MUSE: Visual Analysis of Musical Semantic Sequence

Baofeng Chang, Guodao Sun, Tong Li, Houchao Huang, and Ronghua Liang

Abstract— Visualization has the capacity of converting auditory perceptions of music into visual perceptions, which consequently opens the door to music visualization (e.g., exploring group style transitions and analyzing performance details). Current research either focuses on low-level analysis without constructing and comparing music group characteristics, or concentrates on high-level group analysis without analyzing and exploring detailed information. To fill this gap, integrating the high-level group analysis and low-level details exploration of music, we design a musical semantic sequence visualization analytics prototype system (MUSE) that mainly combines a distribution view and a semantic detail view, assisting analysts in obtaining the group characteristics and detailed interpretation. In the MUSE, we decompose the music into note sequences for modeling and abstracting music into three progressively fine-grained pieces of information (i.e., genres, instruments and notes). The distribution view integrates a new density contour, which considers sequence distance and semantic similarity, and helps analysts quickly identify the distribution features of the music group. The semantic detail view displays the music note sequences and combines the window moving to avoid visual clutter while ensuring the presentation of complete semantic details. To prove the usefulness and effectiveness of MUSE, we perform two case studies based on real-world music MIDI data. In addition, we conduct a quantitative user study and an expert evaluation.

Index Terms—Musical semantic sequence, semantic analysis, temporal sequence, feature extraction

1 Introduction

Temporal sequences are omnipresent in many scenarios (e.g., temporal event data, time-series mobility data, and musical note sequence). Generally, temporal sequence data contains rich information such as changeable structures and semantic information [8], [13]. For example, in a musical sequence, a piece of music can be abstracted into one single sequence or multiple parallel sequences with multi-dimension information for each element (e.g., music genre, pitch value, and instrument type) [4], [27]. These domain-specific contexts not only provide abundant semantic information for temporal sequence data, but also bring challenges to data exploration and analysis.

Many applications focus on assisting people in visualizing and understanding dynamic changes in sequence data with contextual information. For example, a music novice can identify a comprehensive understanding of music fields from a complete picture of multi-field music sequences. In terms of music experts, they can compare their musical works in a music database to absorb the similarities, differences, and new musical ideas [9], [14]. Furthermore, because most sequence datasets have similar primary forms (i.e., temporal data with numerical/categorical attributes), evaluating sequence characteristics and developing appropriate tools in a single subject could help other fields.

In this study, we use music sequences as experimental data for semantic exploration and analysis. However, understanding music sequence data is hampered by two main challenges: (1) quantitative model and measurement of music sequence data and (2) interactive visualization of the measured results. Current studies have researched quantitative analysis of music sequence data [40], [42]. These methods generally ignore order uncertainty of musical notes or cannot fully incorporate other attributes such as instruments or genre in musical representation. Hence, how to produce a rational and complete quantification result of musical sequence data is the first challenge in musical sequence analysis. Secondly, a music sequence can be regarded as a temporal event with numerical/categorical information. Existing studies either display an overall distribution of the whole sequence data neglecting the detailed comparison, or present a series of sequences ignoring the overall context information. Consequently, how to combine overview and detailed exploration based on music sequences is another challenge in analyzing music sequences.

This study presents a visual analysis prototype system (MUSE) to facilitate the exploration and analysis of multidimensional music semantic sequence data. We incorporate a machine learning model (ML) and a heuristic method for transmitting the characteristics and relations of musical sequences. Each music sequence is equipped with a dense vector representation based on its computational text representation. However, the black box feature of ML models may hinder the exploration and analysis of music sequences. Thus, rationally displaying the complete semantic details and statistical information of music sequences is needed in this work. To help users dig out insights into the overall characteristics and semantic change patterns of the sequence data, MUSE projects sequence vectors into a 2D space called "distribution view" emphasizing music genre and instruments. The distribution view equips with a

• *Baofeng Chang, Guodao Sun, Tong Li, and Ronghua Liang are with College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. E-mail: baofeng.chang@zjut.edu.cn, guodao@zjut.edu.cn, tongli@zjut.edu.cn, and rhliang@zjut.edu.cn. Houchao Huang is with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. E-mail: houchaohuang@zjut.edu.cn.*

density contour and a pathfinding component. The density contour considers the geometry position of points and their similarities to display the distribution of musical sequences, and the pathfinding component can help users discover the transition patterns among specific musical sequences or groups. To help users understand and distinguish musical sequence patterns, MUSE presents these sequences in a sequential chart called “semantic detail view”, displaying the semantic details such as notes’ and chords’ semantic information (e.g., pitch, instrument, and duration) of musical sequences. The semantic detail view presents the complete semantic information of music sequences considering the elimination of visual clutter, which helps users compare and explain sequences at the micro-level. Additional assistant views in the MUSE include a genre tendency view, a sequence connection graph, and a parallel coordinate plot to help users drill down to the exploration of music sequences and their corresponding genres. The main contributions of our work are summarized as follows:

- We present a visualization workflow for interactive analysis of multi-dimensional sequence data incorporating the ML model and heuristic idea.
- We design MUSE, a visual analysis system that can assist analysts in investigating the complex distribution of the musical sequence data.
- We offer profound insights into music by employing MUSE to explore large-scale music data covering multiple musical semantics.

2 Related Work

In recent years, researchers have employed visual techniques in music domain to mine musical insights such as genre evolution, note features, and emotion analysis. Many works transfer music data to sequence data for mining the music insights based on sequence analysis methods [2], [6]. Visualization systems are also proposed to help users analyze music semantic information [16], [46]. To investigate and summarize the research status related to our work, we conduct a literature review on sequence analysis, sequence visualization, and music visualization.

2.1 Sequence Analysis

Sequential analysis receives considerable attention in data mining, medical research, and visualization, etc. Unraveling meaningful and significant time structures from large-scale sequence data is a fundamental problem [57]. Liu et al. proposed temporal skeletonization to proactively reduce the effort of sequence representation [29], and they further improved it to reduce the cardinality of representation [30]. In medical research, capturing the similarity between nucleotides (DNA/RNA) or amino acids (protein) and combining with multiple sequence alignments are vital to achieve visual alignment of gene sequences [53].

Similarly, the essence of musical melody is a series of notes that form a multi-variate sequence. Many works research music sequences in a similar manner. For example, Needleman et al. proposed an algorithm that finds an optimal alignment between two sequences [38]. Mongeau et al. applied alignment algorithms to the musical sequence data; they used an extended version of the Needleman-Wunsch algorithm and considered both pitch and time [37]. Gasser et al. proposed Needleman-Wunsch Time Warping (NWTW), a pure dynamic programming method for aligning music recordings, which contain structural differences [15]. This method optimizes the faults of traditional alignment methods such as the dynamic time warping and Needleman–Wunsch algorithm. The Hidden Markov model is another popular method for sequence analysis. Chai et al. clarified folk music from different countries based on monophonic melodies using the Hidden Markov models [3]. Kness et al. summarized and evaluated the multiple similarities of music context to help people select a rational similarity in different scenes [23].

The study of sequence alignment and difference is the basis for analyzing the similarity of the musical melody. Typical recent works concentrate on the visual exploration of melodic similarity [9], [50], [51]. De Prisco et al. visualized and evaluated the plagiarism of musical works; they calculated similarity through pitch difference and designed three visualization forms to display the characteristics, similarity intervals, and plagiarism of music works. This study helps users understand the complex melodic similarity and plagiarism among songs effectively [9]. Walshaw et al. proposed the TuneGraph, which generates the *abc notation* corpus similarity graph based on the similarity measurement of melodies [50]. A subsequent study aims to explore the relationships at a global (corpus-based) level [51]. At first, this work uses multi-level recursive subsequence alignment to calculate the similarity, and a complete weighted graph is generated based on the dataset. Then, a corpus graph is implemented by matching thresholds. Subsequently, the authors proposed an optimization algorithm for the corpus graph to improve the global visualization effect.

2.2 Sequence Visualization

Sequence visualization is an effective method for analyzing multi-variate sequences. Plaisant et al. proposed the LifeLines to visualize personal timeline event sequences [41]. The LifeLines displays an event sequence completely as an overview, and users can focus on their interested event details. However, the LifeLines lacks a comparison function among multiple event sequences. Gotz et al. proposed the DecisionFlow to visualize high-dimensional temporal event sequences [17]. In the DecisionFlow, a rectangular flowchart is implemented to display temporal events and a scatter plot is used to show comparison results. Baur et al. combined listening records and life event stream data (photos and calendar events) to produce a temporal data. Based on the produced temporal data, they used a vertical scatter plot to display listening stories [10].

Zhang et al. designed IDMVis and examined the effects of different alignment schemes used in visualizing event sequences [55], [56]. Mathisen et al. presented a method to offer better aggregation results and help users identify insights into event sequences [34]. Khoa et al. integrated gradient vector analysis flow in aligning multiple sequences to help users dig out patterns in large-scale multiple sequence alignment data [39]. Guo et al. proposed EventThread to offer a visual summarization and latent stage analysis function of the large-scale and high-dimensional event sequence data [19].

The preceding related works analyze music sequences from various aspects and expand research targets into the sequence alignment, comparison, and representation with corresponding information. However, most of the studies only focus on analyzing limited-level sequence information while ignoring the comparative analysis of multiple sequences or comprehensive exploration of pattern insights. Consequently, integrating a comparison function among multiple sequences should be supported during displaying detailed semantic information of music sequences.

2.3 Music Visualization

In music visualization works, verctorization of music has received a considerable attention in music recommendation, music similarity measurement, and music genre classification [22], [28]. Cheng et al. proposed a method to explore music performance sequence (MPS) information in the Matrix Factorization (MF) to improve the performance of music recommendations [5]. Inspired by the Word2Vec techniques, a Song2Vec method is proposed to compute the similarity between songs in the work. Yu et al. proposed a deep multimodal correlation learning architecture involving deep two-pronged neural networks for the audio and textual modality (lyrics) [54]. This work is the first study which uses deep architectures to learn the temporal correlation between audio and lyrics. A pre-trained Doc2vec model followed by fully connected layers is used to generate vectors of songs' lyrics. However, the works mentioned above only focus on exploring and analyzing in a local scope while ignoring the promotion of music vectorization to a global level liking the works of Lopez et al. [31] and Valle et al. [48]. We utilize the Doc2Vec [25] to generate vectors of music sequences and perform dimension reduction with the t-distributed stochastic neighbor embedding (t-SNE) [33] to produce distribution results of music sequences. We further combine music genre and instrument information in the distribution results to support exploration and analysis of global distribution patterns.

