A METHOD FOR STRUCTURAL ANALYSIS OF OTTOMAN-TURKISH MAKAM MUSIC SCORES

Sertan Şentürk, Xavier Serra
Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
{sertan.senturk,xavier.serra}@upf.edu

ABSTRACT

From a computational perspective, structural analysis of Ottoman-Turkish makam music (OTMM) is a research topic that has not been addressed thoroughly. In this paper we propose a method, which processes machine-readable music scores of OTMM to extract and semiotically describe the melodic and lyrical organization of the music piece automatically using basic string similarity and graph analysis techniques. The proposed method is used to identify around 50000 phrases in 1300 music scores and 21500 sections in 1770 scores, respectively. The obtained information may be useful for relevant research in music education and musicology, and it has already been used to aid several computational tasks such as music score content validation, digital music engraving and audio-score alignment.

1. INTRODUCTION

In analyzing a music piece, scores provide an easily accessible symbolic description of many relevant musical components. Moreover they typically include editorial annotations such as the nominal tempo, the rhythmic changes and structural markings. These aspects render the music score a practical source to extract and analyze the melodic, rhythmic and structural properties of the studied music.

Analyzing the structure of a music piece is integral in understanding how the musical events progress along with their functionality within the piece. Automatic extraction of the melodic and lyrical structures, as well as their roles within the composition, might be used to facilitate and enhance tasks such as digital music engraving, automatic form identification and analysis, audio-score and audio-lyrics alignment, music prediction and generation.

Structural analysis is a complex problem which can be approached in different granularities such as sections, phrases and motifs (Pearce et al., 2010). To find such groupings there has been many approaches based on music theory (Jackendoff, 1985), psychological findings and computational models (Cambouropoulos, 2001; Pearce et al., 2010). On the other hand, there are a few studies that has investigated automatic structural analysis of makam musics. Lartillot & Ayari (2009) has used computational models to segment Tunisian modal music and compared the segmentations with the annotations of the experts. Lartillot et al. (2013) has proposed a similar segmentation model for OTMM and also conducted comparative experiments between the automatic segmentations and human annotations. Due to the lack of musicological agreement on how to segment makam music scores, Bozkurt et al. (2014) focused on learning a model from a dataset of music scores annotated by experts and segmenting larger score datasets automatically using the learned model. They propose two novel culture-specific features based on the melodic and rhythmic properties of OTMM and conduct comparative studies with the features used in the state-of-the-art methodsand show that the proposed features improve the phrase segmentation performance. [1] These methods typically focus on finding the segment boundaries and do not study the inter-relations between the extracted segments.

In this study, we propose a method which extracts both the melodic and lyrical organization on phrase-level and section-level using symbolic information available in the music scores of Ottoman-Turkish makam music. The method labels the extracted sections and phrases semiotically according their relations with each other using basic string similarity and graph analysis. Our contributions are:

- An automatic structural analysis method applied on Ottoman-Turkish makam music scores
- A novel semiotic labeling method based on network analysis
- An open implementation of the methodology extending our existing score parser
- A dataset of sections and phrases automatically extracted from more than 1300 and 1750 music scores, respectively

The structure of the rest of the paper is as follows: Section 2 decribes the OTMM score collection we use in our analysis, Section 3 defines the problem and scope of the analysis task, Section 4 presents the proposed methodology, Section 5 explain the experiments and Section 6 discusses our findings, Section 7 gives the use cases where we have already integrated the extracted structure information, finally Section 8 suggests future directions to be investigated and concludes the paper. [2]

2. SCORE COLLECTION

In the analysis, we use the release v2.4.1 of the SymbTr score collection (Karaosmanoğlu, 2012). [3] This release includes 2200 scores from the folk and classical repertoires.

[1] For a detailed review of structural analysis applied to OTMM and relevant state of the art we refer the readers to (Bozkurt et al., 2014) and (Pearce et al., 2010), respectively.

[2] The relevant content such as the implementation of the methodology, the score collection, the experiments, the results are also accessible via the companion page http://compmusic.upf.edu/node/302.

[3] https://github.com/MTG/SymbTr/releases/tag/v2.4.1

It is currently the largest and the most representative machine-readable score collection of OTMM aimed at research purposes (Uyar et al., 2014). The scores typically notate the basic melody of the composition devoid of the performance aspects such as intonation deviations and embellishments. The scores also include editorial metadata such as the composer, the makam, the form, the usul (rhythmic structure) of the composition. We use the scores in txt format in our analysis, as they are the reference format from which the other formats are generated.

The content in the SymbTr-txt scores are stored as "tab separated values," where each row is a note or an editorial annotation (such as usul change) and each column represent an attribute such as the note symbol, the duration, the measure marking and the lyrics. The pitch intervals are given according to both the 24 tone-equal-tempered (TET) system defined in the Arel-Ezgi-Uzdilek theory and the 53-TET system. [4] The lyrics are synchronous to the note onsets on the syllable level. The final syllable of each word ends with a single space and the final syllable of each poetic line ends with double spaces (Karaosmanoğlu, 2012). Some columns may be overloaded with additional types of information. For example the lyrics row also includes editorial annotations such as the section names, instrumentation and tempo changes, entered in capital letters.

