Melody Retrieval and Classification Using Biologically-Inspired Techniques

Dimitrios Bountouridis[1(✉)], Dan Brown[2], Hendrik Vincent Koops[1], Frans Wiering[1], and Remco C. Veltkamp[1]

[1] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
d.bountouridis@uu.nl
[2] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada

Abstract. Retrieval and classification are at the center of Music Information Retrieval research. Both tasks rely on a method to assess the similarity between two music documents. In the context of symbolically encoded melodies, pairwise alignment via dynamic programming has been the most widely used method. However, this approach fails to scale-up well in terms of time complexity and insufficiently models the variance between melodies of the same class. Compact representations and indexing techniques that capture the salient and robust properties of music content, are increasingly important. We adapt two existing bioinformatics tools to improve the melody retrieval and classification tasks. On two datasets of folk tunes and cover song melodies, we apply the extremely fast indexing method of the Basic Local Alignment Search Tool (BLAST) and achieve comparable classification performance to exhaustive approaches. We increase retrieval performance and efficiency by using multiple sequence alignment algorithms for locating variation patterns and profile hidden Markov models for incorporating those patterns into a similarity model.

1 Introduction

The retrieval and classification of music documents is a fundamental problem in Music Information Retrieval (MIR), and vital to the music recommendation task that underpins the ever-growing digital music industry. Assessing the similarity between two musical recordings or scores is at the core of both tasks, with direct ties to musicological analysis and music cognition. The proliferation of large music collections has placed efficiency at the center of MIR research therefore, accurate and efficient similarity methods, applied on various retrieval scenarios, are currently of vital importance.

Retrieval and classification, in general, requires building representations of previously seen classes. Two popular metaphors to this problem derive from cognition models that explains how humans categorize a concept or object [27]. The *exemplar* metaphor assumes that a new object is compared to all examples

J. Correia et al. (Eds.): EvoMUSART 2017, LNCS 10198, pp. 49–64, 2017.
DOI: 10.1007/978-3-319-55750-2_4

in a class. In computational systems this corresponds to the nearest-neighbor method. The alternative posits that a new object is compared with a representation called a *prototype* or a profile model, an abstraction created through experience that contains the most common features of the members of the class.

In the exemplar metaphor, the most common similarity method comprises converting the musical documents into sequential strings, and comparing them using alignment via dynamic programming. This approach is mainly driven by a pre-defined set of relationships between symbols, encoded as a fixed scoring matrix, and pre-defined gap penalties for changes made to one sequence or the other. Theoretically, the final similarity score between the sequences is as "meaningful" as the relationships themselves [13]. Alignment via dynamic programming has been proven very useful for identification or classification tasks where strong similarities are present [34,38]. However, the dynamic programming technique is slow and fails to efficiently scale to large datasets and long sequences. As a consequence, scalability and efficiency in MIR have been investigated by a range of researchers [4,9,23,29,30].

Most methods generally agree on the idea that representations that capture salient and robust properties of musical content are a more intuitive approach to speed up the retrieval and classification tasks. Consequently, we argue that profiles or prototypes for each class of musical documents are more appropriate than exemplars. However, applying alignment via dynamic programming on a prototype-retrieval metaphor is not straight forward. This relates to another drawback of the alignment method: its inappropriate way of modelling similarity. Studies suggest that music similarity is a complex process [32,41]. Intuitively, certain salient parts of a melody or a chord progression are less likely to change than others in a folk melody rendition or a cover song. Additionally, musicologists have argued that similarity raises from a listening process that involves altered, but not beyond recognition musical patterns called *variations*. The simplistic nature of the substitution matrix clearly cannot accommodate for the proper handling of music uncertainty and variance (or stability), because the matrix focuses only locally on individual notes.

Interestingly, in bioinformatics and biological sequence analysis, a similar situation has emerged. Various solutions have been proposed to both reduce time-complexity and properly model similar evolutionary relationships. For example, in an exemplar retrieval metaphor, the widely popular indexing framework BLAST (Basic Local Alignment Search Tool) [1,2] uses heuristics to filter out unnecessary comparisons from the database while it only explores a small part of the dynamic programming space by identifying high-matching substrings. On the other hand, in a prototype retrieval metaphor, profile hidden Markov models (profile HMMs) [14,26] have been successfully used to summarize alignments of multiple biologically-related sequences. Such profiles capture the variance in a class of sequences and can be used for accelerated database searches.

We propose the adaptation of BLAST and profile HMMs to increase the efficiency and performance of the melody classification and retrieval tasks. Although BLAST has already found application in the other music domains [22,29], we are

interested in reestablishing its practical benefits in melodies. On the other hand, in a prototype retrieval metaphor, we consider multiple sequence alignment and profile HMMs appropriate for capturing the shared salience between music variations. We conduct an empirical study and evaluation on two symbolic datasets of folk tunes and cover song melodies.

The remainder of the paper is organised as follows: Sect. 2 presents the general background and previous related research on MIR and bioinformatics. Sections 3 and 4 describe our proposed tools for the exemplar and prototype retrieval metaphors. Sections 5 and 6 present our experimental setup and results respectively. Finally, concluding remarks are in Sect. 7.

