \omega_A & t \in O \subseteq P \subseteq A \\ 0 & t \notin O \subseteq P \subseteq A \end{cases} \tag{2}$$

where ω_A is the weight assigned to algorithm A.

An exemplary polling curve of Chopin's Mazurka Op. 24 No. 4 using several algorithms from the MIREX task is shown in Figure 2. The polling curve provides us with a clue of where there is a salience change in the music. Critical values (i.e. prominent changes) in salience values will be regarded as boundaries in the polling curve times series. In the following subsection, we will explain how to decide what are the prominent changes and how to reduce the possibly irrelevant micro-changes in the polling curve and then find the pattern boundaries.

Smoothing. One common way to reduce the effects of possibly irrelevant micro-changes in time series is smoothing. In our algorithm, we use the Savitzky-Golay filter [30], which is a linear least-square polynomial fitting filter. Each time we apply the smoothing, we reduce some effects of micro-changes, but at the same time, we might also lose potentially valuable details. With different degrees of smoothing, we capture different levels of details in salience's changes. Therefore, we make the degree of smoothness, s, to be one of the two parameters in PPA.

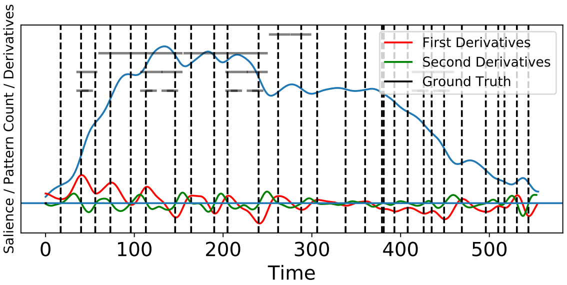

Figure 3. Extracted pattern boundaries using PPA. The dashed vertical lines are the boundaries. Many dashed lines are aligned with the boundaries of human annotations. We also plotted the polling curve, the ground truth, first and second derivatives for reference.

Derivative. After smoothing, to find the prominent changes of the salience in music, we calculate the first and second discrete derivatives of the polling curve and take their critical zero-crossing points as the pattern boundaries. More formally: let $P'(t) = P(t+1) - P(t)$ and let $P''(t) = P'(t+1) - P'(t)$, $t > 0, t \in T$. We are interested in the zero crossing $\bar{t}$ of $P'(t)$ and $P''(t)$ because the zero crossing points $\bar{t}$ represent a change of direction in the polling curve. For example, when $P'(t) < 0$ and $P'(t+1) > 0$, we have a dipping point $P'(\bar{t}) = 0$ in the curve. There are more patterns discovered starting from this point: it is likely to be a beginning of a pattern.

One question remains as for how strong the dipping, tipping, concave and convex in the curve should be so that we pick it as a boundary. Here we introduce the second parameter: a threshold on the steepness of the zero crossing points λ. With different values of λ, we create a set of boundary sets which consist of the time at which zero crossing happens. In Figure 3, an example of the extracted boundaries can be found. We notice that some boundaries line up well with ground truth boundaries. We will evaluate the extracted pattern boundaries in Section 2.4.

2.3 Time Indexed Novelty Algorithm (TINA)

Since PPA extracts local boundaries, we use TINA to assess globally how the extracted patterns are similar to human-annotated patterns. Using the notions provided in Section 2.2, TINA can be described concisely as follows: We use the pattern vote representation in Equation (2) as the input. Formally, the input matrix is $M = (I_O^{A,P_1}(t); I_O^{A,P_2}(t); ..., I_O^{A,P_n}(t))$, where n is the count of output patterns we would like to combine. The main component of TINA is the calculation of the time indexed novelty scores described in [9]. This includes calculating the similarity matrix S of M using the Euclidean distance and then multiplying the diagonal with a checkerboard kernel $K = (1, -1; -1, 1)$, which gives us the novelty curve

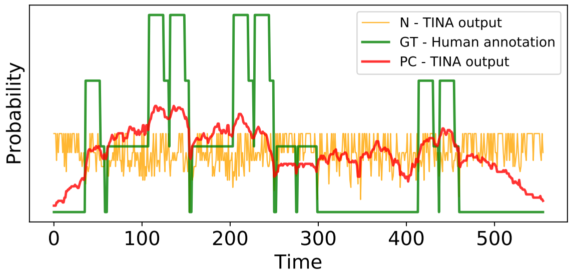

Figure 4. The TINA output novelty curve (N) distribution calculated using patterns from all algorithms, the TINA output polling curve (PC) distribution calculated using patterns from MIREX algorithms (see Section 3) and the ground truth (GT) pattern distribution. The x-axis represents time in the unit of quarter note. Correspondences of the time series can be seen from the three curves.