As will be explained in Section 4.1, we use the explicit section names along with the poetic line ends mentioned above in the section extraction step. However, this set of editorial annotations does not convey the complete information about the section boundaries and the section names. First, the section name (and hence the first note of a section) is only given for the instrumental sections and the final note of these sections are not marked at all. Moreover, the section name does not indicate if there are any differences between the renditions of the same section. Regarding the vocal sections, only the last syllable of a poetic line is marked as explained above. This mark does not typically coincide with the actual ending of the vocal section since a syllable can be sung for longer than one note or there might be a short instrumental movement in the end of the vocal section. Out of 2200, 1771 txt-scores in the SymbTr collection has some editorial section information. The remaining 429 scores either lack the editorial section information or they are very short such that they do not have any sections.

3. PROBLEM DEFINITION

As explained in Section 1, symbolic structural analysis is a complex problem that can be approached from different perspectives and granularities. For our initial work in the topic, we assume that the structural elements of the same type are non-overlapping and consecutive (e.g. the last note of a sectoion is always adjacent to the first note of the next section). Consecutiveness restriction also implies that any transitive interactions between two consecutive structural elements are ignored.

Given the note sequence $N := \{n_1, n_2, \dots\}$ and the measure sequence $M := \{m_1, m_2, \dots\}$ in the score, our aim is to extract the sections $S := \{s_1, s_2, \dots\}$ and the phrases $P := \{p_1, p_2, \dots\}$ (which we call as structural elements collectively, throughout the text) along with their boundaries, and the melodic and lyrical relationship with other structural elements of the same type. We assume each poetic line as a section.

Remark that each subsequence [5] might cover or overlap with subsequences of different types, e.g. the note sequence in a section would be a subsequence of N or a phrase might start in the middle of a measure and end in another. We denote the index of the first note and the index of the last note of an score element x in the note sequence N as $\beta(x)$ and $\gamma(x)$, respectively. For example, the start of an arbitrary section s_i, phrase p_j and measure m_k are denoted as $\beta(s_i)$, $\beta(p_j)$ and $\beta(m_k)$, respectively.

4. METHODOLOGY

We first extract the section boundaries from the score using a heuristic process taking the editorial structure labels in the score as an initial reference (Section 4.1). In parallel, we automatically segment the score into phrases according to a model learned from the phrases annotated by an expert (Section 4.2). Next, we extract the synthetic pitch and the lyrics of each section and phrase (Section 4.3). Then, a melodic and a lyrical similarity matrix are computed between the extracted phrases and the sections separately. A graph is formed from each similarity matrix and the relation between the structural elements in the context of the similarity (melodic or lyrical) is obtained (Section 4.4). Finally semiotic labeling is applied to the computed relations (Section 4.5).

4.1 Section Extraction

We infer section boundaries using the explicit and implicit boundaries given in the lyrics column of the SymbTr-txt scores (Section 2). As a preprocessing step to distinguish the instrumental section labels from other editorial annotations in the lyrics column, we extract the unique strings in the lyrics column of all SymbTr scores. We only keep the strings, which are written in capital letters and obtain the set of all editorial annotations in the SymbTr-scores. Then, we pick the section annotations manually. [6]

Given a score, we first search the set of instrumental section names in the lyrics column. The matched note indices mark the actual beginning $\beta(s_i)$s of the instrumental sections $s_i \in S \,|\, \lambda(s_i) = \varnothing$. Next, the lyrics column is searched for syllables ending with double spaces. The index of the matched notes are assigned to the final note $\gamma(s_i)$s of the vocal sections $s_i \in S \,|\, \lambda(s_i) \neq \varnothing$. As explained in Section 2, the index $\gamma(s_i)$s may not coincide

[4] The unit interval of the 53-TET, which is simply the $1/53$th of an octave, is called a Holderian comma (Hc).

[5] or element, which can also be regarded as a subsequence composed of a single element

[6] https://github.com/sertansenturk/symbtrdataextractor/blob/master/symbtrdataextractor/makam_data/symbTrLabels.json

with the actual ending and it may be moved to a subsequent note.

Up to here we have found the section sequence $S := \{s_1, s_2, \dots, s_I\}$, where $_I$ is the total number of sections. The first note of the vocal sections and the last note of the instrumental sections are unassigned at this stage. We proceed to locate the section boundaries using a rule-based scheme iterating though all sections starting from the last one.

If a section s_i is instrumental, the $\beta(s_i)$ is already assigned. If a section s_i is vocal and the previous section s_{i-1} is instrumental, we find the last instrumental measure, $m_k \in M \mid \lambda(m_k) = \varnothing$, before the last note $\gamma(s_i)$ of the section s_i. We then assign the first note $\beta(s_i)$ to the first note $\beta(m_{k+1})$ of the next measure m_{k+1}. If both the current section s_i and the previous section s_{i-1} are vocal, we assign $\beta(s_i)$ to the index of the first note with lyrics after the last note $\gamma(s_{i-1})$ of s_{i-1}. If $\beta(s_i)$ and $\gamma(s_{i-1})$ are not in the same measure, we reassign $\beta(s_i)$ to the first note of its measure, i.e. $\beta(m_k) \mid \beta(s_i) \in m_k$. Finally the last note $\gamma(s_i)$ of the section is moved to the index of the first note $\gamma(s_{i+1})$ of the next section s_{i+1} minus one. The pseudocode of the procedure is given in Algorithm 1. Note that the start of the first section and the end of the final section are assigned to 1 and $|N|$, respectively, where $|N|$ is the number of notes in the score. This detail omitted from the pseudocode for the sake of brevity.