2 Background and Related Work

Sequence alignment via dynamic programming is common in a variety of domains, including bioinformatics and medicine [33]. Music documents of sequential format, such chord transcriptions or melodies, can also be compared using the same method; gaps "–" are introduced in the sequences, until they have the same length and the amount of "relatedness" between symbols at the same index position is maximized. Given that the quality of an alignment between two sequences $s_1 : c_1, c_2, .., c_n$ and $s_2 : c'_1, c'_2, .., c'_m$ is the sum of alignment scores of the individual symbols, most pairwise alignment methods use a dynamic programming method credited to Needleman and Wunsch [31]. The optimal (highest scoring) alignment can be generated by filling a cost matrix D recursively:

$$D(i,j) = max \begin{cases} D(i-1, j-1) + sub(c_i, c'_j) \\ D(i-1, j) - \gamma \\ D(i, j-1) - \gamma \end{cases}$$

where $sub(c_i, c'_j)$ is the substitution scoring function (typically encoded as a matrix) and γ is the gap penalty which is assigned when a symbol is aligned to gap. The score of the optimal alignment is stored in $D(n, m)$, while the alignment itself can be obtained by looking for adjacent maxima backwards from $D(n, m)$ to $D(0, 0)$. The Needleman and Wunsch approach is a global alignment method, since it aims to find the best score among alignments of full-length sequences. On the other hand, local alignment introduced by Smith and Waterman [35], aims to find the highest scoring alignments of partial sequences by tracking back from $\max(D(i, j))$ instead of $D(n, m)$, and by forcing all $D(i, j)$ to be non-negative. Local alignment allows for the identification of substrings (patterns) of high similarity which can be mapped to the concept of musical segments (e.g. verses, choruses, chord progressions).

We consider pairwise alignment via dynamic programming to be unfit for music classification and retrieval and particularly for melodic sequences. First, the $O(nm)$ time complexity is impractical when it comes to large databases. Although various heuristics have been introduced to reduce the time complexity of dynamic programming [33], in practice, for k sequences in the database the system would run in $O(k^2nm)$ time [29]. Müller *et al.* [30] reduce the computational

cost by first computing alignment at a coarse resolution level. The alignment is then projected into a finer level for refinement. Hu *et al.* [19] proposed an approach that relies on locating and extending promising matches incrementally on the D cost matrix to find the "best" match. Inspired by bioinformatic heuristics, Martin *et al.* [29] adapted BLAST, which reduces the cost of local alignment, for cover song identification on a large dataset of audio recordings. Martin *et al.* argue that substantial decrease in retrieval performance (compared to pairwise alignment) is compensated by the gain in computation times.

The second drawback of alignment via dynamic programming is its scoring function (or substitution matrix), which is the only component where domain knowledge can be incorporated. Such a simple, and globally applied, structure cannot accommodate for the particularities of certain classes of music documents and music similarity in general. Nevertheless, an expert-annotation study [41] of a folk-song melody dataset highlighted the importance of melodic contour, rhythm, lyrics and motifs in melodic similarity. Based on these findings, van Kranenburg [38] extended the scoring function of the typical pairwise alignment to include multiple musical dimensions (e.g. inner-metric analysis, phrase boundaries). On a melody classification task, he showed that expert-based heuristics could achieve an almost perfect 99% accuracy. At placing songs into families, Hillewaere *et al.* [17] showed that a simple all-versus-all adaptation of standard edit distance using only pitch information gives 94% accuracy for the same task. In a related work, Boot *et al.* [5] investigated the importance of repeated patterns on melodic similarity. Based on the assumption that shared patterns between variations carry more salience, they showed that pattern-based compression can achieve almost state-of-the-art classification accuracy. However, their approach failed to show similarly promising results in a retrieval scenario, presumably due to the quality of the automatic pattern extraction algorithms.

The problem of sequence similarity is well studied in bioinformatics. In a prototype retrieval metaphor, profile methods are used by most protein class identification pipelines. Sequences that include the active site in a protein (the position in the 3-dimensional structure of the protein where it binds to other molecules) are aligned using multiple alignment pipelines, and then the common region is identified using a variety of summarization methods. The summarization methods can range from the very simple PROSITE patterns (which are simple ungapped regular expressions) [3] to position-specific scoring matrices (essentially a probabilistic extension of regular expressions where there is a distribution of symbols at each position) [2]. Profile hidden Markov models (profile HMMs) [14] are also architectures that summarize multiple sequence alignments and have made major contributions to the field of computational molecular biology [13]. They are probabilistic automata that take a multiple sequence alignment and convert it into a position-specific scoring system that can be used for database searching. These models still often enable complex indexing strategies [15] that speed up the assignment process, which is necessary when studying millions of newly-generated sequences.

Despite their success, profile HMMs have found little application in MIR. Most notably, Chai and Vercoe [10] have used them to model and classify folk songs into their corresponding country of origin. Although their results showed that folk tunes from different countries share commonalities, they suggested applying profile HMMs to more discriminable data sets. Wang *et al.* [43] have used them to align different performances of the same musical piece. Results showed that profile HMMs can improve alignment accuracy and robustness over state-of-the-art pairwise methods but not significantly compared to other approaches (e.g. progressive alignment). However, they have shown to be faster, rendering them ideal for large music databases. Bountouridis *et al.* [6] have shown that profile HMMs perform better that other MSA summarization methods, in a inlier-outlier separation scenario on datasets of folk melodies, chord transcriptions and musical audio.

3 Exemplar Retrieval

The exemplar metaphor, manifested as the nearest-neighbor method, is widely used in alignment-based music retrieval. In the previous section we saw that near-perfect melody classification accuracies can be achieved by extending the typical alignment scoring with musical heuristics. The question therefore becomes whether similarly high performance can be achieved while reducing the comparison times. Boot *et al.* [5] already established the practical benefits of melodic patterns for efficiently tackling the task, but the retrieval results were not very promising. Martin *et al.* [29], on the other hand, already established that the audio-derived music sequences can be compared in near-linear time using the pattern-based BLAST. Intuitively, BLAST can find application to melodic sequences.