Researchers have studied different visualization techniques to represent different-level information of music in recent years. Chan et al. proposed an innovative visualization solution to demonstrate the semantic structure in classical music works, including macro-level layer relations, micro-level theme variations, and macro-micro interactions in the layers and themes [4]. However, this study only focuses on displaying the aesthetics of music works while lacking a view to provide other related information. Lima et al. proposed a visualization system, named SongVis, to represent musical semantic descriptors. The system establishes a prototype view by extracting the abstract features of music, and it apply the familiar emoji as a visual metaphor to display the relevant information of songs [27]. The SongVis aims to bridge the semantic gap between features and visual depictions. However, SongVis only focuses on features while ignoring the visualization of musical melody. Cantareira et al. introduced a visualization framework called MoshViz, and used the note as the base unit for the calculation and the rectangle as the fundamental element of visualization. [2]. Three formulas for calculating rectangle height, rectangle color and transparency are used to quantify the pitch range, the degree of instability of the music, and the number of notes, respectively. Visualization is used in other aspects of the music domain, too. Jänicke et al. proposed a profiling visual analysis system to help users analyze musicians of interests [21]. A rich framework is constructed to reveal the degree of clutter, complexity, similarity, and interval change of musical works.

The preceding work either analyzes the hierarchical structure from the macro perspective, or abstracts the musical features from the micro perspective. There is no systematic combination of the aspects mentioned above is available to provide users with a better approach to musical exploration. In this study, we aim to integrate macro and micro perspectives to present the music genre, instrument, and notes with various pitches, as well as to provide users with a comprehensive analysis of musical sequences.

3 Requirement Analysis & System overview

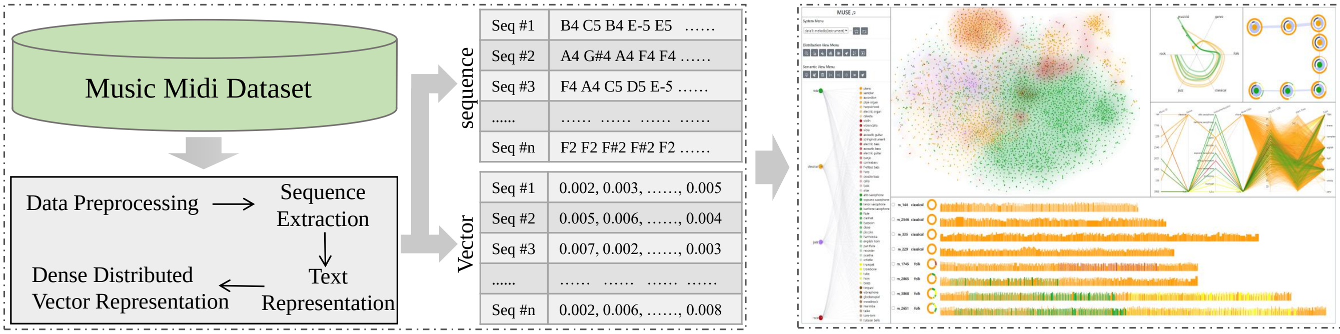
3.1 Requirement Analysis

To strengthen the practicability and generality of this work, we conduct an adequate literature review and discussion with domain experts. Music experts, music amateurs, music novices, and visual analysis experts are invited to propose any requirements, questions, and suggestions about analyzing, comprehending, and visualizing semantic music sequences. Some experts were interviewed by us face to face, while others were online. After interviews and discussions, we summarized four main requirements **(R1-R4)** introduced as follows for guiding our study.

R1 **Representation of sequence characteristic.** How to extract and depict the characteristics of musical sequences in a comparable space? Two aspects can hinder the analysis of the raw musical sequence. First, information on the raw musical sequences is very abstract, mutable, and contextual, thereby causing challenges during representing musical sequences in a numerical form. Secondly, the analysis of raw musical sequences requires substantial background knowledge and experience. In addition, the extracted functions should contain the semantic information as much as possible while maintaining the sequence comparison, which may facilitate further analysis. Therefore, the proper extraction and representation of sequences are vital for the visual analysis of musical sequences.

R2 **Visualization of semantic detail.** How to present the semantic details of musical sequences to help users get the information efficiently? Although the extraction of sequence features simplifies analysis progress, analysts may question the visualization results in case of the insufficient support for raw semantic information. In addition, the raw sequence information is generally dynamic and hierarchical (e.g., a variety of instruments and notes). Hence, proper visualization of sequence semantic details is essential.

R3 **Correlation of hierarchical information.** How to correlate the hierarchical information in the musical sequences? Associating and presenting hierarchical information can help analysts explore the relationships between various attributes. For example, a musical sequence can contain gradually fine-grained hierarchical information including genres, instruments, and notes. However, displaying various hierarchies of information



(a) Sequence and feature extraction **(b) Visualization design and analysis**

Fig. 1. The pipeline of this work includes the following steps: **(a) Sequence and feature extraction** based on ML model and t-SNE-based projection; **(b) Visualization design and analysis** with five views including *distribution view*, *semantic detail view*, *genre instrument tree*, *genre radar plot*, *sequence node-link graph*, and *parallel coordinate plot*.

directly and simultaneously may cause the visual clutter. Hence, integrating the hierarchical information into previous feature representations and detailed visualizations is another challenge.

R4 **Elimination of potential uncertainty.** How to avoid potential uncertainty issues in analyzing musical semantic information? Uncertainty issues may arise in different stages during processing and analyzing the data. For example, the vector-based numerical representations of music sequences are inexplicable due to the features of the ML model (Doc2Vec [25]). On the other hand, using t-SNE to project high-dimensional vectors to 2D space may lose important vectors information. In addition, incorrect visual encodings can lead to erroneous insights in the analysis. Consequently, integrating explainable and heuristic methods to avoid misleading insights caused by uncertainty and assisting users in finding out potential uncertainty issues expediently and quickly could not be neglected in our work.

3.2 Pipeline

In response to the requirement analysis, we propose a pipeline of this work. The pipeline includes (a) sequence and feature extraction and (b) visual design and analysis.

In **sequence and feature extraction** (Fig. 1 (a)), to conduct *data preprocessing* and *sequence extraction*, we first employ a computer-aided musicology toolkit called Music21 [7] to clean noise or repetitive data in the music midi dataset and extract music semantic to form a sequence. Secondly, we extract the *text representation* from the musical sequences to facilitate subsequent mathematical modeling. Thirdly, we employ a well-established model called Doc2Vec to generate a *dense distributed vector representation* of each music sequence to represent the sequence data computably. We use text representations of music sequences to train the Doc2Vec model and compute the vectors of music sequences based on the text representations.

In **visual design and analysis** (Fig. 1 (b)), we employ t-SNE [33] to perform dimension reduction of the musical sequence and project the results into a 2D plane to provide a distribution overview. The *distribution view* has a capability of simultaneously presenting genres, instruments, and relationships of music sequences in an in-place scheme. The *semantic detail view* uses pixel-based bar charts to present the semantic details of the music sequences. The additional visualization views including *genre instrument tree, genre radar plot, sequence node-link graph, and parallel coordinate plot* are designed to help users effectively reveal hidden patterns and detailed semantic information of musical sequences.

4 Feature and Semantic Representation

4.1 Feature Extraction and Expression

4.1.1 Feature extraction

A piece of music is a sequence that contains many notes. Each note in music sequences has its performing characteristics such as instrument, time, group (chord). Dynamically changeable notes and parallel instruments make it possible for musical sequences to express complex semantic information. Traditional methods of feature extraction, clustering, and dimensionality reduction for such temporal sequences are insufficient to extract the semantic information of music sequences. On the other hand, works such as MuseGAN [11] and MuseNet [45] have proven that ML models perform well in feature extraction and application in the music field. Therefore, we employ ML models to generate the feature vectors of musical sequences in the feature extraction stage.

Before extracting the music features, we constructed the semantic sequence and produced the text representation of each music. Music21 can extract performance elements from each music (MIDI files) to generate a music sequence with semantic information [32]. The elements (i.e., notes and chords) in a musical semantic sequence are sorted into a sequential data according to the order rule: (1) start time, (2) instrument type, and (3) pitch value.

After extracting musical semantic sequence, we combine corresponding semantic information to form a word to represent the sequence's element (note or chord). Next, we apply these words to produce the text representations of musical semantic sequences. Each note and chord have its semantic information such as instrument name, pitch value, and type. Fig. 2 (a) shows an example to generate words of a note and a chord. For each note, we construct three kinds of words including "pitch-type, diff-type, instrument-pitch-type" as its text representations. The "diff" is the pitch difference value from the previous element to the current element, which represents the melodic features. For the chord, the "pitch" will be "pitch-pitch-pitch" to represent all notes' pitch in this chord, which is used in the "diff" of the chord's word, too. During calculating the "diff", if the previous element is a chord, the pitch value of this "previous element" is the pitch of the root note, which can be extracted

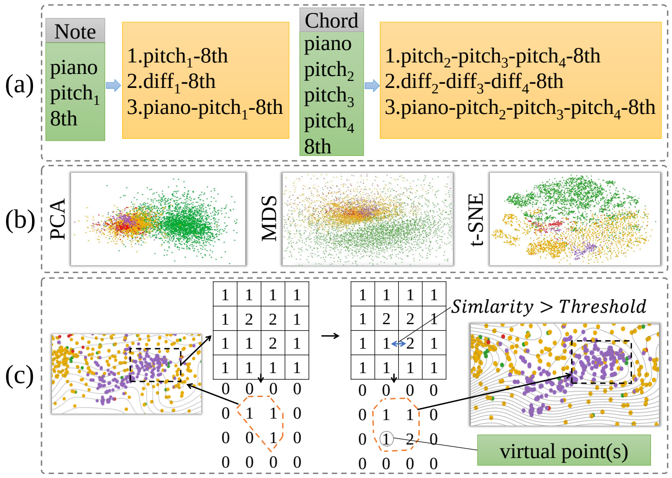


Fig. 2. (a) Four type words are constructed to represent note and chord semantic information. (b) Comparison of PCA, MDS, and t-SNE in dimensionality reduction of sequence vectors. (c) Generation of density contour considering sequence distance and semantic similarity.

by the "chord.root()" function in Music21 library. In the end, we apply these generated words to construct the text representations of music sequences.