Algorithm 1 Locate section boundaries

```
for i := L → 1 do                          ▷ from the last index to the first
    if s_i is vocal then                   ▷ find β(s_i) of the vocal section
        if s_{i−1} is instrumental then
            β(s_i) ← arg_k min (β(m_k) > β(s_{i−1}) ∧
                                 λ(m_k) ≠ ∅)
        else                                           ▷ s_{i−1} is vocal
            β(s_i) ← arg_k min (k > γ(s_{i−1}) ∧
                                 λ(n_k) ≠ ∅)
            if β(s_i) ∈ m_k ∧ γ(s_{i−1}) ∉ m_k then
                β(s_i) ← β(m_k)
    γ(s_i) ← β(s_{i+1}) − 1                ▷ sections are consecutive
```

Having located the boundaries, the sections are extracted by simply taking all information (i.e. rows in the SymbTr-txt score) between these note boundaries. Figure 1 shows the section boundaries obtained on a mock example.

4.2 Automatic Phrase Segmentation

In our method we use the automatic phrase segmentation methodology proposed by Bozkurt et al. (2014) (Section 1). The source code and the training dataset (Karaosmanoğlu et al., 2014) are open and available online. [7]

In order to train the segmentation model, we use the annotations of Expert 1, who annotated all the 488 scores in the training dataset (Karaosmanoğlu et al., 2014). There are a total of 20801 training phrases annotated by the first expert. Using the trained model, we apply automatic phrase segmentation on the score collection (Section 2) and obtain the phrase boundaries $\beta(p_k)$ and $\gamma(p_k)$ for each phrase $p_k \in P := \{p_1, p_2, \dots\}$, where P is the automatically extracted phrase sequence. In Figure 5 (Appendix A), the vertical red and purple lines shows the phrase boundaries extracted from the score "Kimseye Etmem Şikayet." [8]

4.3 Synthetic Pitch and Lyrics Extraction

We use the information in the lyrics column to determine the boundaries of the vocal sections in Section 4.1. Later, the lyrics of each structural element are extracted in Section 4.4 and a lyrical similarity is computed between each structural element of the same type using the extracted. The lyrics associated with a sequence or an element x is a string denoted as $\lambda(x)$, simply obtained by contatenating the syllables of the note sequence $\{\beta(x), \dots, \gamma(x)\}$ of x. The editorial annotations (Section 2) and the whitespaces in the lyrics column are ignored in the process. Then the characters in the obtained string are all converted to lower case. Trivially, $\lambda(n_i)$ of a note n_i is the syllable associated with the note n_i in the lyrics column.

Given a subsequence or element x in the score, the synthetic pitch $\rho(x)$ is computed by sampling each note in x according to the note symbol and the duration, and then concatenating all of the samples (Şentürk et al., 2014). The synthetic pitch is used in melodic similarity computation parallel to the lyrics (Section 4.4). Figure 2 shows the lyrics and the synthetic pitch extracted from an except of the SymbTr-score of the composition "Gel Güzelim".

4.4 Melodic and Lyrical Relationship Computation

Given the structure sequence $F := \{f_1, f_2, \dots\}$ (which is either the section sequence S or the phrase sequence P) extracted from the score, we first compute the synthetic pitch and extract the lyrics of each structural element (Section 4.3). Then, we compute a melodic similarity and lyrical similarity between each element using a similarity measure based on Levenshtein distance (Levenshtein, 1966). The similarity measure $\hat{L}(x, y)$ is defined as:

$$\hat{L}(x, y) := 1 - \frac{L(x, y)}{max\,(|x|, |y|)} \qquad (1)$$

where $L(x, y)$ is the Levenshtein distance between the two "strings" x and y with the lengths $|x|$ and $|y|$, respectively and $max()$ denotes the maximum operation. In our case, x and y are the synthetic pitch or the lyrics of two structural elements. The similarity yields a result between 0 and 1. If the strings of the compared structural elements are exactly the same, the similarity will be one. Similar strings (e.g. the melodies of two instances of the same section with volta brackets) will also output a high similarity.

From the melodic and lyrical similarities, we build two separate graphs, in which the nodes are the structural elements and the elements are connected to each other with

[7] http://www.rhythmos.org/shareddata/turkishphrases.html

[8] https://github.com/MTG/SymbTr/blob/a50a16ab4aa2f30a278611f333ac446737c5a877/txt/nihavent--sarki--kapali_curcuna--kimseye_etmem--kemani_sarkis_efendi.txt

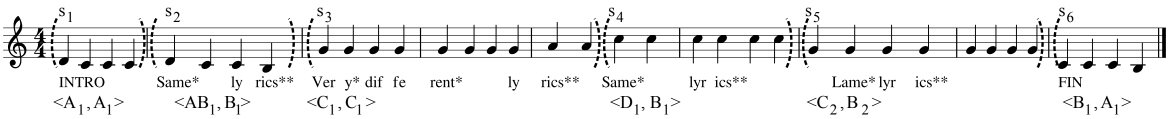

Figure 1: Section analysis applied to a mock example. The section labels ("INTRO" and "FIN") are given in the lyrics written in capital letters, The spaces in the end of the syllables are visualized as *. The semiotic $<$ *Melody*, *Lyrics* $>$ label tuples of each section are shown below the lyrics. The similarity threshold in the similar clique computation step is selected as 0.7 for both melody and lyrics.