$N(t)$. The novelty curve represents the changing rate of the pattern vote time series $I_O^{A,P_i}(t)$, serving the same role as the derivatives in PPA. In the end, we obtain a novelty curve for each algorithm and the ensemble of algorithms, depending on which patterns are included in M.

Next, the comparison in TINA requires the input from the human-annotated ground truth patterns and the polling curve. To convert the ground truth into the same time series format as the novelty curve and the polling curve, we construct the polling curve from the ground truth patterns as $GT(t)$. Furthermore, taking the frequentists point of view, we normalize the time series by the sum of the entire time series so that we get the distributions of the novelty curve $\overline{N}(t)$, the polling curve $\overline{P}(t)$ and the ground truth patterns $\overline{GT}(t)$. Similarly, we can also construct the pattern distributions of individual algorithms $\overline{P}_A(t)$.

In Figure 4, we give an example of the novelty curve distribution, the polling curve distribution and the ground truth pattern distribution. In an initial visual inspect, we see some correspondences among the three curves: some fluctuations and the tipping/dipping points of the curves tend to coincide. We will evaluate the distribution similarities globally in the next subsection.

2.4 Evaluation

We use two evaluation methods to assess how similar the human-annotated patterns are to the output boundaries of PPA and the output distributions of TINA.

Pattern Boundaries. To evaluate the extracted pattern boundaries, we use the boundaries of the ground truth patterns. Following the standard MIREX evaluation metrics, we calculate the precision, recall and F1 score of the boundaries with a degree of fuzziness: we look for a match of boundaries with a tolerance of one quarter note length because of the one-quarter-length discretization we used for creating the polling curve.

Pattern Distribution. To evaluate globally how similar the normalized novelty curve and the polling curve are to the ground truth pattern distribution, we calculate the Bhattacharyya coefficients [13] and the Pearson correlation coefficients. Bhattacharyya coefficients measure the amount of overlaps between two distributions and the Pearson correlation coefficients measure the linear correlation of distributions. For the extracted patterns to be similar with the ground truth patterns, we expect high correlation values and high overlap values.

3. RESULTS

In this section, we first introduce the input we use for PPA and TINA and provide a meta-analysis on the individual algorithms. Then we explore the effects of the two parameters s and λ in PPA and the necessity of cross-validation. Using our evaluation metrics, we show the performance of the two fusion algorithms is on average similar to individual algorithms, and we provide analysis as to why the fusion methods do not excel.

3.1 Input: Algorithms and Music Data

We use two sets of algorithms and music data. The first set is from the Annotated Corpus of the Dutch Song Database (MTC-ANN) and the algorithms used in [1], namely PatMinr [17], MotivesExtractor (ME) [25], SIATEC [22], COSIATEC [19], and MGDP [7]. MTC-ANN [32] consists of 360 dutch folk songs in 26 tune families. Because we are interested in finding shared patterns between songs in the same tune family, the pattern discovery algorithms are computed on the concatenation of the songs in the same tune family, and then the patterns discovered on the boundaries of concatenation are filtered out (same as the intra-opus task described in [1]). The 360 individual songs are taken as the input of PPA and TINA.

The second set is from the MIREX Discovery of Repeated Themes & Sections task. For music data, we use a subset of the task's training dataset. The original dataset contains five pieces in both polyphonic and monophonic format. We take three pieces in the monophonic format: Chopin's Mazurka Op. 24 No. 4, Mozart's Piano Sonata K. 282, 2nd movement, and Beethoven Piano Sonata Op. 2 No. 1, 3rd movement. For the sake of the consistency of the task and the compatibility with MTC-ANN, we leave out the two music pieces which are constructed by a concatenation of voices in the piece. The algorithm input consists of all algorithms submitted to the MIREX task during 2014-2016: MotivesExtractor (ME) [25], SIATECCompress-TLP (SIAP), SIATECCompress-TLF1 (SIAF1), SIATECCompress-TLR (SIAR) [20], OL1 & OL2 [17], VM1 & VM2 [33], SYMCHM (SC) [27], along with SIARCT-CFP (SIACFP) [6], the algorithm developed by the task captain. The output patterns of these state-of-the-art algorithms for our example piece are shown in Figure 5. We make several observations:

