

## **Interactive Interpretation of Serial Episodes: Experiments in Musical Analysis**

Béatrice Fuchs, Amélie Cordier

► **To cite this version:**

Béatrice Fuchs, Amélie Cordier. Interactive Interpretation of Serial Episodes: Experiments in Musical Analysis. 21st International Conference on Knowledge Engineering and Knowledge Management (EKAW-2018), Amedeo Napoli; Yannick Toussaint, Nov 2018, Nancy, France. pp.131-146, 10.1007/978-3-030-03667-6\_9 . hal-01933264

**HAL Id: hal-01933264**

**<https://hal.archives-ouvertes.fr/hal-01933264>**

Submitted on 17 Mar 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Interactive Interpretation of Serial Episodes: experiments in Musical Analysis.

Béatrice Fuchs<sup>1</sup>, Amélie Cordier<sup>2</sup>.

<sup>1</sup> Université de Lyon, UJML3, IAE, LIRIS, F-69 008, Lyon, France,  
beatrice.fuchs@liris.cnrs.fr

<sup>2</sup> Université de Lyon, LIRIS, F-69 100, Villeurbanne, France,  
amelie.cordier@liris.cnrs.fr

**Abstract.** We propose an interactive approach for post-processing serial episodes mined from sequential data, *i. e.* time-stamped sequences of events. The strength of the approach rests upon an interactive interpretation that relies on a web interface featuring various tools for observing, sorting and filtering the mined episodes. Features of the approach include interestingness measures, interactive visualization of episode occurrences in the mined event sequence, and an automatic filtering mechanism that remove episodes depending on the analyst's previous actions. We report experiments that show the advantages and limits of this approach in the domain of melodic analysis.

## 1 Introduction

The aim of knowledge discovery (KD) is to identify pieces of knowledge in large volumes of data through non trivial methods, during an *interactive* and *iterative* process [1]. This process is said to be iterative because several iterations are often needed to understand complex phenomena captured in data, each one contributing to gradually improve the global understanding of the knowledge hidden in the data. The process is said to be interactive because it is fully guided by the user, an analyst expert in the domain. At each stage, the analyst organizes the tasks to be performed, identifies the relevant knowledge and decides on the next actions to be taken. The role of the analyst is even more important when the domain knowledge is hard to capture and/or when it is not available in the system. In KD, research has long focused on the mining step because it raises a few challenging issues for computer scientists. However, to transform the mining results in actionable knowledge, human expertise is required, and this is a tedious task for the analyst. First, finding good settings for the mining is far from obvious and requires most of the time several trials. Next, the interpretation step is very long and tedious because there may be thousands of results that have to be processed manually. However, all the steps of the KD process should be taken into account when seeking to improve interactivity [2]. The involvement of a human analyst in the KD process is a key challenge, as discussed in [3]. Therefore, the user is given a central role in the KD process and there is a real need to integrate several features such as visualization techniques, human-computer interactions and knowledge engineering.

In this paper, we focus on the assistance to interpretation during the post-processing stage. Our approach aims to increase the involvement of the analyst by giving him a

leading role in order to enhance his cognitive mobilization and by assisting him through different interaction capabilities. The approach is reified in a tool, the TRANSMUTE prototype, a web-based interface supporting interactive interpretation of serial episodes in event sequences that assist the work of the analyst. The approach supports the tracking of the analyst work in an interactive and iterative way. On each iteration, the analyst may visualize, sort and interact with the results of the mining. His actions are taken into account to manage his work and to enable him to focus more quickly on other relevant episodes. The main contributions of the approach rely on the following features :

- a visual and customisable interface to enable user interactions, to navigate into episode occurrences and display them in the sequence,
- the possibility of combining different interestingness measures to rank serial episodes in association to an original and automatic filtering process that discards episode occurrences and episodes depending on previously selected episodes.
- the possibility for the user to express an interpretation of the selected episodes by creating and adding them an annotation in the form of a new type of event,
- and finally to record a new sequence resulting from the interpretation into a data storage.

Thereafter, section 2 presents several related approaches in the literature. Then the main features of TRANSMUTE are presented in section 3 and the underlying principles of the KD process and principles of our approach for interpretation are presented formally in section 4. Section 5 reports on some experiments in the musical domain to verify the efficiency. The paper ends with a discussion on the limits of this approach followed by a conclusion.