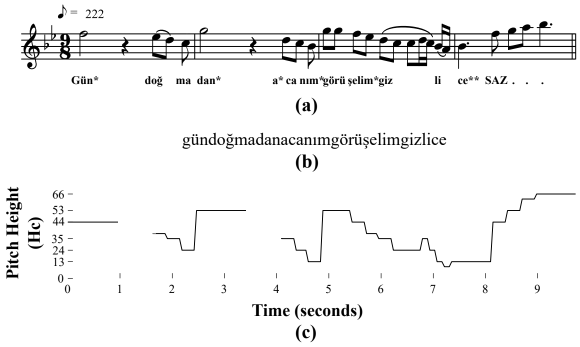

Figure 2: A short excerpt from the score of the composition, *Gel Güzelim*. **a)** The score, **b)** the lyrics, **c)** the synthetic pitch computed from the note symbols and durations. The spaces in the end of the syllables are displayed as *s.

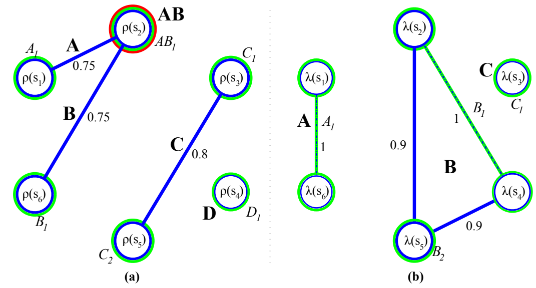

Figure 3: The graphs, the cliques and the semiotic labels obtained from the mock example (Figure 1) using an edge weight threshold of 0.7 for both melody and lyrics. The circles represent the nodes and the lines represent the edges of the graphs, respectively. The edge weights are shown next to the lines. Green, blue and red colors represent the unique cliques, the similar cliques and the intersection of similar cliques, respectively. The semiotic label of each similar clique and each intersection is shown in bold and the semiotic label of each unique clique is shown in italic, respectively.

undirected edges. The weight of an edge connecting two structural elements f_i and f_j is equal to $\hat{L}\left(\rho\left(f_i\right), \rho\left(f_j\right)\right)$ in the melodic relation graph and $\hat{L}\left(\lambda\left(f_i\right), \lambda\left(f_j\right)\right)$ in the lyrics relation graph, respectively. Next, we remove the edges with a weight less than a constant similarity threshold $w \in [0, 1]$. In Section 5, we will investigate the effect of using different w values.

Given the graph, we obtain the groups of structural elements having similar strings by finding the maximal cliques in the graph (Tomita et al., 2006). A maximal clique is a subgraph, which has its each node connected to each other and it cannot be extended by including another node. We denote these cliques as $v_j \in V$, where V is the set of "similar cliques." We additionally compute the maximal cliques of the graph only considering the edges with zero weight. These cliques show us the groups of structural elements, which have exactly the same string. We call each of these cliques as "unique clique" $u_k \in U$, where U is the set of the unique cliques. Note that two or more similar cliques can intersect with each other. Such an intersection resembles all the relevant similar cliques. We denote these "intersections" as $w_l \in W$, where W is the set of intersections between different similar cliques. Also, $\eta\left(x\right)$ denotes the nodes of an arbitrary graph x. Here we would like to to remark a few relations:

- A unique clique is a subgraph of at least one similar clique, i.e. $\forall u_k \in U, \exists v_j \in V \mid \eta\left(u_k\right) \subseteq \eta\left(v_j\right)$.
- A unique clique cannot be a subgraph of more than one intersection, i.e. $\forall u_k \in U, \nexists\{w_l, w_m\} \subseteq W \mid \eta\left(u_k\right) \subseteq \eta\left(w_l\right) \wedge \eta\left(u_k\right) \subseteq \eta\left(w_m\right)$.
- A structural element belongs to only a single unique clique, i.e. $\forall f_i \in F, \exists! u_k \in U \mid \eta\left(f_i\right) \subseteq \eta\left(u_k\right)$.

Figure 3 shows the graphs computed from the sections of the mock example introduced in Figure 1. In the melodic relations graph, each section forms a unique clique since the melody of each section is not exactly the same with each other. Using a similiarty threshold of 0.7, we found four similar cliques formed by $\{s_1, s_2\}$, $\{s_2, s_6\}$, $\{s_3, s_5\}$, $\{s_4\}$. Notice that $\{s_4\}$ is not connected to any clique. so it forms both a unique and a similar clique. Moreover, s_2 is a member of both the first and the second similar cliques and hence it is the intersection of these two cliques. For the lyrics, there are four unique cliques, formed by the sections $\{s_1, s_6\}$ (aka. instrumental sections), $\{s_2, s_4\}$, $\{s_3\}$ and $\{s_5\}$. The lyrics of s_5 is very similar to $\{s_2, s_4\}$ and they form a similar clique composed of these three nodes and the relevant edges.