1. Different algorithms find very different patterns: some tend to find shorter patterns, some longer; some find many patterns while others are more "picky".
2. We have three algorithm families (SIA, VM, and OL) which consist of more than one algorithm. The algorithms from the same algorithm family tend to find sim-

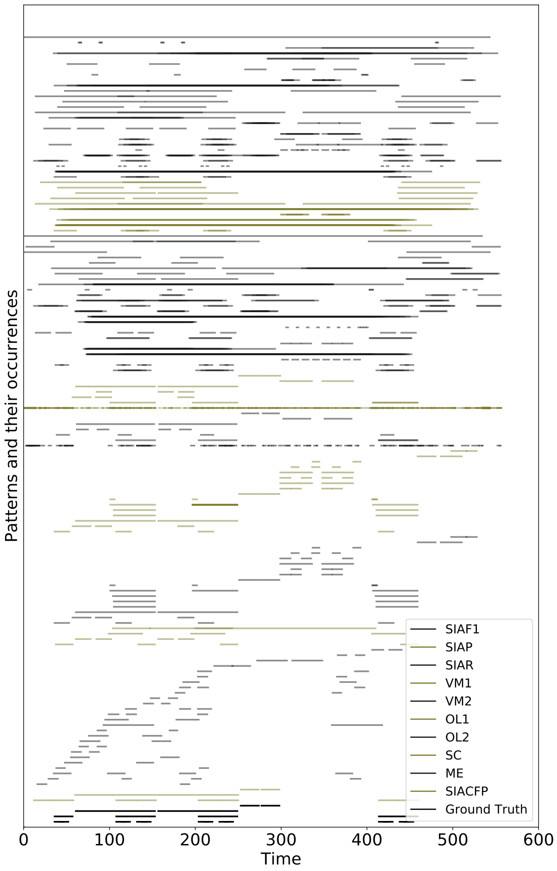

Figure 5. Patterns extracted by all algorithms submitted to the MIREX task 2014-2016 plus SIARCT-CFP on the monophonic Chopin's Mazurka Op. 24 No. 4. The horizontal bars show where the patterns are present. The x-axis represents time in the unit of quarter note. We can see the algorithms find different amount of patterns and patterns of different length, etc.

ilar patterns. Similarities here include the number of patterns discovered, the coverage of the song and the overlaps of the occurrences.

3. The ground truth is sparse in comparison to the patterns discovered by the algorithms.
4. From eyeballing the entire visualization, we see some correspondence and similarities between the algorithms and the ground truth patterns.

3.2 PPA: Parameter Space and Cross Validation

PPA extracts the local boundaries using the output patterns from individual algorithms. We start investigating the effects s and λ in PPA using the MIREX set as input, because a small number of music pieces gives us a clear idea of the relation between the parameters and the performance of PPA. Ideally, if there is a consistent best-performing s and λ across the three pieces for precision, recall and F1 score, it would be possible that the parameters can be generalized. However, we find that no single choice of s and λ performs well across all pieces. Nevertheless, to avoid over-fitting using the ground truth patterns, we perform a three-fold cross-validation using a split of two-pieces training and one piece testing in the MIREX dataset. The results of the MIREX set are shown in Table 1 and the results of MTC-ANN are shown in Table 2.

Algorithm	Precision	Recall	F1
ME	(0.125, 0.086)	(0.184, 0.077)	(0.149, 0.083)
SC	(0.396, 0.022)	(0.419, 0.068)	(0.402, 0.046)
OL1	(0.420, 0.038)	(0.565, 0.044)	(0.462, 0.023)
OL2	(0.422, 0.061)	(0.565, 0.044)	(0.483, 0.054)
SIAF1	(0.139, 0.049)	(0.670, 0.005)	(0.228, 0.041)
SIAR	(0.213, 0.039)	(0.427, 0.000)	(0.279, 0.021)
SIAP	(0.117, 0.043)	(0.596, 0.008)	(0.195, 0.037)
VM1	(0.137, 0.035)	(**1.0**, **0.0**)	(0.240, 0.029)
VM2	(0.206, 0.073)	(0.543, 0.024)	(0.296, 0.060)
SIACFP	(**0.819**, 0.030)	(0.82, 0.064)	(**0.815**, 0.046)
PPA-P	**0.478**	0.206	0.249
PPA-R	0.228	**0.867**	0.35
PPA-F1	0.248	0.738	**0.360**

Table 1. MIREX: (Mean, Variance) of the precision, recall and F1 score of different algorithms at the pattern boundary extraction task. The PPA-P, PPA-R and PPA-F1 are obtained using a 3-fold cross-validation training process optimizing the precision, the recall and the F1 scores. Because we only have one piece in the test set, there is no variance value. Bold numbers are the best results from individual algorithms and PPA.