Inspired by the idea proposed by Galex [26], which extracts article features based on the article summarization data by using Doc2Vec to generate the feature vectors for articles, we apply Doc2Vec to compute the music vector representations by using music text representation. At first, we use text representations of the music sequences to train the Doc2Vec model. Then, we compute the corresponding feature vector based on the text representation of each music sequence. As a result, each music MIDI file is extracted as a musical semantic sequence, and each musical semantic sequence equips a corresponding 128-dimension feature vector by constructing the text representation and employing the Doc2Vec model.

4.1.2 Feature expression

To visualize the distribution pattern among music more intuitively, we project the result of Doc2Vec to a 2D plane. We have considered three alternative dimension reduction methods includes principal component analysis (PCA) [12], multidimensional scaling (MDS) [24], and and t-distributed stochastic neighbor embedding (t-SNE) [33]. Although these dimension reduction methods are based on different underlying algorithms, we have examined multiple parameter groups to select a proper method to produce the distribution results. The test data includes 6,632 musical sequence vectors, and the size of each vector is 1×128. These vectors are equipped with 4 genre attributes which are represented by 4 color points including yellow (classical), green (folk), red (rock), and purple (jazz).

As shown in Fig. 2 (b), the result of t-SNE has more intuitive distribution patterns than the other two dimension reduction methods (MDS and PCA). Consequently, we select t-SNE to project musical sequences into a distribution view with one theoretical consideration. Compared with MDS and PCA, t-SNE maximally preserves local neighborhoods of each data point (high dimensional vector), which ensures each musical sequence to have its similar neighbors based on the feature vectors in the distribution view [49].

In the distribution view, we use points to represent the musical sequences. To assist users in discovering distribution patterns intuitively, we map two music features (i.e., music genres and instruments) to the points. In addition, to help users recognize the distribution patterns in the distribution view, we equip the distribution view with two visual components: **a density contour component** and **a pathfinding component**.

The density contour is helpful for users to reveal the overall density distribution of the numerous points. However, the traditional density contour only considers the geometric position of points and ignores the similarity. Thus, inspired by the marching square algorithm [20] in the density contour, we propose an extended algorithm to modify the generation of density contours based on the similarity of musical sequence vectors. We apply the similarity to modify the basic matrix in the original marching square algorithm.

As shown in Fig. 2 (c), the left contour is the raw result. In the marching square algorithm matrix, each number represents the number of musical sequences in a grid. Our algorithm calculates and integrates the musical sequence similarities to modify the density value of the original matrix. When the grid similarity is larger than the threshold, these grids tend to be similar so that the number in the two grids would be added with a virtual value to change the state of the matrix. After the steps mentioned above, the modified density contour is generated to ensure that similar sequences have the same contour line. From the density contour shown in the right part of Fig. 2 (c), the points indicating the same musical genre (jazz) are circled by the contours, which may indicate the apparent similarity of these musical sequences.

Including the density contour, we integrate a semantic change path component into the distribution view based on our proposed pathfinding algorithm to help users analyze the semantic change from one pattern to another pattern. The pathfinding algorithm, whose perplexity is O(n^2), integrates the distances of music sequence projection coordinates and feature vectors. This heuristic algorithm follows four steps (**S1** to **S4**) to find a path from the source music sequence to the target one in the distribution view. Four steps are introduced as follows.

S1 Divide music sequences into corresponding grids based on their projection coordinates.

S2 Obtain music sequences in the neighbor search grids of the source music sequence. If the neighbor search grids have no available sequences, enlarge the scope of neighbor search grids to search.

S3 Calculate and determine the smallest Euclidean distance of feature vectors between the source music sequence and each music sequence obtained in step **S2**.

S4 Put the music sequence (found in step **S3**) into the path and make it a new source music sequence. Repeat step **S1** to step **S4** until the target music sequence is in the neighbor search grids.

To ensure each new source music sequence to be approaching the target music sequence, we set the principles of neighbor search grids [44]. Comparing the coordinates of the source music sequence and the target one, as shown in Fig. 3, we summarize eight neighbor search grids schemes. The eight schemes are "upper left, low right, upper right,

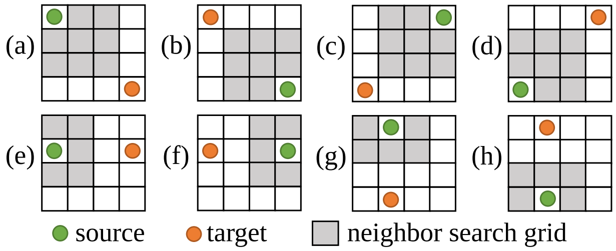


Fig. 3. Eight schemes of neighbor search grids from the source music sequence to the target one.

low left, horizontally left, horizontally right, vertically upper, and vertically low" based on position status from the source music sequence to the target one. Users can sample the path points (sequences) if the path has too many music sequences to analyze the semantic change. In addition, the scheme of grids can be customized in different scenes. In this work, we offer two grids schemes including 128×128 (which is selected in case 2) and 256×256.

4.2 Semantics Representation

Users can identify the sequence distribution patterns in the distribution view and explore hidden patterns interactively. To support a detailed semantic comparison and analysis of patterns, we visualize the semantic details of music sequence in the semantic detail view (Fig. 5 (g)).

Before implementing the semantic detail view, we have conducted six experiments (Fig. 4 (a-f)) to eliminate visual clutter while maintaining semantic information and displaying the semantic details of music sequences. Six experiments for displaying semantic details are respectively raw visualization, height smoothing, instrument reordering, pitch reordering, moving window, notes aggregation.

Raw visualization. In the raw visualization of musical semantic details (Fig. 4 (a)), each small rectangle is mapped as one semantic element (i.e., note or chord) in musical sequences with three attributes (i.e., color, width, and height). The color encodes the instrument type information, the width encodes the playing duration information, and the height represents the pitch value. For the chord, we present the pitch of the chord's root note. Displaying all the notes and chords is a valuable method to visualize the semantic data of musical sequences. However, the visual clutter arises during visualizing semantic information directly. We consider that the visual clutter is due to two reasons. The first reason is that the semantic information in musical sequences is numerous, leading to the minimal width of rectangles. The second reason is that the changeable semantic information such as the instrument type and pitch value are too inconstancy to acquire effectively. To relax the visual clutter, we have conducted the following experiments to present all the semantic details maximumly.

Height Smoothing. To avoid the effect of changeable pitch semantic information, we smooth out the height of the rectangles and map the pitch value to another aspect to eliminate visual clutter. As shown in Fig. 4 (b), we mapped the pitch value to the transparency of the rectangle. However, the irregular transparency brings more visual clutter and an additional analytical burden to the users.

Instruments Reordering. To avoid the visual clutter caused by the changeable instruments, we reorder the instruments in a small area and reconstruct the musical sequence. In this manner, the position of instruments in the entire sequence can be maintained while the visual clutter is reduced. As shown in Fig. 4 (c), the visual clutter is reduced partly, but the changeable pitches still cause visual clutter, particularly with numerous pitches.

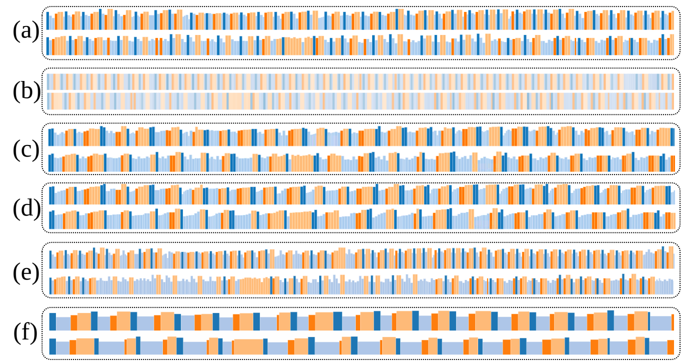


Fig. 4. (a–f) Six visualization attempts are considered to avoid the visual clutter in implementing the semantic detail view.

Pitch Reordering. In order to avoid visual clutter caused by the changeable pitches, we reorder the pitches after rearranging the instruments. As shown in Fig. 4 (d), the notes are represented as a regular form, which may help users explore and compare the semantic information in a small extent. However, when visualizing the long sequences and fitting the entire screen, the rectangles are compressed to estimate a line, which may confuse users. Visual clutter may re-emerge when viewing long music sequences.

Moving Window. Inspired by the idea of "Moving Window" [18], [58], we attempt to remove the instrument with a small proportion in the "Window". This approach can remove the effects from the small-proportion instruments and make the distribution of instruments in the sequence look aggregated. The regional gathering patterns of instruments can reduce the visual clutter caused by adapting to the screen to visualize long sequences. As shown in Fig. 4 (e), instruments show a certain degree of aggregation. However, the small improvement is not sufficient to support the practical expression of long sequences.

Notes Aggregation. To further reduce the visual clutter in visualizing the long sequences, we merge the notes to a new element after ordering the instruments in a collation area. The, we construct a new music sequence by the new elements. The pitch of the new element is the average pitch of the same merged instruments. As shown in Fig. 4 (f), the results can maximumly avoid the visual clutter and preserve the semantic information. MUSE provides users with the interactions such as zooming into the semantic detail view and customizing the collation area width. Users can analyze overall characteristics and semantic details of music sequences with the interactions.

In addition, we propose an algorithm to calculate the width of the collation area to guide the user's customization. To avoid visual clutter while displaying multiple sequences, we assume that the collation area width is proportional to the entropy values of the pitch sequence and position difference of the same instrument, number of instruments, and number of sequences. The considerations are as follows:

- If the entropy of pitches in a sequence is larger, the rectangle height tends to vary more chaotically. As a result, the visual clutter is more likely to occur.

- If the entropy of the index difference sequence based on the instrument position is more significant, the instrument distributes more irregularly in the musical sequence. The color of the rectangle changes varies more chaotically, resulting in the greater possibility to appear visual clutter in visualizing semantic details.
- Additional musical instruments result in more colors in the rectangle, which aggravates the visual clutter. In addition, the number of sequences visualized is more significant, and the visual clutter appears more likely.