4.5 Semiotic Labeling

After forming the cliques, we use semiotic labeling explained in Bimbot et al. (2012) to describe the structural elements. First we label similar cliques with a base letter ("A", "B", "C", ...). Then we label the intersections by concetanating the base letters of the relevant similar cliques (e.g. "AB", "BDE", ...). We finally label each unique clique with the label of the relevant intersection, if exists and with respect to the relevant similar clique otherwise, plus a number according to the occurence order of the clique in the score. Right now, we only use the simple labels (e.g. "A_1", "A_2", "AB_2") as termed by Bimbot et al. (2012) to label the unique cliques.

The pseudocode of the process is given in Algorithm 2. During labeling, V, W and U are sorted with respect to the index of their first occurence in the score. We denote the label of an arbitrary element x as $\Lambda(x)$. In the algorithm, we also use iterators $\#(v_j)$ and $\#(w_l)$ for each similar clique v_j and each intersection w_l, which are used to assign the numerical index to each unique clique $u_k \in U$ according its relation with the relevant similar clique or intersection.

Algorithm 2 Semiotic labeling

```
λ ←"A"                          ▷ Start the base letter from "A"
#(v_j) ← 1, ∀v_j ∈ V             ▷ Init. the iterators for all v_j
#(w_l) ← 1, ∀w_l ∈ W             ▷ Init. the iterators for all w_l
for v_j ∈ sort(V) do             ▷ Label similar cliques
    Λ(v_j) ← λ
for w_l ∈ sort(W) do             ▷ Label intersections
    Λ(w_l) ← concat. Λ(v_j), ∀(v_j) | η(w_l) ⊆ η(v_j)
for u_k ∈ sort(U) do             ▷ Label unique cliques
    if ∃w_l | η(u_k) ⊆ η(w_l) then        ▷ e.g. "ACD_1"
        Λ(u_k) ← Λ(w_l)_#(w_l)
        #(w_l) ← #(w_l) + 1
    else                                  ▷ e.g. "C_2"
        Λ(u_k) ← Λ(v_j)_#(v_j) | η(u_k) ⊆ η(v_j)
        #(v_j) ← #(v_j) + 1
for f_i ∈ F do                   ▷ Label structural elements
    Λ(f_i) ← Λ(u_k) | η(f_i) ⊆ η(u_k)
```

The label of each section of the mock example is shown below the staff in Figure 1. The same semiotic labels are also shown on the computed graphs in Figure 3. Notice that the melodic semiotic label of s_6 is B_1 because the first occurence of the relevant similar clique is at s_2.

By extracting the relations in the graphs computed from the melodic and lyrics similarity matrices (Section 4.4) and then applying semiotic labeling to each section and phrase according to its relation, we obtain a $< Melody, Lyrics >$ tuple for each section and phrase (Section 4.5). For each phrase we additionally mark the sections, which enclose and/or overlap with the phrase. Appendix A shows the results of the structural analysis applied to the score "Kimseye Etmem Şikayet." We leave the examination of the analysis to the readers as an exercise.

5. EXPERIMENTS

In (Bozkurt et al., 2014) report the evaluation of the phrase segmentation method (Section 4.2) on an earlier and slightly smaller version of the annotations that we use to compute the segmentation model. We refer the readers to (Bozkurt et al., 2014) for the evaluation of the training data. Furthermore, the labels of the automatic phrase segmentations need to be validated by musicologists parallel to the discussions brought by Bozkurt et al. (2014). For this reason, we leave investigating the effects of the similarity threshold w in phrase analysis as future research.

To observe the effect of the similarity threshold in the melodic and lyrical relationship extraction (Section 4.4), we have collected a small dataset from the SymbTr collection. The test dataset consists of 23 vocal compositions in the şarkı form and 42 instrumental compositions in peşrev and sazsemaisi forms. These three forms are the most common forms of the classical OTMM repertoire. Moreover their sections are well-defined within music theory; the two instrumental forms typically consists of four distinct "hane"s and a "teslim" section, which follow a verse-refrain-like structure; the sections of the şarkıs typically coincide with the poetic lines. In our initial experiments we focused on şarkıs with the poetic organization "zemin, nakarat, meyan, nakarat," which is one of the most common poetic organization observed in the şarkı form. Using the automatically extracted section boundaries (Section 4.1) as the ground-truth, the first author has manually labeled the sections in the scores with the same naming convention explained in Section 4.5.[9] Due to lack of data and concerns regarding subjectivity, we leave the evaluation of section boundaries as future research.

We have conducted section analysis experiments on the test dataset by varying the similarity threshold from 0 to 1 with a step size of 0.05. After the section labels are obtained, we compare the semiotic melody and lyrics labels with the annotated labels. We consider an automatic label as "True," if it is exactly the same with the annotated label and "False," otherwise. For each score, we compute the labeling accuracy for the melody and the lyrics separately by dividing number of correctly identified (melody or lyrics) labels with the total number of sections. We additionally mark the number of similar cliques and its ratio to the unique cliques obtained for each score. For each experiment, we find the average accuracy for the similarity threshold w by taking the mean of the accuracies obtained from each score.