3.1 BLAST

BLAST [1,2] uses a local alignment method at its core, which implies that similarity is modelled as a simple substitution matrix. Its efficiency mainly relies on the idea that a good alignment contains highly similar sub-strings (called seeds). This allows for the substantial reduction of the number of database comparisons needed for a single query, since target sequences with no matching sub-strings can be filtered out. After the initial seeds are located, BLAST extends them in both directions to find longer regions of similarity above a certain threshold. Those regions are later used as anchor points that aid the local alignment. The efficiency of BLAST also relies on its indexing strategy; the whole database is divided into words, of equal size to the seed length, which are stored in a look-up table for fast accessing.

The balance between sensitivity (true positive rate) and specificity (true negative rate), in a retrieval scenario using BLAST, is mainly determined by the size of seed length s_L. Although, the work by Martin *et al.* [29] has investigated that particular relationship, their findings cannot be generalized to other datasets, as they were considering digitizations of audio recordings with specific settings.

4 Prototype Retrieval

Volk *et al.* [40] presented a comprehensive study that shed light on the importance of variations for the perceived music similarity. They argue that a variation-based computational model of music similarity requires two steps: first, the detection of variation (or stability) patterns and second, the development of an overall similarity measure that incorporates the patterns. This is a challenging task, mostly because musicology and music cognition have yet to provide us with sufficient knowledge regarding the nature of musical variations [40].

Both components of a variation-based model of music similarity can be mapped to ideas and tools from the field of computational biology. As Krogh states: "the variation in a class of sequences can be described statistically, and this is the basis for most methods used in biological sequence analysis" [25]. This agrees with the sequential, and widely supported probabilistic [36], nature of certain music representations (e.g. melodies, chord progressions) which has allowed for the successful application of sequential alignment and analysis algorithms.

Multiple sequence alignment (MSA) is an extension of pairwise alignment to more sequences, and it is a widely used sequence-analysis tool in bioinformatics. It is a way to organize sequences such that relevant features are aligned together [21]. Although "relevancy" depends on the context, the major benefit of MSA is that it allows us to identify regions of low stability or high variation. In a musical context, this can potentially translate to locating variation patterns; a prerequisite of any model of music similarity according to Volk *et al.* [40].

Profile HMMs, briefly described in Sect. 2, are theoretically more appropriate for assessing the similarity between sequences than the typical pairwise alignment because they provide a powerful framework for dealing with uncertainty and randomness in sequential data. The scoring and penalizing at each position is dependent on the MSA and not on some arbitrary chosen fixed values (e.g. scoring matrix, gap penalties). This agrees with our music intuition that certain parts of a melody are more likely to change in a variation than others.

Our proposed pipeline to enhance melody retrieval, in a prototype retrieval metaphor, comprises two steps. First, we reveal the salient parts of the variations belonging to a class, by building an MSA. Second, we encode them in a structure that allows for comparison and database searching, by building a profile HMM on top of the MSA. These can be mapped to the notions of variation and similarity modelling respectively. Both are further explained in the following sections.

4.1 Multiple Sequence Alignment (MSA)

Our approach firstly requires us to identify salient patterns between the variations belonging to a class. We propose representing melodies as sequences of symbols (from a finite alphabet) before aligning them using an MSA algorithm, such that shared patterns are revealed.

Multiple sequence alignment is the output of a process that introduces gaps "–" to sequences of symbols so that they have the same length. Formally, given k sequences $s_1, s_2, ..., s_k$ over an alphabet $\mathcal{A}$, a gap symbol "–" $\notin \mathcal{A}$ and let

$g : (\{-\} \cup \mathcal{A})^* \rightarrow \mathcal{A}^*$ a mapping that removes all gaps from a sequence containing gaps. A multiple sequence alignment A consists of k sequences $s'_1, s'_2, ..., s'_k$ over $\{-\} \cup \mathcal{A}$ such that $g(s'_i) = s_i$ for all i, $(s'_{1,p}, s'_{2,p}, .., s'_{k,p}) \neq (-, ..., -)$ for all p; and $|s'_i|$ is the same for all i.

There is a great number of possible MSAs for a single input of sequences [13]. We typically want to pick the most "meaningful" considering our task at hand. More formally: given a scoring function $c : A \rightarrow \mathbb{R}$ that maps each alignment to a real number, we are interested in $A' = \arg\max(c(A))$. The most widely used such function is the weighted sum-of-pairs (WSOP) [37]:

$$c(A) = \sum_{p=1}^{L} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} w_{i,j} v(s_{i,p}, s_{j,p}) \tag{1}$$

where L is the length of the MSA, $w_{i,j}$ is a weight of the pair of sequences i, j and $v(a, b)$ is a "relatedness" score between two symbols $a, b \in \{-\} \cup \mathcal{A}$. The scores are typically stored in a matrix format called the substitution matrix. Literature suggests that A' would be "meaningful" as long as the substitution matrix captures "meaningful" relationships between symbols [13]. WSOP can also be extended to take into consideration affine gap scores (different scores for gap insertions and gap extensions).