Furthermore, we assume that the music with only one instrument does not affect generating the collation area width since the sequence with only one instrument can be presented without visual clutter issue in the semantic detail view. However, a simple proportional relationship is vulnerable to extreme values. Therefore, we introduce the logarithmic function to eliminate the influence of extreme values. The following formulas are proposed to calculate the collation area width:

$$W = m + R(\sum_{i=1}^{m} W_i) \tag{1}$$

where W is the collation area width suggested to users, m is the number of musical sequences with multiple instruments, R is a function to return a number rounded to a given value, and W_i is the collation area width contribution value of music sequence i. The W_i is computed as follows:

$$W_i = \log n * [E(Seq_{pitch}) + \sum_{j=1}^{n} E(Seq^j_{diff})/n] \tag{2}$$

where n is the number of instruments in music sequence i, Seq_{pitch} is the pitches collected from each element of musical sequence i, and Seq^j_{diff} is the sequence of instrument j index difference in the musical sequence i. E is a entropy function of sequences, and it is described as follows:

$$E(X) = -\sum_{x \in X} p(x) \log p(x) \tag{3}$$

where X represents the sequence, and $p(x)$ is the appearance probability of element x in the sequence X.

5 Visual Design

5.1 Design Goals

According to the requirement analysis (Section 3.1), we have discussed design goals before implementing the visualization. The specific design goals are as follows:

G1 Overview of the sequence distribution. The visualization should project music feature vectors into a 2D plane space as a distribution overview of music sequences. In this projection space, users can examine distribution patterns of music sequences and obtain guidance for further analysis (**R1 & R3**).

G2 Representation of the semantic information. The visualization should present the semantic details and encode hierarchical information such as genre, instrument, and note to support effective exploration and comparison of semantic details (**R2, R3 & R4**).

G3 Comparison of multi-variate sequences. The visualization should support users in comparing the semantic information of multivariate sequences, which is helpful for the exploration of hidden semantic change, genre, instruments patterns, etc. In patterns analysis, users should be allowed to explore hierarchical information efficiently and validate whether the potential uncertainty issue is avoided or not (**R3 & R4**).

G4 Exploration of semantic changes. The visualization should allow users to mine the semantic changes, which is helpful for the exploration and analysis of distribution patterns in the projection space (**R1, R3 & R4**).

G5 Support Interactions. The visualization should equip with effective interaction techniques to help users access to data insights expediently. The reason is that integrating appropriate multi-view linked interaction techniques is essential in characteristic mining, pattern analysis, and semantic comparison (**R1 to R4**).

5.2 System Interface

To comprehensively analyze semantic information of music sequences, we design an interactive visual analysis prototype system (MUSE) that follows a series of guidelines of design goals. The MUSE integrates a scatter plot (**distribution view**) to present music sequences, a multiple bar chart (**semantic detail view**) to display semantic details of music sequences, a node tree (**genre instrument tree**) to show the connection between genres and instruments, a radar plot (**genre radar plot**) to introduce genre similarity of each music sequence, a node-link graph (**sequence node-link graph**) to offer statistic information of music sequences, and a **parallel coordinate plot** to help users access to semantic distribution patterns in different semantic dimensions.

5.2.1 *Distribution view*

To help users explore distribution patterns of music sequences, we present a distribution view (**G1**). In the feature extraction step, each music sequence is equipped with a feature vector. Then, we employ t-SNE to project feature vectors into a 2D plane to produce the distribution results of music sequences in the distribution view (Fig. 5 (c)). In this view, each point represents a musical sequence. We encode the point color as two kinds of semantic information including genres or most proportion instruments (**G2**), providing a multi-dimension comparison function. As shown in Fig. 5 (c), the points have four fill colors, which are encoded as the music genres. The yellow points are classical sequences, the green points are folk sequences, the purple points are jazz sequences, and the red points are rock sequences.

To help users discover the hidden patterns of musical sequences, we integrate two components in the distribution view (**G5**). As shown in Fig. 5 (c), one is a density contour considering musical sequence similarity (Section 4.1.2) to enable similar musical sequences to be encircled by the same contour line (**G1**). Another is the pathfinding algorithm integrating coordinates in the scatter plot and similarity of feature vector (Section 4.1.2), which can help users explore semantic change trends between musical sequences (**G4**).

5.2.2 *Semantic detail view*

Visualizing the original details of musical sequences helps users identify the semantic information and explain the

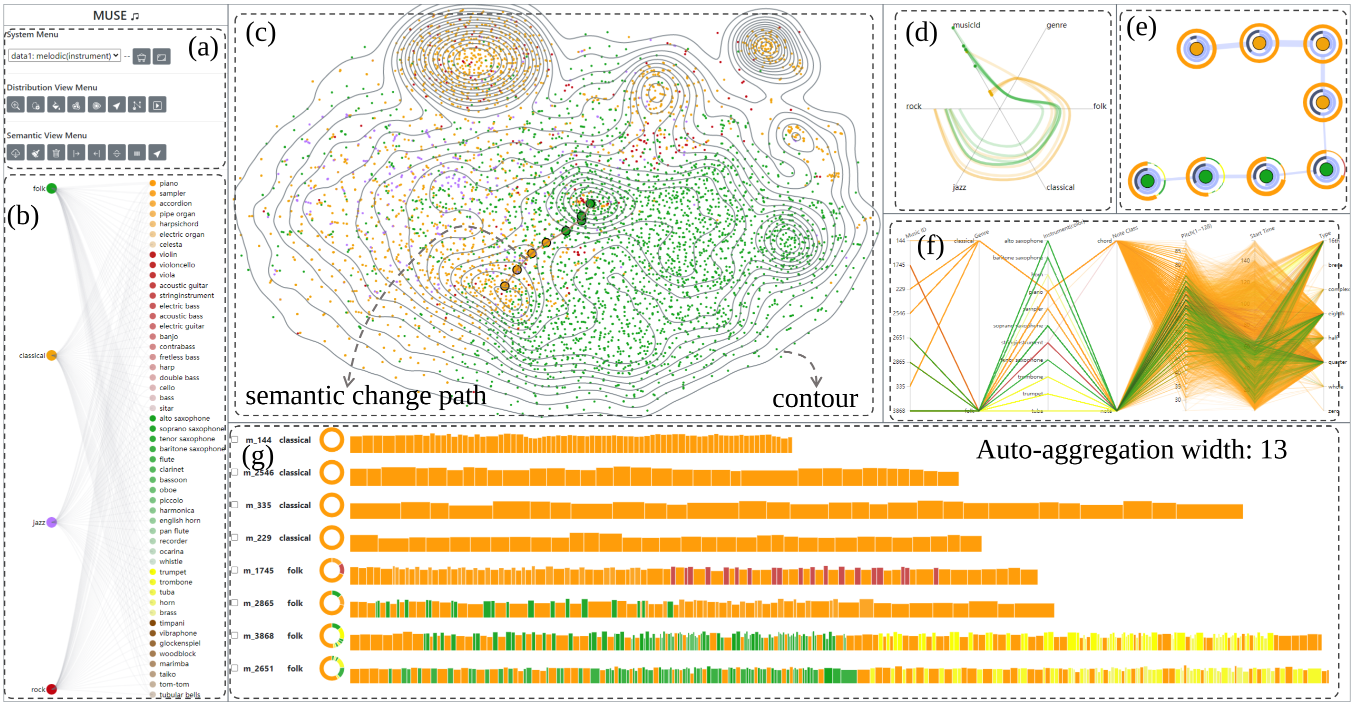


Fig. 5. MUSE includes a system menu (a) to support interaction, a tree plot (b) to present genre instrument connections, a scatter plot (c) to project music sequence, a radar plot (d) to introduce genre similarity of music sequence, a node-link graph (e) to show statistic data of music sequence, a parallel coordinate plot (f) to offer semantic distribution in different dimensions, a bar chart (g) to exhibit semantic details of music sequences.

semantic change insights of music sequences. After the visualization experiments of semantic details (Section 4.2), we utilize the **Notes Aggregation** scheme to present the semantic details of music sequences (**G2 & G3**).

As shown in Fig. 5 (g), a row represents a music sequence respectively, which contains four parts to display the semantic information. The first part is the music sequence ID, which helps users explore the musical sequences in other visualization views. The second part is a genre's text description to provide users with the genre information. The third part is a pie chart component to display the proportion of instruments in the music sequence. The last part presents the semantic details of the music sequence. Each small rectangle represents the semantic information, which has three attributes including the width, height, and fill color. The width indicates the duration, the height represents the average pitch, and the color depicts the type of instruments. As shown in Fig. 5 (g), the music sequence could be aggregated based on the **Notes Aggregation** scheme (Section 4.2) with a collation width of *13*, which is automatically calculated by the Formula (1), (2), and (3).

5.2.3 Genre instrument tree

To present the connection between genres with instruments and assist users in accessing the distribution patterns in the distribution view, we design a genre instrument tree on the left of the system interface (**G1 & G2**).

In the genre instrument tree, as shown in Fig. 6 (a), each genre is displayed as a circle in the first column of this tree plot. For example, the green circle represents the folk genre while the yellow circle represents the classical genre. Each circle in the second column indicates the instrument information. The fill color of the circles indicates the instrument name. All instruments are collected and divided into different categories of instruments (**G2**). Instruments in the same category are encoded as a series of same color class circles with different color saturation. Higher color saturation indicates that the proportion of the instrument is larger than other instruments in the same category. For example, in Fig. 6 (a), two circles filled by two kinds of red colors represent two string instruments (i.e., violin and viola). The higher color saturation indicates that the violin has larger proportion than the viola in music sequences.

5.2.4 Genre radar plot

To help users identify and analyze the similarity to the music genre of each music sequence(**G2**), we design a radar plot (Fig. 6 (b)). The radar plot has six axes/dimensions including *musicID*, *genre*, *folk*, *classical*, *jazz*, and *rock*. Each line in the radar plot represents a music sequence, and the color of the line is encoded as the genre information. The position in the *musicID* axis demonstrates the ID of the music sequence. The position in the *genre* axis indicates the genre information of the music sequence. The positions in the *folk*, *classical*, *jazz*, and *rock* axes present the similarity to music genres. The higher position depicts the larger similarities to the corresponding music genres.

To access the genre similarity of each music sequence, we calculate music genre vectors. For an example of the folk genre vector, we collected all the feature vectors of folk music sequences. Then we average the collected feature vectors of folk music sequences as the folk genre vector. In the end, we calculate the Euclidean distance from the music sequence vector to the folk genre vector as the similarity between the music sequence and the folk genre.