Figure 4 shows the notched boxplots of the accuracies, the total number of similar cliques and the ratio between the number of unique cliques and the number of similar cliques obtained for each similarity threshold. For the melody labels, the best results are obtained for the similarity threshold values between 0.55 and 0.80 and the best accuracy is 99%, when w is selected as 0.70. For lyrics labeling, any similarity value above 0.35 yields near perfect results and 100% accuracy is obtained for all the values of w between 0.55 and 0.70. In parallel, the number of similar cliques and the ratio between the unique cliques and the similar cliques gets flat in these regions. From these results we select the optimal w as 0.70 for both melodic and lyrical similarity.

[9] The experiments and results are available at `https://github.com/sertansenturk/otmm-score-structure-experiments/releases/tag/fma_2016`

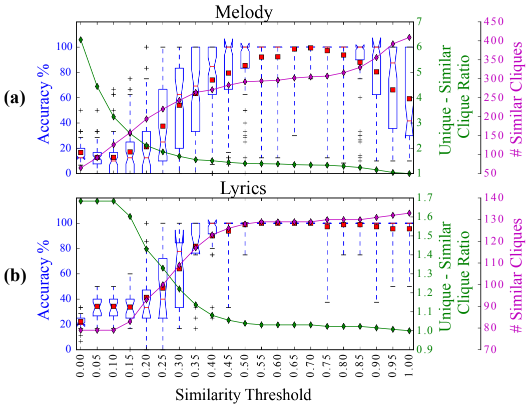

Figure 4: The notched boxplots of the accuracies, number of similar cliques and the ratio between the number of unique cliques and similar cliques obtained for **a)** the melody labels and **b)** the lyrics labels (only for vocal compositions) using different similarity thresholds. The squares in the boxplots denote the mean accuracy.

6. DISCUSSION

As shown in Section 5, the similarity threshold w has a direct impact on the structure labels. A high threshold might cause most of the similar structural elements regarded as different, whereas a low threshold would result in many differences in the structure disregarded. In this sense the extreme values of w (around 0 or 1), would not provide any meaningful information as $w = 0$ would result in all the structures being labeled similar and $w = 1$ would be output all the structures as unique. We also observe that the melodic similarity is more sensitive to value of w than lyrics similarity. This is expected as the strings that make up the lyrics are typically more diverse than the note symbols used to generate the synthetic pitch. In our experiments we found the optimal value of w as 0.7 for the small score dataset of compositions in the peşrev, sazsemaisi and şarkı forms. Moreover we observe that the curves representing the number of similar cliques and the ratio between the unique cliques and the similar cliques are relatively flat around the same w value, where we obtain the best results (Figure 4). This implies that there is a correlation between decisions of the annotator and our methodology.

Nevertheless, we would like to emphasize that the w value found above should not be considered as a general optimal. First of all, the sections were annotated by a single person and therefore our evaluation does not factor in the subjectivity between different annotators. Second, the section divisions in different forms are much different from the forms we have experimented upon, which might influence the structure similarity. For example, we expect many vocal compositions of OTMM with"terennüm"s (repeated words with or without meaning such as "dost," "aman," "ey") need a lower similarity threshold in the lyrics relationship computation step. Moreover the poetic lines might not coincide with melodic sections in many vocal compositions especially in folk music genre. Third, the threshold can be different in different granularities. For example, the phrases are much shorter than the sections as can be seen in Appendix A. The human annotators might perceive the intra-similarity between sections and phrases differently.

7. APPLICATIONS

We have implemented the structural analysis methodology in Python and integrated it to the *symbtrdataextractor* package, a SymbTr-score parser written by us. [10] We have also forked the open automatic phrase segmentation package by Bozkurt et al. (2014), which is written in MATLAB scripting language. The fork modularizes the code and packages it into a standalone binary so it can be integrated to other tools without the need of a MATLAB proprietary license. Moreover, the code is optimized such that it performs considerably faster than the original code. [11] We have been using the information extracted from the structural analysis in several applications:

Score collection analysis: Using the optimal similarity threshold ($w = 0.7$), we applied structural analysis on the latest release of the SymbTr collection (Section 2). We have extracted and labeled 49259 phrases from 1345 scores, which have both their makam and usul covered in the phrase segmentation training model. Because there is no training data for the usul variants "Yürüksemai II", "Devrihindi II", "Müsemmen II", "Raksaksağı II", "Devrituran II" and "Kapalı Curcuna," we treat them as the most common variant of the same usul, namely "Yürüksemai", "Devrihindi", "Müsemmen", "Raksaksağı", "Devrituran" and "Curcuna". In parallel, 21569 sections are extracted from 1771 scores. [12] The data can be further used to study the structure of musical forms of OTMM.

Automatic score validation: Structural analysis, along with the other functionalities of the *symbtrdataextractor* package are used in unittests applied to SymbTr collection in a continuous integration scheme to automatically validate the contents of the scores. [13]

Score format conversion: We are currently developing tools in Python to convert the SymbTr-txt scores to the MusicXML format [14] and then to the LilyPond format [15] to improve the accesibility of the collection from popular music notation and engraving software. The converters use the information obtanined from *symbtrdataextractor* to add the metadata and the section names in the converted scores.