Representation. Related literature [21] and the previous definitions suggest that, in order to achieve a "meaningful" alignment we need to carefully select the music features that we will represent as sequences. Consequently, the representation of melodies into sequences of symbols and their relationship are of major importance. The works of van Kranenburg [38] and Hillewaere *et al.* [17] revealed the importance of the pitch dimension, so our work considers melodies as pitch-contours, meaning series of relative pitch transitions constrained to the region between $+11$ and -11 semitones (folded to one octave). Besides their simplicity and key-invariance, pitch contours have been found to be more significant to listeners for assessing melodic similarity than alternative representations [16].

Scoring. The next step towards building a "meaningful" MSA is to define the a relationship between symbols which translates to a similarity scoring matrix $\in \mathbb{R}$ and gap open and extend penalties. It is typical to vary the gap values and investigate their effect, since they have shown to affect the quality of an MSA [8]. We use the simplest scoring matrix: $v(i, i) = 1$ if $i = j$ and $v(i, j) = -1$ if $i \neq j$.

The previous paragraphs briefly explained the alignment score function c and the representations in $A' = \arg\max(c(A))$. The following paragraphs explain the two different MSA algorithms (the heuristics for the arg max function) that are investigated in our work. Remember that the literature suggests that A' would be meaningful as long as c is meaningful too. Considering the simple sequence representation and the naive scoring model, we hypothesize that the intrinsic properties of different music-agnostic MSA algorithms will lead to better and more meaningful alignments.

Progressive Alignment. The exact computations of A' is NP-hard [42], so it cannot be used in practice. Therefore, heuristic approaches that give good alignments not guaranteed to be optimal have been developed. The most popular approach is progressive alignment (PA) [18], which comprises three fundamental steps. At first, all pairwise alignments between sequences are computed to determine the WSOP similarity between each pair. At the second step, a similarity tree (guide tree) is constructed using a hierarchical clustering method. Finally, working from the leaves of the tree to the root, one aligns alignments, until reaching the root of the tree, where a single MSA is built. The drawback of PA is that incorrect gaps are retained throughout the process since the moment they are first inserted.

MAFFT. MAFFT [20] is a progressive alignment method that uses the fast Fourier transform (FFT), or an FFT approximation, to identify short subregions of one sequence or intermediate alignment that are high-scoring matches with sub-regions from another sequence or alignment. This pre-processing allows MAFFT to guide its alignment by an initial "anchoring" phase, thus reducing the overall runtime. Besides being an efficient alignment method, MAFFT's FFT approach is in theory more well-suited to music sequences: according to Margulis [28], the phrase structure of a melody is of major importance for the human perception of variation patterns. By treating the located sub-regions as gap-free segments, MAFFT can be the closest to partitioning melodies into perceptually meaningful units. In addition, Bountouridis *et al.* [6] have shown that MAFFT performs better than PA in a inlier-outlier separation task on music datasets of different nature.

4.2 Profile Hidden Markov Models (Profile HMMs)

An introduction and literature review of profile HMMs can be found at [14,25]. Here we briefly describe them. Profile HMMs are linear, left-to-right models comprised of three types of states: match, delete and insert states. Match states correspond to columns in the MSA with low variability and capture the distribution of symbols in that column. Insert states model columns of high variability, and capture the likelihood of a column being extended with the insertion of symbols. Delete states simply allow for the skipping of match states.

An example will better explain the creation of a profile HMM. Figure 1 shows a profile HMM generated from a small MSA. The structure has six match states (besides the "start" and "end" states), although the length of the MSA is eight columns. That is because we have set a gap-to-symbol ratio $\theta_{ms} = 0.3$ below which a column can be modeled as a match state. For n match states there are n delete states and $n + 1$ insert states. The transition probabilities from state to state are initially set to 0 but are modified as we parse the MSA. In our example, the columns with index 6 and 7 are modelled as an increase in the transition probability from the match state M5 corresponding to column 5 to its insert state I5. Each match state M_i has an emission distribution probability P_{M_i} corresponding to symbol distribution of that column. In order to avoid

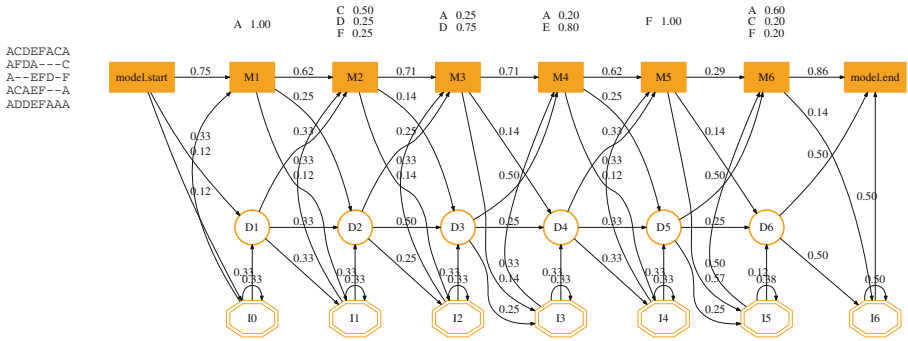

Fig. 1. An example of a profile HMM (right) created from a multiple sequence alignment (left). From top to bottom: probability distribution of the match states (only the most frequent symbols are shown) and match, delete, insert states. Not all columns have corresponding match states since $\theta_{ms} = .3$ in that particular example.

over-fitting, such as in the case of M1 where only the symbol "A" is found in the column, it is common to use pseudo-counts; where we manually increase the counts of every symbol in the alphabet. Insert states have also an emission probability distribution, but in contrast to the match states, they are assigned a fixed background symbol distribution P_I. Pseudo-counts can also be applied to the transitions between states. Altering the gap-to-symbol ratio θ_{ms}, emission pseudo-counts c_{em} and transition pseudo-counts c_{tr}, affects the flexibility of the profile HMM, meaning the allowed variation from the MSA sequences.