## 2 Related works

Related works stem from several complementary research areas: data mining, interactive visual analysis and knowledge engineering tend to combine together several complementary features.

In the literature, several methods have been proposed to cope with the surabundance of mining results, such as interestingness measures, concise representations, data compression and post-processing filtering. The exclusive use of objective interestingness measures has proved to be unable to solve the recurring problem of overabundance and redundancy of mining results. An alternative principle has been successfully applied, *minimum description length* (MDL), which is concerned by selecting the best set of patterns based on their ability to compress data [4], [5]. Although this approach is well suited for itemsets, its application for serial episodes in event sequences is more delicate because the events composing pattern occurrences are not necessarily contiguous in the sequence and there may be repetitions of the same event types. This makes the coding more complex and decreases the efficiency and finally, the advantage of compression [6]. Moreover, the MDL approach is not always practicable because data may not be suitable for compression. Nevertheless these principles can be useful to develop other methods, without keeping the objective of data compression.

Taking into account the user as a leading actor in the KD process has gradually imposed itself. For a long time the research community recognized that the active involvement of a human expert in the KD process is essential to benefit from his knowledge and expertise [2], [7]. In this sense, visualization has been widely explored in order to increase human cognition to better understand the data, through graphical representations into which it is possible to navigate in order to change the point of view on data, according to the mantra "*preview first, zoom / filter, details on request*" [8].

Although apparently close, interactive data exploration using data mining techniques is quite different in its objectives [3] although it benefits from features of interactive data visualization. If it relies on an interface to visualize data and results obtained by mining, it consists more in an assistance to the user looking for significant patterns in the data in order to construct a model explaining the data. Several research works in interactive data exploration have been conducted and workshops dedicated to this field appeared (IDEA, IDA). For example, [9] present an interactive approach to episode mining and analysis that supports visual exploration of episodes mined from health data. [10] specifies precisely outlines of information visualization and visual analysis, recalls that data mining is only one step in the process of KD and shows the different steps of this process related to the visualization facilities. The paper discusses the typical components found in these systems, including a list of interaction techniques reflecting the user's intentions: *select, explore, reconfigure, encode, abstract / elaborate, filter and convert*.

More recent approaches claim that interestingness is subjective in essence and take into account the goals of the user and its knowledge [3] in an interactive way to involve the user in the exploration process. Several works aim to take into account user's actions to learn preferences as for example [11], [12], [13]. In this trend, most recent works aim to capture subjective interestingness by learning user's preferences or intentions when interacting with mining results.

To our opinion, these approaches give the user a rather passive role in the KD process. Furthermore, we argue that subjectivity can be expressed in another way: *a priori* knowledge on data or knowledge that, once encoded in the form of interestingness measures or properties can be helpful for choosing mining parameters, choosing a pre-processing strategy or helping ranking and interpreting the results. Our approach aims to integrate several approaches to take full advantage of their combination in an interactive way, where the user is considered a leading actor of the KD process. It does not focus on visualization techniques but aims at providing a simple interface to support interactions and understanding data, where the user has the initiative and leads the KD process with several complementary tools. Interactions aim at assisting the interpretation process and model construction. The main difference of our approach is that interactivity is not focused on the same aspects as most related works: the user drives really the KD process, have an active role, and can bring his expertise to the interpretation. It takes advantage of event-based or temporal-based interestingness measures close to the compactness measure [14] where compactness represents the number of gap events between the events of the episode. Furthermore, the filtering process is closely related to these measures. The results of interpretation are recorded in a persistent data store which can be compared to [5], but with a different objective and in an interactive way. The objective is not

to compress data but rather to characterize the potential interestingness of episodes by their relative importance in the sequence explored, and to remove useless episodes after selection, enabling to quicker focus on other candidate episodes. Next, the recording of the chosen episodes associated with their interpretation in a transformed sequence fall rather under a knowledge engineering objective than compression.

We present in the next section the TRANSMUTE system before studying formally its principles.

### 3 TRANSMUTE

The main motivation of this work is to explore *traces*, *i. e.* sequences of actions performed by users during their activity supported by a computer system. Meanwhile, the approach and the tools developed aims at being applied to any sequential data such as a musical score, which serves as an application domain for experiments in this paper.