5.2.5 Sequence node-link graph

To help users compare semantic features and explore the semantic change trends, we design a node-link graph to

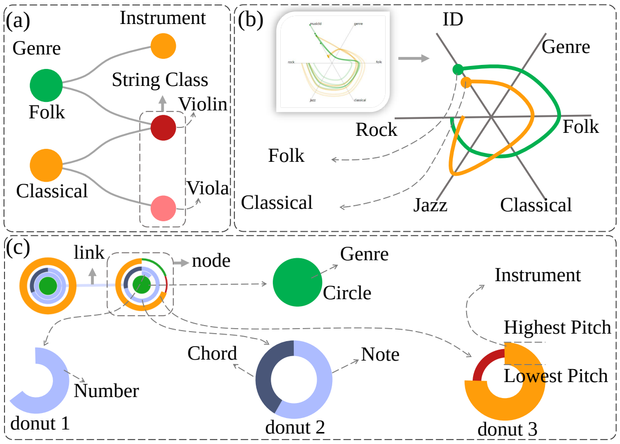


Fig. 6. The sketches include (a) a genre instrument tree to introduce the connections between genres with instruments, (b) a radar plot to show genre similarity of each music sequence, and (c) a node-link graph to present statistic data of music sequence semantic.

present statistics of music sequences (**G3 & G4**). This graph displays the basic information of music sequences and the similarity information between neighbor music sequences based on the feature vectors.

In the sequence node-link graph, as shown in Fig. 6 (c), each music sequence is represented by a "node", which is a combination of one circle and three donuts. In the "node", the fill color of the circle represents the genre of the music sequence. As shown in Fig. 6 (c), the combined donuts are donut 1, donut 2, and donut 3 from inner to outer. The donut 1 displays the element number of the music sequence. The donut 2 reveals the proportions of notes and chords in the music sequence. In the donut 2, the light color part displays the ratio of notes in the music sequence, while another deep color part corresponds the ratio of chords. The donut 3 presents the ratios of instruments in the music sequence. In the donut 3, each part indicates statistic information of one instrument with four visual encodings including fill color, proportion, inner radius, and outer radius. The fill color is encoded as the instrument name. The proportion shows the ratio of the instrument in the music sequence. The inner radius displays the lowest pitch value performed by the instrument in the music sequence, while the outer radius displays the highest pitch value.

As shown in Fig. 6 (c), the similarity between two music sequences is encoded as the "link", which is a gray line in this graph. The line width of the "link" indicates the similarity value, which is the reciprocal of Euclidean distance based on the feature vectors. Thicker line width indicates that the similarity between two music sequences is larger.

5.2.6 Parallel coordinate plot

To help users access the semantic distribution patterns of music sequences, we design a parallel coordinate plot to display the semantic details simultaneously in multiple dimensions(**G2 & G3**).

As shown in Fig. 5 (f), each line is encoded as a note in the parallel coordinate plot. It has seven dimensions including *musicID, genre, instrument, note class, pitch, start time*, and *type*. The dimensions are divided into three levels the sequence level, instrument level, and note level. The sequence level contains *musicID* and *genre* dimensions. The *MusicID* indicates the music sequence ID of the note, while the *genre* indicates the music sequence genre. The Instrument level has one dimension named *instrument*, which presents the instrument name of the note. The color of the line reveals the instrument name. The last note level includes *note class, pitch, start time*, and *type* dimensions. The *note class* dimension indicates that the note is a note or a chord's note. The *pitch* dimension presents the pitch value of each note. The *start time* dimension displays the timestamp of the note. The last *type* dimension introduces note's type information such as *half, quarter, eighth, 16th*, and *whole*.

5.3 Interaction

Following the visualization principle of *"overview first, zoom and filter, then details-on-demand"* [43], we integrate many interaction methods into MUSE (**G5**) to help users dig out semantic patterns and explore hidden patterns.

As shown in Fig. 5 (a), we offer a interaction menus in the upper left of MUSE to integrate interactions. In the genre instrument tree, users can click each circle to enlarge the points with interests, which helps them explore and mine the distribution patterns. The distribution view allows users to zoom overview, select interested points, overlay a semantic heat map, overlay a density contour, and find a semantic change path, etc. The genre radar plot and node-link graph allow users to zoom the radar and highlight interesting music sequences. In the parallel coordinate plot, users are supported to brush in different dimensions to dig out semantic distribution. In the semantic details view, MUSE provide users with many interactions such as identifying, expanding, aggregating, and removing semantic details.

6 Case Study

We collect a dataset manually to validate the usability of MUSE. The dataset includes 5837 music sequences containing multi-level semantic information such as ID, genre, instrument, note, chord, pitch, time, type, etc. Each music sequence is formed by semantic elements (i.e., note and chord). Based on this dataset, we study two different target cases including the analysis of *semantic distribution pattern* and *semantic change tendency* of music sequences.

6.1 Case 1: Semantic Distribution Pattern

To illustrate the usability of MUSE in analyzing semantic sequence data and exploring the semantic distribution patterns, we introduce a case study as follows.

Identifying semantic distribution pattern. As shown in Fig. 7 (a), all music sequences are projected into a scatter plot with the semantic features of the instrument and melody (**R1**). Each music sequence is encoded as a point. The green points represent the folk genre music sequences. Similarly, other color encodings of the points are yellow-classical, red-rock, and purple-jazz. Genre distribution patterns are recognized easily after overlying a genre's semantic heat map on the scatter plot (**R3**). As shown in Fig. 7 (a), folk music sequences gather in a big pattern; classical music sequences distribute into four patterns; rock music sequences distribute into three intuitive small patterns, while jazz music sequences lay dispersedly in the right of Fig. 7

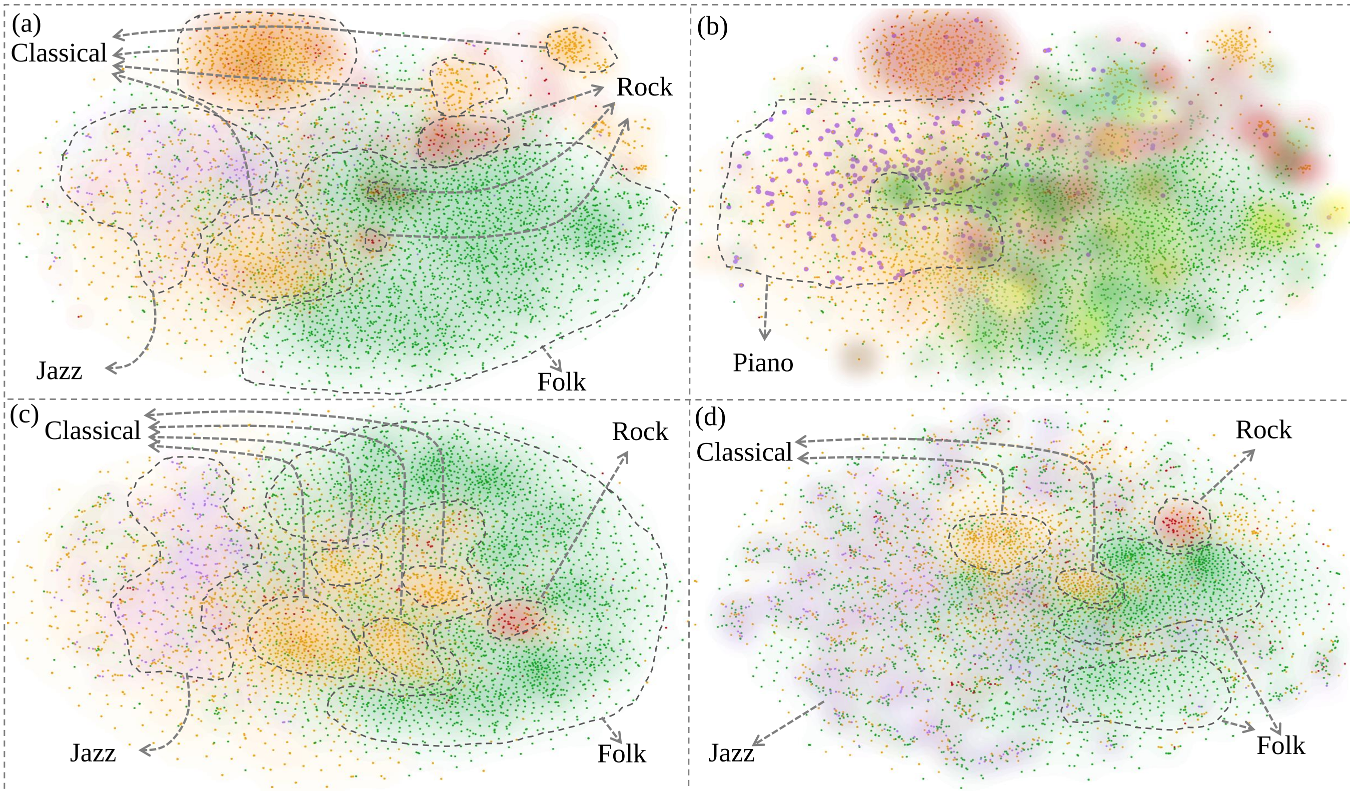


Fig. 7. Different semantic features includes the semantic feature of melody and instrument (a & b), pure melodic semantic feature (c), and the semantic feature of melodic difference (d) are used to project music sequences into 2D space (d). A genre's semantic heat map is overlaid on the distribution in (a, b, and d), and a semantic heat map based on the most percentage instrument is overlaid on the distribution result (b).

(a). Why do jazz music sequences distribute dispersedly instead of gathering into one or many patterns like folk or rock? This question arises when we analyze the distribution pattern of jazz music sequences based on the projection result with the semantic features of the instrument and melody.