Profile HMMs can be trained by unaligned sequences too, using algorithms such as the Baum-Welch expectation maximization or gradient descent. A more reliable technique is to estimate a draft structure from a satisfactory MSA and then re-estimate the model's parameters using the Baum-Welch algorithm [14].

Comparing or aligning a sequence to a profile HMM is performed by finding the most likely path of states given the sequence. The Viterbi or Forward algorithms are typically used for this task, with time complexity $O(NM)$ similar to other dynamic programming methods, where N and M are the sequence length and number of states of the model respectively.

4.3 Alternative MSA Summarizations

Profile HMMs are not the only method for summarizing an MSA. In contrast to profile HMMs, all of the methods described below aim to represent an MSA as a single sequence rather than a probabilistic model. We denote this representation as "prototype" in order to avoid confusion with profile HMMs.

Random Exemplar (Random). The most naive approach to prototype modeling is picking a random sequence from the class and considering it the prototype: no MSA is required. This approach works in practice only when the

variation between the sequences is relatively small and the classes are easily separable. Then to assign a similarity between a sequences and a prototype, one just computes their pairwise alignment.

Majority-Vote Consensus (MjV). The most intuitive method to summarize a multiple alignment is to generate a single sequence, the *consensus*, that considers each aligned sequence to be of equal importance. For each column, the majority vote process determines if the frequency of the most common symbol is above a threshold, θ. If so, that symbol represents that column in the consensus; otherwise, the column is represented by an ambiguous symbol. For this simple method, the threshold is key in bioinformatics applications [11].

Data Fusion (Fusion). Data Fusion can be seen as an extension of the majority vote approach. In addition to finding the most common symbol per column, it also uses the agreement between rows as a weight to favor values of rows with higher agreement [12]. Data Fusion has already found successful application in music, i.e. in the task of automatic chord recognition from audio [24]. We omit the details due to lack of space, however the reader is forwarded to the aforementioned publications.

5 Experiments

We have proposed using bioinformatics-inspired techniques for both exemplar and prototype based retrieval metaphors. In both cases, we design a classification and a retrieval experiment. For the first task the goal is to assign individual melodies to their corresponding class. For the latter given a class, the goal is to rank higher the melodies that belong to it.

5.1 Datasets

Our experiments use two datasets of symbolically represented melodies of varying size and nature. Variations of the same melody are grouped into classes. Summary statistics are presented in Table 1.

The Annotated Corpus of the Meertens Tune Collections [39] is a set of 360 Dutch folk songs grouped into 26 "tune families" and annotated by Meertens Institute experts. Each contains a group of melody variations related through an oral transmission process. For this TuneFam-26 data set, expert annotators assessed the perceived similarity of every melody over a set of dimensions (contour, rhythm, lyrics, *etc.*) to a set of 26 prototype "reference melodies".

The Cover Song Variation data set [7], or Csv-60, is a set of expert-annotated, symbolically-represented vocal melodies derived from matching structural segments (such as verses and choruses) of different renditions of sixty pop and rock songs. Csv-60 is inherently different from TuneFam-26 in two ways. First, the grouping of melodies into classes is certain: the songs were pre-chosen as known covers of songs of interest. Secondly, cover songs are typically not a by-product of an oral transmission process: cover artists have access to the original version. All melodies in both datasets, are represented as sequences as described in Sect. 4.1.

Table 1. Summary statistics for the two datasets of our experiments.

	TuneFam-26	Csv-60
Number of classes	26	60
Number of sequences	360	243
Class size	13.0 *(4.0)*	4.0 *(1.1)*
Sequence length	43.0 *(14.9)*	53.0 *(21.4)*

5.2 Evaluation Framework

Exemplar Retrieval Settings. We are interested in evaluating whether BLAST can achieve comparable performance to the typical pairwise alignment in a nearest-neighbor retrieval scenario. Each target sequence in the dataset is compared to the query and ranked according to its similarity. For each query, the 1st-nearest neighbor is used to predict its class (e.g. tune family). Mean Average Precision (MAP) of the correct class in the ranked list is also computed. Since pairwise alignment (denoted as "PW-NN") can be highly dependant on gap open and extend settings, we experiment with different configurations (0.3–0.1, 0.5–0.1, 0.7–0.1, 0.7–0.3, 0.9–0.1, 0.9–0.3). Since BLAST is dependent on the seed size s_L, we experiment with different values $(3, 4, 5)$. BLAST's substitution matrix is set to the simple $v(i, i) = 1$ if $i = j$ and $v(i, j) = -1$ if $i \neq j$, while gap open and extend setting are set to default (1.1–0.1).

Prototype Retrieval Settings. We are interesting in evaluating whether our proposed profiling approach, i.e. MSA and profile HMMs, enables internal variation while still characterizing a class. We set up a leave-50%-out cross-validation retrieval system that randomly partitions each class into two equal-size training and testing sets. The split datasets are denoted TuneFam-26-H and Csv-60-H respectively. The *training* set is used to generate the prototypes. This includes building the MSA and summarizing with different prototype methods. For the MSA, we experiment with different algorithms (see Sect. 4.1) and different gap open and extent settings (0.3–0.1, 0.5–0.1, 0.7–0.1, 0.7–0.3, 0.9–0.1, 0.9–0.3). The match and mismatch scores are set to 1 and -1 respectively. Regarding the profile HMM training settings, we set $\theta_{ms} = 0.4$, $c_{em} = 1$ and $c_{tr} = 1$ which are considered typical. We re-estimate the model parameters by using the Baum-Welch algorithm. For the majority vote method, we experiment with three different threshold settings: 0.3, 0.5 and 0.7.