Algorithm	Precision	Recall	F1
PatMinr	(0.465, 0.054)	(0.957, 0.020)	(0.598, 0.050)
ME	(0.366, 0.103)	(0.353, 0.098)	(0.314, .0879)
COSIATEC	(0.482, 0.049)	(0.774, 0.042)	(0.569, 0.040)
SIATEC	(0.468, 0.046)	(**0.975**, 0.017)	(**0.610**, 0.041)
MGDP	(**0.515**, 0.072)	(0.754, 0.093)	(0.557, 0.065)
PPA-P	(**0.489**, 0.135)	(0.201, 0.023)	(0.264, 0.035)
PPA-R	(0.486, 0.057)	(**0.657**, 0.046)	(**0.534**, 0.044)
PPA-F1	(0.477, 0.054)	(0.652, 0.047)	(0.526, 0.042)

Table 2. MTC-ANN results in the format of Table 1, the only difference being that we use a 10-fold cross-validation. Best results are bold.

In the MIREX set, the best F1 score of PPA ranks the fifth out of ten when using the optimal parameters found by cross-validation. The best F1 score of PPA 0.360 is better than the average of the F1 scores of individual algorithms 0.3549. The SIACFP algorithm performs overall the best on the MIREX set. With small differences, PPA ranks the fourth out of five algorithms in MTC-ANN. However, the best F1 score 0.534 of PPA is better than the average F1 score of four individual algorithms 0.510. Although PatMinr has the best F1 score in this set of music data and algorithms, other algorithms follow very closely and therefore it is hard to determine whether there is a best algorithm in this set of data and algorithms. On both datasets, we observe that PPA performs slightly better than the average of the individual algorithms.

3.3 TINA: Pattern Distributions

From a global point of view, to measure the similarities of novelty distributions, we calculate the polling curve distributions and the pattern distributions of ground truth and individual algorithms using TINA. To evaluate how similar the distributions are, we calculate the Bhattacharyya coef-

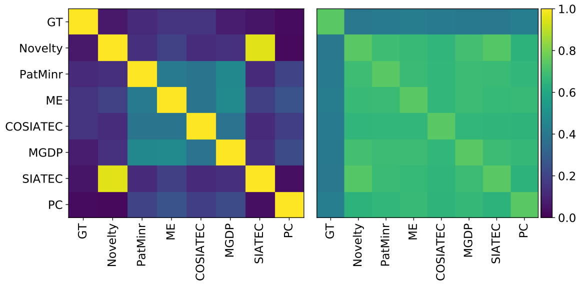

Figure 6. Left: Pairwise Pearson correlation coefficients of the ground truth distribution, individual algorithm distributions, the novelty curve distribution (Novelty) and the polling curve (PC) distribution using the 360 songs in MTC-ANN. All p-values $\ll 0.05$. Right: Pairwise Bhattacharyya coefficients of the same distributions.

ficients and the Pearson coefficients. The pairwise values of the two measurements of the MTC-ANN set of music data and algorithms are shown in Figure 6.

An obvious observation in both figures is the large distance and small correlation between the ground truth pattern distribution and the output of algorithms. Using the Bhattacharyya coefficients, we see that, in comparison to the distribution overlap differences between the ground truth and the individual algorithms, the differences among the individual algorithms and the fusion algorithms are smaller. Using the Pearson correlation coefficients, we see less linear correlation between the polling curve distribution and the ground truth distribution. Other algorithms have similar correlation values except SIATEC and the novelty curve, which have specially high correlation. This means the novelty curve is largely based on the SIATEC algorithm's output, and this is caused by the large number of output patterns generated by the algorithm. Looking at both figures, from a global point of view, output of the algorithms have similarities among themselves, but they show less correlation and similarity compared to the ground truth patterns. Similar observations are made in the MIREX dataset and hence the matrices are not shown here.