Audio-score alignment: In the performances of OTMM compositions, the musicians occasionally insert, repeat and omit sections. Moreover they may introduce musical passages, which are not related to the composition (e.g. im-

[10] https://github.com/sertansenturk/symbtrdataextractor/
[11] The fork is hosted at https://github.com/MTG/makam-symbolic-phrase-segmentation
[12] The data is available at https://github.com/sertansenturk/turkish_makam_corpus_stats/tree/66248231e4835138379ddeac970eabf7dad2c7f8/data/SymbTrData
[13] https://travis-ci.org/MTG/SymbTr/
[14] https://github.com/burakuyar/MusicXMLConverter
[15] https://github.com/hsercanatli/makam-musicxml2lilypond

provisations). In (Şentürk et al., 2014), we have proposed a section-level audio-score alignment methodology proposed for OTMM, which considers such structural differences. In the original methodology the sections in the score are manually annotated with respect to the melodic structure. Next, the candidate time intervals in the audio recording are found for each section using partial subsequence alignment. We replaced the manual section annotation step with the automatic section analysis part of our alignmet method, where we use the melody labels to align relevant audio recordings and music scores. Using the modified method we have aligned the related audio and score pairs in the CompMusic Turkish makam music corpus (Uyar et al., 2014) and linked 18.770 sections performed in 1767 pairs of audio recordings and music scores. The aligned audio-score pairs are accessible via *Dunya makam*, our prototype web application for the discovery of OTMM (Şentürk et al., 2015). [16] In the application, the audio can be listened synchronous to the related music score(s) on the note-level and the sections are displayed on the audio timeline.

We have additionally conducted experiments using the melodic relations of the extracted phrases. Our preliminary results suggest that phrase-level alignment may provide better results than section-level alignment.

8. CONCLUSION

We proposed a method to automatically analyze the melodic and lyrical organization of the music score of OTMM. We applied the method on the latest release of the SymbTr collection. We extracted 49259 phrases from 1345 scores and 21569 sections from 1771 scores. We are also using the extracted structural information in automatic score validation, score engraving and audio-score alignment tasks.

In the future, we would like to test other string matching and dynamic programming algorithms (Serrà et al., 2009; Şentürk et al., 2014) in general, for similarity measures with different constraints and select the optimal similarity threshold w automatically according to the melodic and lyrical characteristics of the data. We would also like to solidify our findings by working on a bigger dataset annotated by multiple experts and cross-comparing the annotated and the automatically extracted boundaries as done in (Bozkurt et al., 2014). Our ultimate aim is to develop methodologies, which are able to describe the musical structure of many music scores and audio recordings semantically and on different levels.

9. ACKNOWLEDGEMENTS

We are thankful for Dr. Kemal Karaosmanoğlu for his efforts in the SymbTr collection, Dr. Barış Bozkurt for his suggestions on the phrase segmentation and Burak Uyar and Hasan Sercan Atlı for developing the score conversion tools in Python. This work is partly supported by the European Research Council under the European Union's Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583).

[16] `http://dunya.compmusic.upf.edu/makam`

10. REFERENCES

Bimbot, F., Deruty, E., Sargent, G., & Vincent, E. (2012). Semiotic structure labeling of music pieces: Concepts, methods and annotation conventions. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, (pp. 235–240)., Porto, Portugal.

Bozkurt, B., Karaosmanoğlu, M. K., Karaçalı, B., & Ünal, E. (2014). Usul and makam driven automatic melodic segmentation for Turkish music. *Journal of New Music Research*, *43*(4), 375–389.

Cambouropoulos, E. (2001). The local boundary detection model (lbdm) and its application in the study of expressive timing. In *Proceedings of the International Computer Music Conference*, (pp. 17–22).

Jackendoff, R. (1985). *A generative theory of tonal music*. MIT Press.

Karaosmanoğlu, K. (2012). A Turkish makam music symbolic database for music information retrieval: SymbTr. In *Proceedings of 13th International Society for Music Information Retrieval Conference (ISMIR)*, (pp. 223–228)., Porto, Portugal.

Karaosmanoğlu, M. K., Bozkurt, B., Holzapfel, A., & Doğrusöz Dişiaçık, N. (2014). A symbolic dataset of Turkish makam music phrases. In *Proceedings of 4th International Workshop on Folk Music Analysis*, (pp. 10–14)., Istanbul, Turkey.

Lartillot, O. & Ayari, M. (2009). Segmentation of Tunisian modal improvisation: Comparing listeners' responses with computational predictions. *Journal of New Music Research*, *38*(2), 117–127.

Lartillot, O., Yazıcı, Z. F., & Mungan, E. (2013). A pattern-expectation, non-flattening accentuation model, empirically compared with segmentation models on traditional Turkish music. In *Proceedings of the 3rd International Workshop on Folk Music Analysis*, (pp. 63–70)., Amsterdam, Netherlands.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, (pp. 707–710).

Pearce, M. T., Müllensiefen, D., & Wiggins, G. A. (2010). Melodic grouping in music information retrieval: New methods and applications. In *Advances in music information retrieval* (pp. 364–388). Springer.

Şentürk, S., Ferraro, A., Porter, A., & Serra, X. (2015). A tool for the analysis and discovery of Ottoman-Turkish makam music. In *Extended abstracts for the Late Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain.

Şentürk, S., Holzapfel, A., & Serra, X. (2014). Linking scores and audio recordings in makam music of Turkey. *Journal of New Music Research*, *43*(1), 34–52.