Each prototype is compared to all the sequences in the *test* set, and the MAP of the correct class in this ranked list over ten runs is computed. For the classification task, each sequence in the *test* set is compared to all the prototypes and the highest ranked class is used for prediction. For PW-NN, each sequence in the *training* set is compared to the *test* set.

6 Results

The MAP and mean classification accuracy (Acc) results of the PW-NN method (our baseline) for the Csv-60-(H) and TuneFam-26-(H) datasets are presented in Table 2. Our results for the TuneFam-26 set, agree with the findings of van Kranenburg [38]. Interestingly, gap settings have little effect on the overall performance in both datasets.

Exemplar Retrieval Results. The BLAST results are presented in Table 3. Clearly, the retrieval performance of BLAST cannot compare with that of PW-NN. However for TuneFam-26, the highest classification accuracy (0.85), achieved with $s_L = 5$, is only 0.08 lower than the best baseline performance, 0.93 (0.7–0.1 gap settings). It should be noted that Boot *et al.* [5] achieved 0.89 accuracy on the same dataset by using expert annotations of salient patterns. BLAST, achieves comparable performance without the incorporation of any musical heuristics. With the Csv-60 set, the BLAST results show similar behavior; 0.77 accuracy ($s_L = 5$) while the baseline is at 0.83 (0.7–0.1 gap settings). Due to the previous findings and its high efficiency, it is safe to state that BLAST can be a reliable and fast solution for melody classification.

Prototype Retrieval Results. The MAP and mean classification accuracy (Acc) results for the Csv-60-H and TuneFam-26-H datasets are presented in Tables 4 and 5 respectively. Our profile HMM-based model shows the highest retrieval performance in general. More specifically Mafft-pHMM achieves MAP scores of 0.76 and 0.83 for the Csv-60-H and TuneFam-26-H respectively. Both

Table 2. The baseline MAP and mean classification accuracy (Acc) results for the Csv-60-(H) and TuneFam-26-(H) datasets using pairwise alignment run with different gap settings (0.3–0.1, 0.5–0.1 and so on).

	.3–.1		.5–.1		.7–.1		.7–.3		.9–.1		.9–.3	
	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc
Csv-60	.64	.82	.64	.82	.65	.83	.64	.83	.65	.83	.64	.82
TuneFam-26	.61	.92	.62	.93	.58	.92	.60	.93	.57	.92	.61	.93
Csv-60-H	.67	.77	.66	.79	.65	.78	.65	.76	.67	.78	.65	.76
TuneFam-26-H	.57	.86	.59	.85	.61	.87	.60	.86	.62	.88	.63	.88

Table 3. The MAP and mean classification accuracy (Acc) results for the Csv-60 and TuneFam-26 datasets using BLAST run with different seed lengths s_L $(3, 4, 5)$.

	3		4		5	
	MAP	Acc	MAP	Acc	MAP	Acc
Csv-60	.33	.65	.37	.66	.46	.77
TuneFam-26	.50	.85	.47	.84	.51	.85

Table 4. The MAP and mean classification accuracy (Acc) results for the Csv-60-H dataset algorithms are compared on different MSAs created by different algorithms (Mafft, PA) run with different gap settings (0.3–0.1, 0.5–0.1 and so on).

	.3–.1		.5–.1		.7–.1		.7–.3		.9–.1		.9–.3	
	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc
Mafft-fusion	.70	.63	.68	.64	.66	.63	.70	.66	.68	.66	.66	.62
Mafft-pHMM	**.73**	**.73**	**.74**	**.76**	**.73**	**.71**	**.76**	**.74**	**.72**	**.72**	**.74**	**.71**
Mafft-MjV-.3	.60	.49	.59	.51	.59	.51	.60	.52	.57	.51	.58	.51
Mafft-MjV-.5	.60	.48	.59	.51	.59	.51	.60	.52	.57	.51	.59	.50
Mafft-MjV-.7	.53	.38	.51	.38	.49	.38	.46	.35	.46	.38	.48	.39
Mafft-random	.67	.60	.69	.64	.67	.63	.70	.64	.65	.60	.68	.60
PA-fusion	.64	.55	.66	.59	.63	.56	.68	.64	.67	.62	.67	.59
PA-pHMM	.68	.62	.66	.69	.65	.71	.70	**.74**	.64	.65	.68	.65
PA-MjV-.3	.67	.54	.67	.55	.69	.60	.68	.55	.67	.58	.65	.55
PA-MjV-.5	.67	.54	.67	.55	.69	.60	.68	.55	.67	.58	.65	.55
PA-MjV-.7	.64	.49	.65	.51	.69	.54	.61	.44	.63	.52	.59	.45
PA-random	.64	.49	.66	.52	.70	.53	.71	.60	.65	.51	.67	.56

Table 5. The MAP and mean classification accuracy (Acc) results for the TuneFam-26-H dataset algorithms are compared on different MSAs created by different algorithms (Mafft, PA) run with different gap settings (0.3–0.1, 0.5–0.1 and so on).