3.4 Analysis on the Results

Combining all evaluation results, we identify why the fusion methods do not excel over individual algorithms as the fusion approach applied in [15]. First, the available datasets are small and the ground truth patterns are sparse, which is problematic for training the parameters and evaluating a stable performance. Second, the algorithms disagree with each other on pattern length and pattern overlap etc., which reflects the inherent ambiguity of music and a lack of unified goal/application of the musical pattern discovery task. Third, because there are well-performing algorithms and relatively less well-performing algorithms in the fusing portfolio, fusion results are understandably of average quality since it combines results from all these different sources. In the end, although we observed promising correspondence and consensus among algorithms in Figure 4 and Figure 5, a systematic evaluation reveals that the degree of consensus is not yet enough for helping to find patterns that agree with the annotated patterns.

4. DISCUSSION AND CONCLUSION

In this paper, we attempt to combine the output of musical pattern discovery algorithms to improve musical patterns discovery. We devise a new algorithm, PPA, and apply an established method from the audio music similarity field, TINA, to musical pattern discovery. We test the fusion algorithms on pieces in the MIREX and MTC-ANN datasets. The results show that PPA and TINA on average do not improve the performance significantly. More specifically, the results from PPA show that we can extract local boundaries using a combination of musical pattern discovery algorithms, but we need to select the parameters properly. The results from TINA show that the ground truth probability distributions of musical patterns are different from the ones produced by algorithms. The results of using two datasets show that algorithms perform differently given different pieces and it is sometimes hard to select a single 'winner'. The reason of the dissatisfying performance of the fusion algorithms lies in a large number of disagreeing patterns and the sparsity of the human-annotated patterns: the salient parts of music identified by the extracted musical patterns do not align with the human annotations. To break the current limitations of applying data fusion in this domain, our work implies a need for an improved dataset and musical pattern discovery task formulation. It is also possible to improve the fusion methods by incorporating and learning more parameters from the data source.

MIREX From using the MIREX dataset in the fusion task, we identify three potential improvements for the task. First, the ground truth data from the MIREX dataset is sparse and consists of only a few pieces. It would be desirable to obtain more annotations from experts. In addition, the current ground truth consists of annotations from different sources, which could be improved by adopting a collaborative ground truth creation process [2]. Second, an open question is whether the patterns of algorithms should be compared to humanly annotated patterns as a way of evaluation, given that musicologists often disagree on the patterns: more aspects of subjectivity should be taken into account. In addition, since we see that pattern discovery algorithms produce very different patterns, one might ask whether different algorithms' output might be useful for different application scenarios. In the future of the MIREX task, instead of measuring the agreement with annotated patterns only, the testing of pattern quality by providing a range of subtasks which employ extracted patterns into various applications, constitutes a promising direction for improving the evaluation of pattern discovery algorithms.

Acknowledgements. We thank all authors of the algorithms for providing their algorithms and output, and the anonymous reviewers for valuable suggestions.

5. REFERENCES

[1] Peter Boot, Anja Volk, and W. Bas de Haas. Evaluating the role of repeated patterns in folk song classification and compression. *Journal of New Music Research*, 45(3):223–238, 2016.

[2] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga. An expert ground truth set for audio chord recognition and music analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 633–638, 2011.

[3] Chantal Buteau and Guerino Mazzola. Motivic analysis according to Rudolph Reti: Formalization by a topological model. *Journal of Mathematics and Music*, 2(3):117–134, 2008.

[4] Emilios Cambouropoulos. Musical parallelism and melodic segmentation. *Music Perception*, 23(3):249–268, 2006.

[5] Tom Collins. Improved methods for pattern discovery in music, with applications in automated stylistic composition. *PhD thesis. Milton Keynes, UK: Faculty of Mathematics, Computing and Technology, The Open University*, 2011.

[6] Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer. SIARCT-CFP: Improving precision and the discovery of inexact musical patterns in point-set representations. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 549–554, 2013.

[7] Darrell Conklin. Discovery of distinctive patterns in music. *Intelligent Data Analysis*, 14(5):547–554, 2010.

[8] Darrell Conklin and Christina Anagnostopoulou. Comparative pattern analysis of cretan folk songs. *Journal of New Music Research*, 40(2):119–125, 2011.

[9] Matthew Cooper and Jonathan Foote. Summarizing popular music via structural similarity analysis. In *Proceedings of the 7th IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 127–130. IEEE, 2003.