Exploring semantic distribution patterns. We propose an assumption that jazz music sequences have different instrumental semantic features. Different instrumental semantic features lead the jazz music sequences to equip different feature vectors to each other. Then the jazz music sequences are projected dispersedly in the scatter plot based on their feature vectors. As shown in Fig. 7 (b), a heat map based on the most percentage instrument is overlaid on the scatter plot. The orange part of the heat map indicates that the largest percentage instrument of most jazz music sequences is the piano (**R3**). Then, we propose a further consideration that the jazz music sequences have different melodic features although they have similar semantic features of instruments. To validate this consideration, we project music sequences into the scatter plot based on pure melodic semantic features vectors (**R4**). As shown in Fig. 7 (c), the folk, rock, and classical music sequences gather into one or many patterns, while the jazz music sequences still distribute dispersedly, which initially verifies our consideration of the melody of jazz music sequences. However, we consider that if one jazz music sequence has synchronously higher pitch value than another jazz music sequence, they will have different feature vectors based on the melodic semantic feature. To avoid this melody-level error, we calculate music sequence feature vectors based on the melodic difference. Then we project music sequences into 2D space as Fig. 7 (d). As shown in Fig. 7 (d), folk, rock, and classical music sequences still have gathered patterns, while jazz music sequences distribute more dispersedly than the other genre music sequences.

Explaining semantic distribution patterns. After the above exploration, we consider that jazz music sequences have a variety of melodies semantic information. Then, we research why the melody of jazz music sequences is various, resulting in the dispersive distribution in the scatter plot. There is a statement that says "*As jazz spread around the world, it drew on national, regional, and local musical cultures, which gave rise to different styles, and jazz is without regular meter, beat, and formal structures*" [52].

According to this case, we advise music novices to pay attention to access jazz music because it has too many melodies to master. If you are interested in jazz music, be well prepared and start with a particular style of jazz music. For music experts, summarizing the features of jazz music and dividing them into different specific clusters are great works and helpful for the development of jazz music.

6.2 Case 2: Semantic Change Tendency

To analyze underlying semantic change tendency and validate the usability of MUSE in identifying and concluding semantic change tendency, we introduce a case study in analyzing semantic change tendency between the music sequences as follows.

Identifying the underlying semantic change tendency. As shown in Fig. 8 (left), our proposed contour is overlaid on the distribution view. In Fig. 8 (left), each point represents a music sequence and the color encodings of points are introduced in Section 6.1. Fig. 8 (right) is the enlarged region of Fig. 8 (left). In Fig. 8 (right), the contour presents a connection area between *area-a* and *area-b*. Most music sequences in *area-a* are classical music sequences (yellow points), while majority of music sequences in *area-b* are folk music sequences (green points) (**R1**). Based on the above identification, we consider a semantic change trend exists in the connection region between *area-a* and *area-b*.

Exploring the semantic change tendency. To validate the consideration, we apply our proposed pathfinding algorithm to generate a path from *area-a* to *area-b* (**R4**). As shown in Fig. 8 (right), the path contains eight music sequences including #144, #2546, #299, #335, #1745, #2865, #3868, and #2651. In these music sequences, #144 is the source music sequence in *area-a*, while #2651 is the target music sequence in *area-b*. In this path, the fill color of points introduces that the first four music sequences are classical music sequences, while the last music sequences are folk.

As shown in Fig. 9 (a), the node-link graph displays the statistics of music sequences in the semantic change path from #144 to #2651. The first four music sequences (#144, #2546, #299, and #335) only have piano instrument semantic features. The last four music sequences (#1745, #2865, #3868, and #2651) have two or more instrumental semantic features such as piano and saxophone (**R3**). According to the links in Fig. 9 (a), #1745 has the lowest similarities to its neighbors compared with other music sequences. In addition, according to the radar plot (Fig. 9 (b)), #1745 has the lowest genre similarities and it is dispersive with other music sequences in the path. Is #1745 the breaking point between *area-a* and *area-b*? To answer this question, we conduct further study to explore the semantic change tendency from #144 to #229 and the one from #2865 to #2651 (**R4**).

The semantic details of these music sequences are exhibited in Fig. 9 (c) (**R2**). The first four music sequences only have piano semantics, while the last music sequences have many instrumental semantic features (i.e., piano, guitar, saxophone, and trumpet). Specifically, from #2865 to #2651, instruments like saxophones (e.g., soprano and alto saxophone) start joining the music sequence. Then, instruments like trumpet and horn join the music sequences. Above conclusions are easily obtained in Fig. 5 (g) and Fig. 9 (c). However, what is the semantic change tendency from #144 to #229 with the same instrumental semantic features? To answer this question, we brush the piano semantic in parallel coordinates to access the instrument distribution from music

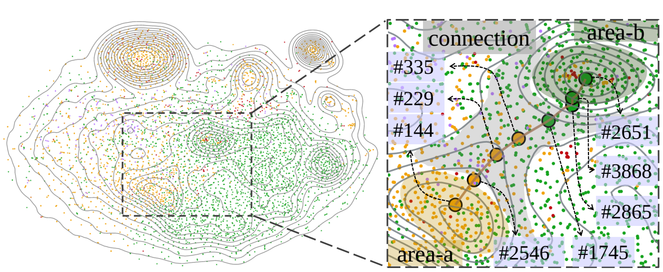


Fig. 8. A contour is overlaid on the distribution view considering sequence distance and semantic similarity (left). An enlarged region of the left view presents a semantic change path from area-a to area-b (right).

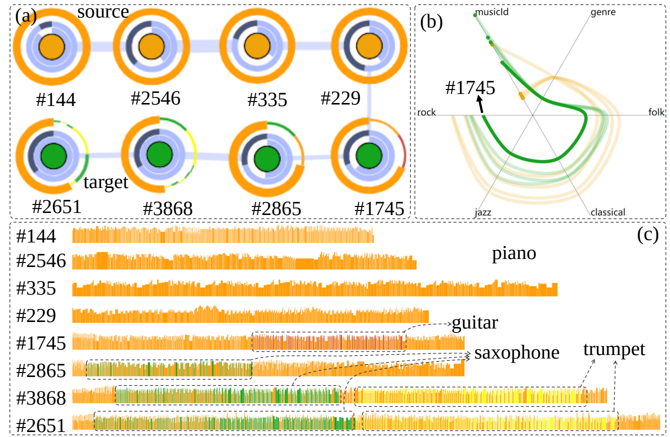


Fig. 9. (a) A node-link graph presents the statistics of music sequences in the semantic change path from #144 (source) to #2651 (target). (b) A radar plot shows that the #1745 has the lowest genre similarity in the path. (c) A semantic details presents semantic details from #144 (source) to #2651 (target).

sequences #144 to #229 (**R3 & R4**). In #144, most pitches aggregate in 65-85, and most types are the *16th*, and *8th*, demonstrating the melody is high and fast. In #2546, most pitches aggregate in 60-80, and the types start changing to be the *8th*, *quarter*, and few *16th*. In #335, most pitches aggregate in 55-75, and most types are *eighth*, *quarter*, and *half*. In #229, most pitches aggregate in 50-70, and most types are the *eighth* and *quarter*.

Semantic change trend from #144 to #229 is that the rhythm of the music sequences changes to be slow, while the pitch changes to be low. The semantic change from #1745 to #2651 in this path is that the instrument feature starts to be abundant. Significantly, instrument semantic features increase with an intuitive trend from #2865 to #2651. The neighbor similarities (links in Fig. 9 (a)) and the genre similarity (highlighted in Fig. 9 (b)) of #1745 are relatively the lowest because its instrumental semantic feature is different from other music sequences. As a result, #1745 plays a role as a breaking point between *area-a* and *area-b*.

According to this case, we advise music novices to learn music along the path from #299 to #144 to master increasing complexity of the rhythm. However, if you are music lovers who want to master how to perform piano in a band, we suggest you learn #2546, #229, and #2561, which include many instruments in the music.

7 Evaluation

7.1 User Evaluation

7.1.1 Participants, Tasks and Procedures

To ensure an in-depth evaluation of this work, we conduct a laboratory user evaluation. The user evaluation also aims to find any issues and suggestions to improve the system.

Participants. We invited six participants (P1-P6) including four males (P1, P4, P5, and P6) and two females (P2 and P3) to perform the user evaluation. They all have different background knowledge, and they are doing further study in different majors as graduate students. P1-P4 have experience in visual analysis, while the major of P5 and P6 is image processing. P1 is interested in automatic visualization and machine learning. P2 and P6 have music background

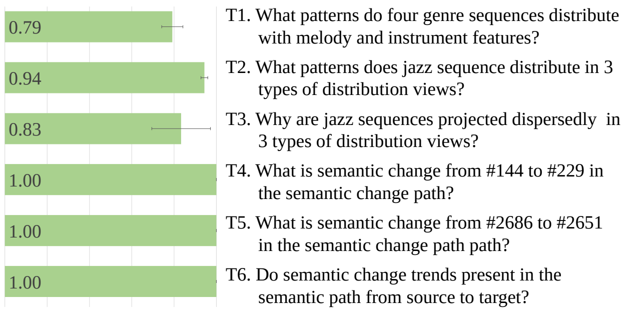


Fig. 10. Tasks in the evaluation and passing rates.

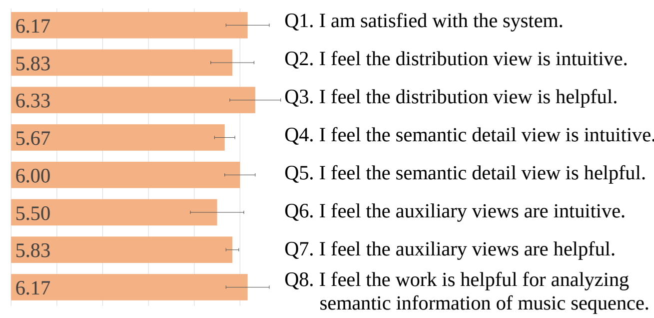


Fig. 11. Questionnaires receive an average rating greater than or equal to 5.5 score in 7 point Likert Scale, which is encouraging.

knowledge, and P6 is interested in parameter optimization. P3's research interest is video digest and visualization. P4 is experienced in the audio process and visualization. The mean age of P1-P5 is 27, and they all are interested in music.

Tasks. We design six tasks (T1-T6) for participants to guide them in performing the targeted analysis including exploring hidden patterns and analyzing semantic change trends. As shown in Fig. 10, T1-T3 are designed to guide participants in exploring genre distribution patterns and explaining why the jazz music sequences distribute dispersedly in the distribution view. T4-T6 are provided to guide participants in analyzing semantic change trends.

Procedures. The user evaluation is divided into four steps. At first, we introduce visual encodings and interaction methods of MUSE to participants. Then, each participant is asked to finish tasks by using MUSE. After finishing tasks, we design a questionnaire to get a objective evaluation from P1-P6. In the end, P1-P6 are allowed to operate MUSE freely and propose any evaluations and suggestions for us. Participants can consult us for visual encodings and interaction methods during the evaluation anytime.