	.3–.1		.5–.1		.7–.1		.7–.3		.9–.1		.9–.3	
	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc
Mafft-fusion	.71	.53	.75	.57	.74	.62	.77	.58	.74	.66	.77	.62
Mafft-pHMM	**.81**	**.72**	**.80**	**.70**	**.83**	**.70**	**.79**	**.69**	**.82**	**.71**	**.81**	**.70**
Mafft-MjV-.3	.67	.55	.70	.60	.76	.63	.74	.63	.73	.65	.75	.64
Mafft-MjV-.5	.66	.51	.67	.52	.72	.55	.72	.56	.71	.57	.72	.54
Mafft-MjV-.7	.50	.28	.48	.29	.46	.32	.45	.28	.48	.30	.42	.25
Mafft-random	.57	.42	.58	.41	.60	.47	.60	.43	.60	.49	.66	.50
PA-fusion	.65	.37	.66	.37	.70	.38	.76	.53	.69	.44	.77	.55
PA-pHMM	.70	.63	.72	.64	.72	.64	.73	.67	.71	.66	.77	.66
PA-MjV-.3	.51	.44	.57	.47	.64	.54	.70	.63	.69	.60	.74	.64
PA-MjV-.5	.51	.44	.57	.47	.64	.54	.70	.64	.69	.59	.74	.63
PA-MjV-.7	.47	.39	.55	.40	.60	.49	.68	.56	.65	.52	.70	.54
PA-random	.50	.30	.55	.28	.63	.34	.65	.40	.61	.32	.67	.41

are significantly better ($p < 0.005$ using Wilcoxon signed-rank test) than the second highest MAP performed by random exemplars and Data Fusion. Mafft-pHMM is also significantly better than PW-NN; 0.67 and 0.63 for the Csv-60-H and TuneFam-26-H respectively. For classification, PW-NN performs generally better than any prototype methods. Most notably in the TuneFam-26-H set, PW-NN achieves an accuracy of 0.88 (0.9–0.1 gap settings), while the second

best (0.72) is achieved by the much faster MAFFT-pHMM. The same pattern holds at smaller extent for CSV-60-H; 0.79 (0.3–0.1 gap settings) and 0.76 for PW-NN and MAFFT-pHMM respectively.

7 Discussion and Conclusions

In this paper we proposed the adaptation of two popular tools of computational biology in the context of melody classification and retrieval. In a nearest-neighbor experiment, we showed that the pattern-based indexing tool BLAST can achieve high classification accuracy, comparable to music-aware sequence-compression methods that use global alignment via dynamic programming. As expected though, due to the improper modelling of similarity using a substitution matrix, the retrieval performance did not show similar behaviour. Nevertheless, we showed that MSA and profile HMMs are a great fit for modelling the randomness and uncertainty of variations and incorporating them into class profiles. The results showed that our proposed similarity model can outperform the global alignment method in a prototype retrieval metaphor. Profile HMMs were also shown to outperform alternative MSA summarization methods. Considering the previous findings and the high efficiency of profile HMMS when it comes to sequence comparison, we find that profile HMMs are a reliable and fast solution for melody retrieval. Interestingly, their performance is notably higher when the MSA comes from MAFFT instead from progressive alignment. This suggests that MAFFT's FFT-based internal heuristics are more appropriate than the typical progressive alignment for melodic sequences.

In general, BLAST and profile HMMs (trained on MAFFT MSAs) can be reliable and efficient solutions for large-scale melody classification and retrieval respectively, without the incorporation of musical heuristics. Future work should investigate their performance on music documents of more popular sequential formats (e.g. chords). Interestingly, profile HMMs, offer the possibility of generating variations based on the training MSA which can find applications in the field of automatic music generation. From a music-cognition perspective, it would be interesting to investigate whether listeners can distinguish generated from real variations. Our work can be considered as the first step towards more exciting research and applications.

References

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. **215**(3), 403–410 (1990)
2. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Res. **25**(17), 3389–3402 (1997)
3. Bairoch, A.: Prosite: a dictionary of sites and patterns in proteins. Nucleic Acids Res. **19**(Suppl), 2241 (1991)