[10] Music Information Retrieval Evaluation eXchange (MIREX) 2013. Discovery of Repeated Themes & Sections. `http://www.musicir.org/mirex/wiki/2013`. Accessed: 2017-05-04.

[11] Dorien Herremans and Elaine Chew. Morpheus: Automatic music generation with recurrent pattern constraints and tension profiles. *Technical Report, Queen Mary University of London (2016).*

[12] Berit Janssen, W. Bas de Haas, Anja Volk, and Peter van Kranenburg. Finding repeated patterns in music: State of knowledge, challenges, perspectives. In *Proceedings of the 10th International Symposium on Computer Music Modeling and Retrieval*, pages 277–297. Springer, 2013.

[13] Thomas Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, 1967.

[14] Ian Knopke and Frauke Jürgensen. A system for identifying common melodic phrases in the masses of Palestrina. *Journal of New Music Research*, 38(2):171–181, 2009.

[15] Hendrik Vincent Koops, W. Bas de Haas, Dimitrios Bountouridis, and Anja Volk. Integration and quality assessment of heterogeneous chord sequences using data fusion. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 178–184, 2016.

[16] Olivier Lartillot. Multi-dimensional motivic pattern extraction founded on adaptive redundancy filtering. *Journal of New Music Research*, 34(4):375–393, 2005.

[17] Olivier Lartillot. PatMinr: In-depth motivic analysis of symbolic monophonic sequences. *Music Information Retrieval Evaluation eXchange (MIREX 2014)*, 2014.

[18] Elizabeth Hellmuth Margulis. *On repeat: How music plays the mind*. Oxford University Press, 2014.

[19] David Meredith. COSIATEC and SIATECCompress: Pattern discovery by geometric compression. In *Music Information Retrieval Evaluation Exchange (MIREX 2013)*, 2013.

[20] David Meredith. Using SIATECCompress to discover repeated themes and sections in polyphonic music. In *Music Information Retrieval Evaluation Exchange (MIREX 2016)*, 2016.

[21] David Meredith, Kjell Lemström, and Geraint A. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345, 2002.

[22] David Meredith, Geraint A. Wiggins, and Kjell Lemström. Pattern induction and matching in polyphonic music and other multidimensional datasets. In *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics*, pages 22–25, 2001.

[23] Leonard B. Meyer. *Style and music: Theory, History, and Ideology*. Chicago: University of Chicago Press, 1989.

[24] Jean-Jacques Nattiez and Jonathan M. Dunsby. Fondements d'une sémiologie de la musique. *Perspectives of New Music*, 15(2):226–233, 1977.

[25] Oriol Nieto and Morwaread M. Farbood. Identifying polyphonic patterns from audio recordings using music segmentation techniques. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 411–416, 2014.

[26] Matevz Pesek, Ales Leonardis, and Matija Marolt. A compositional hierarchical model for music information retrieval. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 131–136, 2014.

[27] Matevz Pesek, Urša Medvešek, Aleš Leonardis, and Matija Marolt. SymCHM: a compositional hierarchical model for pattern discovery in symbolic music representations. *Music Information Retrieval Evaluation Exchange (MIREX 2015)*, 2015.

[28] Iris Yuping Ren. Closed patterns in folk music and other genres. In *Proceedings of the 6th International Workshop on Folk Music Analysis*, 2016.

[29] Rudolph Reti. *The Thematic Process in Music*. New York: Macmillan, 1951.

[30] Ronald W. Schafer. What is a Savitzky-Golay filter? *IEEE Signal Processing Magazine*, 28(4):111–117, 2011.

[31] Wai Man Szeto and Man Hon Wong. A graph-theoretical approach for pattern matching in post-tonal music analysis. *Journal of New Music Research*, 35(4):307–321, 2006.

[32] Peter van Kranenburg, Berit Janssen, and Anja Volk. The Meertens Tune Collections: The Annotated Corpus (MTC-ANN) versions 1.1 and 2.0.1. *Meertens Online Reports*, 2016(1), 2016.

[33] Gissel Velarde and David Meredith. A wavelet-based approach to the discovery of themes and sections in monophonic melodies. *Music Information Retrieval Evaluation Exchange (MIREX 2014)*, 2014.

[34] Anja Volk, W. Bas de Haas, and Peter van Kranenburg. Towards modelling variation in music as foundation for similarity. In *Proceedings of the 12th International Conference on Music Perception and Cognition*, pages 1085–1094, 2012.