7.1.2 Result

P1-P6 cost average 31 minutes to finish the evaluation. The longest time is 37 minutes cost by P5, and the shortest time is 21 minutes cost by P4. To obtain a comprehensive evaluation, we review the result of tasks and questionnaires.

Tasks' Result. As shown in Fig. 10, participants explore distribution patterns (T1-T3) with equal to or more than 79% accuracy. Regarding T1, three participants (P1, P3, and P5) feel it difficult to summarize distribution patterns of the folk and rock music sequences. P4 has a mistake in analyzing the distribution view based on melody features. P1 is wrong in T3 because he feels it difficult to explain the visual result by data without hints. All the participants are right in the semantic change analysis tasks (T4-T6). We have recorded and analyzed the time taken by each task. In these tasks, T3 takes the longest average time, while T4 takes the second long average time, which indicates that participants perform poorly on answering questions that need multiple interactions without hints. If adding the consultation time, T1 takes the longest average time, since participants may not be familiar with the visual encodings and interaction methods in the beginning.

Questionnaires' Result. After finishing the tasks, we ask participants to rate the system objectively. We integrate the 7 points Likert Scale in questionnaires. 1-7 scores are mapped to *strongly disagree*, *disagree*, *somewhat disagree*, *either agree or disagree*, *somewhat agree*, *agree*, and *strongly agree*. As shown in Fig. 11, Q1 and Q7, which are designed for evaluating the work, receive 6.17 (more than *agree*) indicating that this work (MUSE) is helpful for analyzing semantic information of music sequences. Q6 gets the lowest score (5.5) because the visual clutter affects the participants' analysis. The score of Q3 is the highest (6.33), indicating the distribution view is very useful. Q1, Q3, Q5, and Q8 get greater than or equal to 6 scores (*agree*), demonstrating that the MUSE is helpful in analyzing the semantic information of music sequences.

7.1.3 User Feedback

At the end of the user evaluation, we ask participants to explore the system freely and provide any feedback and suggestions. In summary, P1-P6 offer feedback on three aspects including usability, visual design, and suggestions.

System Usability. MUSE receives positive feedback on the usability in analyzing distribution patterns and semantic change analysis. P6 says, "*I am interested in music analysis, and I think this work is useful for analyzing some features of music.*" He expresses that the contour component is very useful in exploring distribution patterns. Participants affirm our proposed workflow in semantic analysis of sequences. P1 and P2 consent that the methods of MUSE can be used in other sequence semantic analysis. P5 says, "*I learn your ideas and I would like to use these ideas in image sequence analysis.*" Compared with the existing sequence visualization system [27], [39], P2 says, "*By integrating effective interactions and visual designs, MUSE offers a more complete analysis flow for analyzing semantic sequences than any other system I know.*" P3 considers that MUSE can do well presenting sequential data instead of ignoring the overview presentation or details exploration in many current works [1], [47].

Visual Design. Participants praise the visual design of this work. Particularly, P3 says that the views in MUSE are intuitive, which helps her finish tasks. P1 says, "*Visual encodings of the scatter plot are easy to understand, and the interactions in this plot are helpful for me in finishing the tasks.*" He also expresses that the semantic change path is interesting, and he would like to use it in his work. Our attempts that eliminate visual clutter during displaying semantic details receive an exciting statement, too. P4 says, "*One of the worst things is the visual clutter while designing the visualization. Your multiple attempts are a good example for me in avoiding the visual clutter.*" Including positive feedback, participants propose many valuable suggestions for this work, which are summarized in the following.

Suggestions. Participants propose many suggestions which are worth adopting to improve this work in the future. The most suggestion is to refine the visual encodings of instrumental semantic information. Three participants (P1, P4, and P5) worry that the colors encoded as instruments are too many to recognize clearly. They suggest employing many possible solutions such as refining the color encodings of instruments, using multiple shapes to represent instruments, and encoding instruments with textures. P3 has two suggestions for MUSE. At first, she suggests we simplify the interactions of MUSE. She feels the interaction methods of the system are too many to remember completely when the MUSE is introduced to her first time. The second suggestion is that MUSE should equip a component to help her find patterns and analyze semantic change trends by offering hints such as a pattern detection model and semantic change overview. P3 says, "*I need more time to master it (MUSE).*" P1 proposes a suggestion about functions of MUSE. He says, "*I want to analyze the distribution pattern based on one or many instrument features. If the system supports this function, it will be a more impressive work.*"

7.2 Expert Evaluation

To validate the usability of MUSE for experts, we invite two experts (E1 and E2), who are experienced in music analysis domain, to comprehensively evaluate the MUSE. E1 is an associate professor who has researched the music analysis, intelligent music, lyrics auto-translation, and music genre transition for more than ten years. E2 has researched the music emotion analysis, music feature extraction, music genre transition and music auto-generation for many years. E1 has published more than thirty papers, and E2 has published five papers in international conferences and publications. In an informal academic report, we introduce MUSE to them and their students for any suggestions.

MUSE receives exciting evaluations from E1 and E2. In the music analysis domain, researchers usually analyze music based on abstract features such as the emotion and artistry. E1 says, "*It is a good idea to comprehensively and systematically analyze the semantic information of music sequences based on the perspective of data instead of abstract features. The result is exciting. Analyzing music based on notes and chords may help us gain more insights into music or validate our previous work.*" E2 says, "*It is a useful work in large-scale music sequence analysis, especially in the feature extraction of music sequences.*" A variety of effective interactions of MUSE receives their positive evaluation, too. E1 says, "*Human in-loop interactive analysis methods can help users carry out analysis purposefully.*" In addition, experts offer an encouraging evaluation for the pathfinding algorithm. E1 says, "*Exploring semantic change trends between different music sequences is difficult, while the path relaxes the problem.*"

E1 and E2 give us valuable advice to improve this work. E1 suggests that we employ a more extensive dataset in music semantic analysis. He says, "*A more extensive dataset will make this work more perfect.*" In addition, he suggest that we integrate emotion analysis in this work. Similarly, E2 says, "Using more semantic information may improve this work to be more comprehensive." He suggests that we apply the Jsymbolic [35], [36] to extract more semantic information of music sequences.

8 Discussion

The case studies and evaluation demonstrate the usability of MUSE in semantic information analysis of music sequence. In this work, we propose a visualization workflow that integrates ML models and heuristic methods to analyze temporal semantic sequences. An interactive multi-view visual analysis system, named MUSE, is designed and implemented to help users identify and explore the musical semantic sequences. We offer insights into music by using MUSE based on music sequences, which is valuable for music novices and experts. Our methods present possible implications in other domains (gene analysis and traffic analysis) that can be summarized as sequence data model. Nevertheless, the current work has several aspects to be discussed and improved.

Generalization. The methods to analyze the semantic information of music sequences can be generalized to other semantic sequences. Our proposed workflow is valuable in many fields. In addition, the work can be generalized to underlying sequential datasets such as the event sequences, temporal sequences, gene sequences, protein sequences, and so on. Extracting feature vectors and projecting them into low dimensional space is helpful for users to dig out the overall patterns and analyze the similarity of semantic sequences. If the semantic information of sequences has many attributes such as temporal features and category labels, the semantic details can be displayed easily like the semantic details view. However, generalizing this work to other types of sequence semantic analysis still needs comprehensive improvement in the future.

Scalability. Employing a bigger dataset in MUSE will make this work more convincing, which is suggested by experts. MUSE is a visualization system based on the web client. The bigger dataset leads that more visual elements needing to be drawn in the system. However, the pixels of the screen are limited. In our current work, we have examined many attempts to eliminate the visual clutter before presenting semantic details. If the data is numerous, how to present information properly will be a problem faced by MUSE. Another constraint is our recognition. We use different color classes to represent instruments in different categories in current work. We use different colors to represent the different genres. If the category information is too many to be encoded as perceived colors, how to relax the color encoding is a challenge. As a result, improving the scalability of this work is essential in the future.

Uncertainty. The uncertainty caused by the defect of ML methods cannot be avoided completely. In our proposed workflow, we apply the ML methods to compute the feature vectors of music semantic sequences from their text representation and project the feature vectors into 2D plane space for analyzing the distribution patterns. During this process, the uncertainty of the black box is included into our work. For example, the uncertainty issue may exist in the distribution view where the sequences gather in the mixed area, while these sequences are completely different. Therefore, we will conduct an improvement such as importing intermediate correction processes or designing a suitable hierarchical visualization to reduce the uncertainty of machine learning methods.

Music theory. The insights of the music offered by us just focus on high genre, instrument, and overall features levels. However, integrating the music theory may fascinate the users who are interested in the music. How to detect and represent music theory is still a problem, which is a coming study in the future. Music theory is often represented in notes and chords level, resulting it difficult to extract the music theory from digital data. Integrating music theory into our work needs a strong background knowledge of musicology, visualization, as well as computer science.

9 Conclusion

This study proposes a complete visualization workflow for interactive analysis of multidimensional sequence data that incorporates ML models and heuristic ideas. We present an interactive visual analysis system called MUSE, which integrates distribution and semantic change analysis functions to help users explore the semantics at the sequence level. By applying MUSE to large-scale musical sequence data, this work offers profound insights into music, which is valuable for both music novices and experts.

Acknowledgment

This work is supported by National Key Research and Development Program of China (No. 2020YFB1707700) and National Natural Science Foundation of China (No. 61972356 & No. 62036009). Furthermore, thanks to the participants for taking part in the requirement analysis and evaluation in the midst of the COVID-19 pandemic. Guodao Sun is the corresponding author.

References

[1] D. Albers, C. Dewey, and M. Gleicher. Sequence surveyor: Leveraging overview for scalable genomic alignment visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2392–2401, 2011.

[2] G. D. Cantareira, L. G. Nonato, and F. V. Paulovich. Moshviz: A detail+ overview approach to visualize music elements. *IEEE Transactions on Multimedia*, 18(11):2238–2246, 2016.

[3] W. Chai and B. Vercoe. Folk music classification using hidden markov models. In *Proceedings of International Conference on Artificial Intelligence*, pages 1–7, 2001.

[4] W.-Y. Chan, H. Qu, and W.-H. Mak. Visualizing the semantic structure in classical music works. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):161–173, 2009.