4. Bertin-Mahieux, T., Ellis, D.P.W.: Large-scale cover song recognition using hashed chroma landmarks. In: Applications of Signal Processing to Audio and Acoustics, pp. 117–120 (2011)
5. Boot, P., Volk, A., de Haas, W.B.: Evaluating the role of repeated patterns in folk song classification and compression. J. New Music Res. 1–16 (2016)
6. Bountouridis, D., Koops, H.V., Wiering, F., Veltkamp, R.C.: Music outlier detection using multiple sequence alignment and independent ensembles. In: Amsaleg, L., Houle, M.E., Schubert, E. (eds.) SISAP 2016. LNCS, vol. 9939, pp. 286–300. Springer, Cham (2016). doi:10.1007/978-3-319-46759-7_22
7. Bountouridis, D., Van Balen, J.: The cover song variation dataset. In: The International Workshop on Folk Music Analysis (2014)
8. Carroll, H., Clement, M.J., Ridge, P., Snell, Q.O.: Effects of gap open and gap extension penalties. In: The Biotechnology and Bioinformatics Symposium, pp. 19–23 (2006)
9. Casey, M., Slaney, M.: Fast recognition of remixed music audio. In: Acoustics, Speech and Signal Processing, vol. 4, p. IV-1425 (2007)
10. Chai, W., Vercoe, B.: Folk music classification using hidden Markov models. In: International Conference on Artificial Intelligence, number 6 in 4. Citeseer (2001)
11. Day, W.H.E., McMorris, F.R.: Threshold consensus methods for molecular sequences. J. Theor. Biol. **159**(4), 481–489 (1992)
12. Dong, X.L., Berti-Equille, L., Srivastava, D.: Integrating conflicting data: the role of source dependence. Proc. VLDB Endowment **2**(1), 550–561 (2009)
13. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge (1998)
14. Eddy, S.R.: Profile hidden Markov models. Bioinformatics **14**(9), 755–763 (1998)
15. Finn, R.D., Clements, J., Eddy, S.R.: Hmmer web server: interactive sequence similarity searching. Nucleic Acids Res. gkr367 (2011)
16. Gómez, E., Klapuri, A., Meudic, B.: Melody description and extraction in the context of music content processing. J. New Music Res. **32**(1), 23–40 (2003)
17. Hillewaere, R., Manderick, B., Conklin, D.: Alignment methods for folk tune classification. In: Spiliopoulou, M., Schmidt-Thieme, L., Janning, R. (eds.) Data Analysis, Machine Learning and Knowledge Discovery, pp. 369–377. Springer, Cham (2014)
18. Hogeweg, P., Hesper, B.: The alignment of sets of sequences and the construction of phyletic trees: an integrated method. J. Mol. Evol. **20**(2), 175–186 (1984)
19. Hu, N., Dannenberg, R.B., Tzanetakis, G.: Polyphonic audio matching and alignment for music retrieval. Computer Science Department, p. 521 (2003)
20. Katoh, K., Misawa, K., Kuma, K., Miyata, T.: Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. Nucleic Acids Res. **30**(14), 3059–3066 (2002)
21. Kemena, C., Notredame, C.: Upcoming challenges for multiple sequence alignment methods in the high-throughput era. Bioinformatics **25**(19), 2455–2465 (2009)
22. Kilian, J., Hoos, H.H.: Musicblast-gapped sequence alignment for MIR. In: International Society for Music Information Retrieval Conference, pp. 38–41 (2004)
23. Kim, S., Narayanan, S.: Dynamic chroma feature vectors with applications to cover song identification. In: Multimedia Signal Processing, pp. 984–987 (2008)
24. Koops, H.V., de Haas, W.B., Bountouridis, D., Volk, A.: Integration and quality assessment of heterogeneous chord sequences using data fusion. In: International Society for Music Information Retrieval Conference, pp. 178–184 (2016)

25. Krogh, A.: An introduction to hidden Markov models for biological sequences. New Compr. Biochem. **32**, 45–63 (1998)
26. Krogh, A., Brown, M., Saira Mian, I., Sjölander, K., Haussler, D.: Hidden Markov models in computational biology: applications to protein modeling. J. Mol. Biol. **235**(5), 1501–1531 (1994)
27. Malt, B.C.: An on-line investigation of prototype and exemplar strategies in classification. J. Exp. Psychol. Learn. Mem. Cogn. **15**(4), 539 (1989)
28. Margulis, E.H.: Musical repetition detection across multiple exposures. Music Percept. Interdisc. J. **29**(4), 377–385 (2012)
29. Martin, B., Brown, D.G., Hanna, P., Ferraro, P.: Blast for audio sequences alignment: a fast scalable cover identification. In: International Society for Music Information Retrieval Conference, pp. 529–534 (2012)
30. Müller, M., Mattes, H., Kurth, F.: An efficient multiscale approach to audio synchronization. In: International Society for Music Information Retrieval Conference, pp. 192–197. Citeseer (2006)
31. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. **48**(3), 443–453 (1970)
32. Pampalk, E.: Computational models of music similarity and their application in music information retrieval. na (2006)
33. Ratanamahatana, C.A., Keogh, E.: Everything you know about dynamic time warping is wrong. In: Third Workshop on Mining Temporal and Sequential Data, pp. 1–11. Citeseer (2004)
34. Serra, J., Gómez, E., Herrera, P., Serra, X.: Chroma binary similarity and local alignment applied to cover song identification. IEEE Trans. Audio Speech Lang. Process. **16**(6), 1138–1151 (2008)
35. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. **147**(1), 195–197 (1981)
36. Temperley, D.: Bayesian models of musical structure and cognition. Musicae Sci. **8**(2), 175–205 (2004)
37. Thompson, J.D., Higgins, D.G., Gibson, T.J.: Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. **22**(22), 4673–4680 (1994)
38. van Kranenburg, P.: A computational approach to content-based retrieval of folk song melodies. Ph.D. thesis, Utrecht University (2010)
39. van Kranenburg, P., de Bruin, M., Grijp, L., Wiering, F.: The Meertens tune collections. In: Meertens Online Reports (2014)
40. Volk, A., Haas, W.B., Van Kranenburg, P.: Towards modelling variation in music as foundation for similarity. In: Proceedings of the 12th International Conference on Music Perception and Cognition (2012)
41. Volk, A., Van Kranenburg, P.: Melodic similarity among folk songs: an annotation study on similarity-based categorization in music. Musicae Sci. **16**, 317–339 (2012). page 1029864912448329
42. Wang, L., Jiang, T.: On the complexity of multiple sequence alignment. J. Comput. Biol. **1**(4), 337–348 (1994)
43. Wang, S., Ewert, S., Dixon, S.: Robust joint alignment of multiple versions of a piece of music. In: International Society for Music Information Retrieval, pp. 83–88 (2014)