Serrà, J., Serra, X., & Andrzejak, R. G. (2009). Cross recurrence quantification for cover song identification. *New Journal of Physics*, *11*(9).

Tomita, E., Tanaka, A., & Takahashi, H. (2006). The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, *363*(1), 28 – 42.

Uyar, B., Atlı, H. S., Şentürk, S., Bozkurt, B., & Serra, X. (2014). A corpus for computational research of Turkish makam music. In *1st International Digital Libraries for Musicology Workshop*, (pp. 57–63)., London, United Kingdom.

A. EXAMPLE ANALYSIS

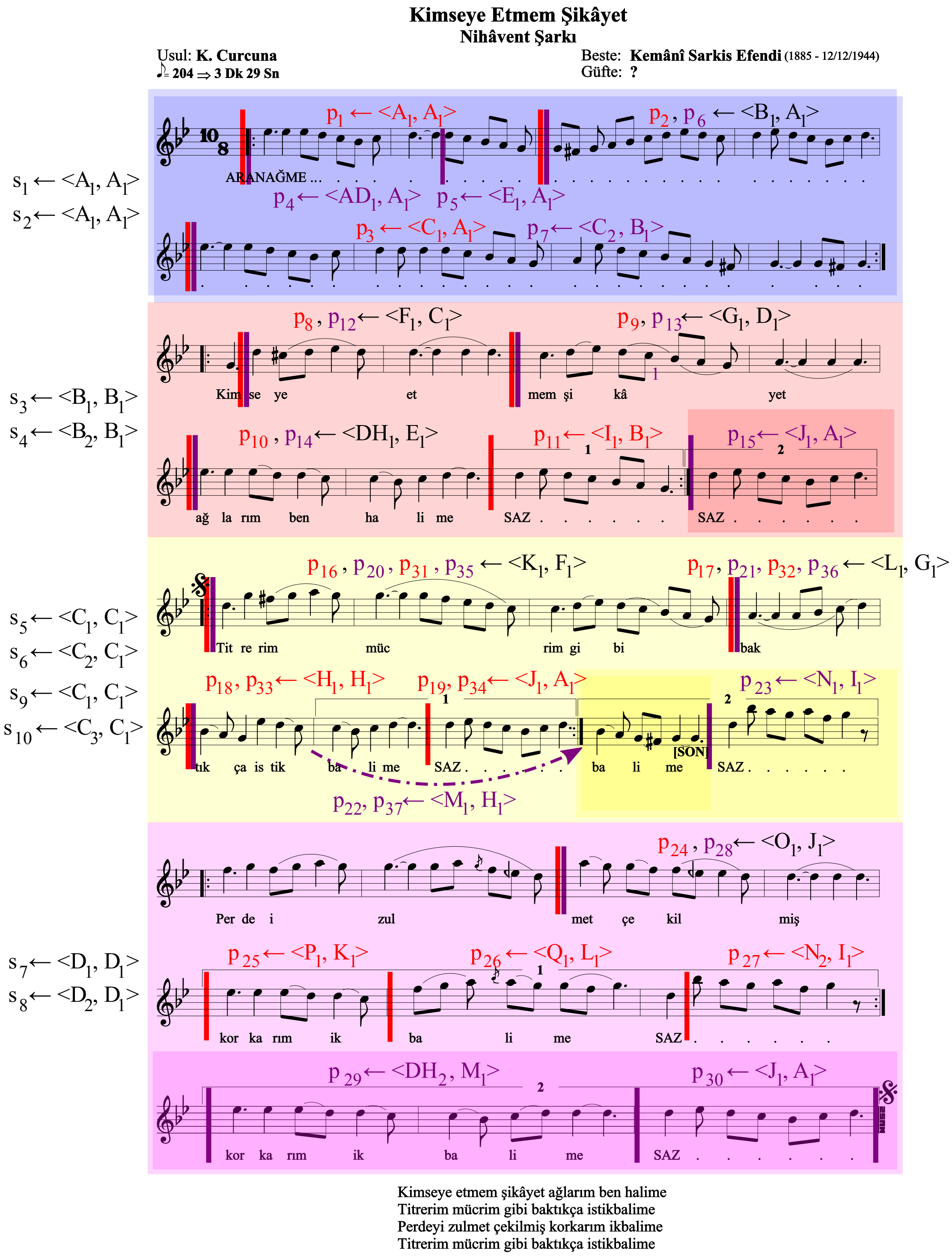

Figure 5: The results of the automatic structural analysis of the score "Kimseye Etmem Şikayet." The sections are displayed in colored boxes with the volta brackets colored with a darker shade of the same color. The section labels and their semiotic $< Melody, Lyrics >$ label tuple is shown on the left. The phrase boundaries are shown as red lines for the first and as purple for the second pass. The phrases and their semiotic labels are shown on top of the relevant interval and on the bottom, when there are differences in the boundaries in the second pass. Note that s_5, s_6, s_9 and s_{10} are the repetitive poetic lines (tr: "Nakarat"). "[Son]" in the end of the "Nakarat" marks the end of the piece. The similarity threshold is taken as 0.7 for both melody and lyrics. The usul of the score is *Kapalı Curcuna*, which we treat as *Curcuna* in the phrase segmentation step.