[5] Z. Cheng, J. Shen, L. Zhu, M. S. Kankanhalli, and L. Nie. Exploiting music play sequence for music recommendation. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 3654–3660, 2017.

[6] D. Coelho, I. Chase, and K. Mueller. Peckvis: A visual analytics tool to analyze dominance hierarchies in small groups. *IEEE Transactions on Visualization and Computer Graphics*, 26(4):1650–1660, 2020.

[7] M. S. Cuthbert and C. Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of International Society for Music Information Retrieval Conference*, pages 637–642, 2010.

[8] H. Dai, R. Umarov, H. Kuwahara, Y. Li, L. Song, and X. Gao. Sequence2vec: a novel embedding approach for modeling transcription factor binding affinity landscape. *Bioinformatics*, 33(22):3575–3583, 2017.

[9] R. De Prisco, N. Lettieri, D. Malandrino, D. Pirozzi, G. Zaccagnino, and R. Zaccagnino. Visualization of music plagiarism: Analysis and evaluation. In *Proceedings of International Conference on Information Visualisation*, pages 177–182, 2016.

[10] B. Dominikus, S. Frederik, S. Michael, and B. Sebastian. The streams of our lives: Visualizing listening histories in context. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1119–1128, 2010.

[11] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 34–41, 2018.

[12] G. H. Dunteman. *Principal components analysis*. Sage, 1989.

[13] M. M. Farbood, H. Kaufman, and K. Jennings. Composing with hyperscore: an intuitive interface for visualizing musical structure. In *Proceedings of International Computer Music Conference*, pages 111–117, 2007.

[14] H. N. T. N. S. Fukayama and M. Goto. Chordscanner: Browsing chord progressions based on musical typicality and intra-composer consistency. In *Proceedings of International Computer Music Conference*, pages 250–255, 2018.

[15] M. G. M. Gasser, A. Arzt, and G. Widmer. Automatic alignment of music performances with structural differences. In *Proceedings of International Society for Music Information Retrieval Conference*, pages 607–612, 2013.

[16] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano. Songle: A web service for active music listening improved by user contributions. In *Proceedings of International Society for Music Information Retrieval Conference*, pages 311–316, 2011.

[17] D. Gotz and H. Stavropoulos. Decisionflow: Visual analytics for high-dimensional temporal event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1783–1792, 2014.

[18] R. Gove and L. Deason. Visualizing automatically detected periodic network activity. In *IEEE Symposium on Visualization for Cyber Security*, pages 1–8, 2018.

[19] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao. Eventthread: Visual summarization and stage analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):56–65, 2017.

[20] C.-C. Ho, F.-C. Wu, B.-Y. Chen, Y.-Y. Chuang, and M. Ouhyoung. Cubical marching squares: Adaptive feature preserving surface extraction from volume data. In *Proceedings of Computer Graphics Forum*, pages 537–545, 2005.

[21] S. Jänicke, J. Focht, and G. Scheuermann. Interactive visual profiling of musicians. *IEEE transactions on visualization and computer graphics*, 22(1):200–209, 2015.

[22] R. Khulusi, J. Kusnick, C. Meinecke, C. Gillmann, J. Focht, and S. Jänicke. A survey on visualizations for musical data. In *Proceedings of Computer Graphics Forum*, pages 82–110, 2020.

[23] P. Knees and M. Schedl. A survey of music similarity and recommendation from music context data. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 10(1):1–21, 2013.

[24] J. B. Kruskal. *Multidimensional scaling*. Sage, 1978.

[25] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of International Conference on Machine Learning*, pages 1188–1196, 2014.

[26] Z. Li, C. Zhang, S. Jia, and J. Zhang. Galex: Exploring the evolution and intersection of disciplines. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1182–1192, 2019.

[27] H. Lima, C. Santos, and B. Meiguins. Visualizing the semantics of music. In *Proceedings of International Conference on Information Visualisation*, pages 352–357, 2019.

[28] H. B. Lima, C. G. D. Santos, and B. S. Meiguins. A survey of music visualization techniques. *ACM Computing Surveys*, 54(7):1–29, 2021.

[29] C. Liu, K. Zhang, H. Xiong, G. Jiang, and Q. Yang. Temporal skeletonization on sequential data. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1336–1345, 2014.

[30] C. Liu, K. Zhang, H. Xiong, G. Jiang, and Q. Yang. Temporal skeletonization on sequential data: patterns, categorization, and visualization. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):211–223, 2015.

[31] O. Lopez-Rincon and O. Starostenko. Music visualization based on spherical projection with adjustable metrics. *IEEE Access*, 7:140344–140354, 2019.

[32] M. Cuthbert and C. Ariza and B. Hogue and J.W. Oberholtzer. Music21 documention. "https://web.mit.edu/music21/doc", February, 2022.

[33] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008.

[34] A. Mathisen and K. Grønbæk. Clear visual separation of temporal event sequences. In *Proceedings of IEEE Visualization in Data Science*, pages 7–14, 2017.

[35] C. McKay, J. Cumming, and I. Fujinaga. Jsymbolic 2.2: Extracting features from symbolic music for use in musicological and mir research. In *Proceedings of International Society for Music Information Retrieval Conference*, pages 348–354, 2018.

[36] C. McKay and I. Fujinaga. jsymbolic: A feature extractor for midi files. In *Proceedings of International Conference on Mathematics and Computing*, 2006.

[37] M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, 1990.

[38] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[39] K. T. Nguyen and T. Ropinski. Large-scale multiple sequence alignment visualization through gradient vector flow analysis. In *Proceedings of IEEE Symposium on Biological Data Visualization*, pages 9–16, 2013.

[40] R. M. Pereira and C. N. Silla. Using simplified chords sequences to classify songs genres. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1446–1451, 2017.

[41] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: Visualizing personal histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 221–227, 1996.

[42] A. Sheh and D. P. Ellis. Chord segmentation and recognition using em-trained hidden markov models. In *Proceedings of International Conference on Music Information Retrieval*, pages 183–189, 2003.

[43] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[44] G. Sun, B. Chang, L. Zhu, H. Wu, K. Zheng, and R. Liang. Tzvis: visual analysis of bicycle data for traffic zone division. *Journal of Visualization*, 22(6), 2019.

[45] A. Topirceanu, G. Barina, and M. Udrescu. Musenet: Collaboration in the music artists industry. In *Proceedings of European Network Intelligence Conference*, pages 89–94, 2014.

[46] M. Torrens, P. Hertzog, and J. L. Arcos. Visualizing and exploring personal music libraries. In *Proceedings of International Society for Music Information Retrieval Conference*, pages 421–424, 2004.

[47] A. Unger, N. Dräger, M. Sips, and D. J. Lehmann. Understanding a sequence of sequences: Visual exploration of categorical states in lake sediment cores. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):66–76, 2017.

[48] R. Valle. Visual display and retrieval of music information. *arXiv preprint arXiv:1807.10204*, pages 1–13, 2018.

[49] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[50] C. Walshaw. Constructing proximity graphs to explore similarities in large-scale melodic datasets. In *Proceedings of International Workshop Folk Music Analysis*, pages 22–29, 2016.

[51] C. Walshaw. A visual exploration of melodic relationships within traditional music collections. In *Proceedings of International Conference on Information Visualisation*, pages 478–483, 2018.

[52] Wikipedia contributors. Jazz — Wikipedia, the free encyclopedia. "https://en.wikipedia.org/wiki/Jazz", Septemper. 2021.

[53] G. Yachdav, S. Wilzbach, B. Rauscher, R. Sheridan, I. Sillitoe, J. Procter, S. E. Lewis, B. Rost, and T. Goldberg. Msaviewer: interactive javascript visualization of multiple sequence alignments. *Bioinformatics*, 32(22):3501–3503, 2016.

[54] Y. Yu, S. Tang, F. Raposo, and L. Chen. Deep cross-modal correlation learning for audio and lyrics in music retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 15(1):1–16, 2019.

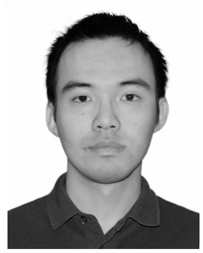
[55] Y. Zhang, K. Chanana, and C. Dunne. Idmvis: Temporal event sequence visualization for type 1 diabetes treatment decision support. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):512–522, 2018.

[56] Y. Zhang, S. Di Bartolomeo, F. Sheng, H. Jimison, and C. Dunne. Evaluating alignment approaches in superimposed time-series and temporal event-sequence visualizations. In *Proceedings of IEEE Visualization Conference*, pages 1–5, 2019.

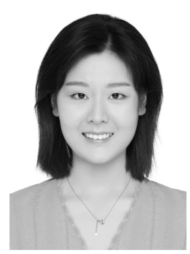
[57] S. Zhu, G. Sun, Q. Jiang, M. Zha, and R. Liang. A survey on automatic infographics and visualization recommendations. *Visual Informatics*, 4(3), 2020.

[58] H. Ziegler, M. Jenny, T. Gruse, and D. A. Keim. Visual market sector analysis for financial time series data. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 83–90, 2010.

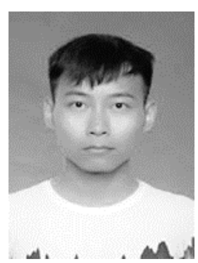
Baofeng Chang is working toward Ph.D. degree in computer science and technology in College of Computer Science and Technology, Zhejiang University of Technology. His main research interests are urban visualization, temporal sequence visualization, and dynamic graph visualization.



Guodao Sun is an associate professor at the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. He received the B.Sc. in computer science and technology, and Ph.D. degree in control science and engineering both from Zhejiang University of Technology. His main research interests are urban visualization, visual analytics of social media, and information visualization.



Tong Li is a master's student in the College of Computer Science and Technology, Zhejiang University of Technology. Her current research interests include information visualization, visual analytics, and visual data mining.



Houchao Huang is a master's student in the College of Information Engineering, Zhejiang University of Technology. His current research interests include information visualization, visual analytics, and visual data mining.

Ronghua Liang received the Ph.D. in computer science from Zhejiang University. He worked as a research fellow at the University of Bedfordshire, UK, from April 2004 to July 2005 and as a visiting scholar at the University of California, Davis, US, from March 2010 to March 2011. He is currently a Professor of Computer Science and Dean of College of Computer Science and Technology, Zhejiang University of Technology, China. His research interests include Visual Analytics and Computer Vision.