TRANSMUTE is an interactive tool that implements the KD process from sequential data. The analyst can interact with the elements of the trace, choose mining parameters, launch the miner, and select episodes outputted by the miner to see its occurrences. The interface of TRANSMUTE is shown on Figure 1, where traces of the serious game TAMAGOCOURS are analyzed. TAMAGOCOURS is a collaborative serious game for learning the rules of diffusion of numerical resources. The main interface of TRANSMUTE has several parts:

On the upper part are the traces: the trace being analysed ① and the trace in which interpreted episodes are stored ②. On top of the trace in ①, new event types are intended to replace occurrences of selected episodes in the trace ②. To facilitate the interpretation of the events in the trace, the trace model is displayed ③: each event type is displayed next to the displayed icon. The lower part of the interface is divided in two sections. On the lower-left ④, the mining parameters chosen by the analyst are displayed. On the lower-right ⑤, the list of episodes found during the mining are displayed, and for each of them, interestingness measures with sorting buttons. At the top we find the episodes chosen by the user, below the remaining episodes to be examined and at the bottom the filtered episodes as a consequence of the previous user's choices. The filtered episodes are faded: they are temporarily discarded and not available for selection. It is possible to undo an episode selection and the remaining episodes are updated consequently.

TRANSMUTE implements the visual interpretation process and is built upon several other components (figure 2). The first one is DISKIT which implements the rest of the KD process. DISKIT encapsulates the DMT4SP miner<sup>3</sup> [15], a tool for extracting episodes or sequential rules from one or several sequences of events, in accordance with the minimal occurrence semantic rule [16]. DMT4SP produces a set of frequent serial episodes satisfying the constraints specified in the settings. Next, a visual framework provides basic methods for visualizing and interacting with traces. Finally, a data storage ensures the storage of traces or sequences, and provides basic data management capabilities such as storing sequences associated with an explicit descriptive model,

<sup>3</sup> <http://liris.cnrs.fr/~crigotti/dmt4sp.html>

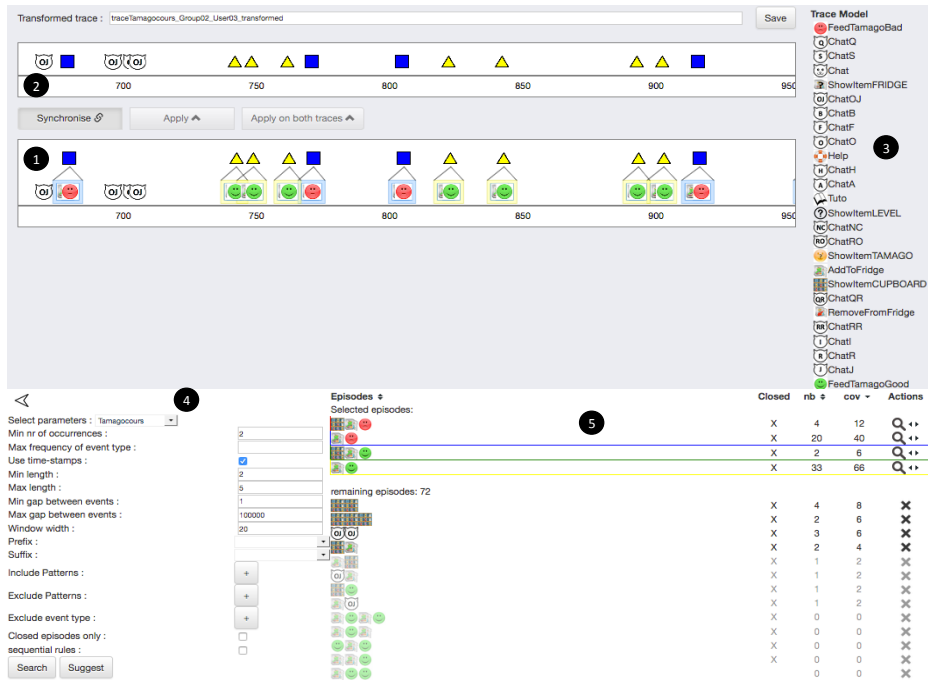
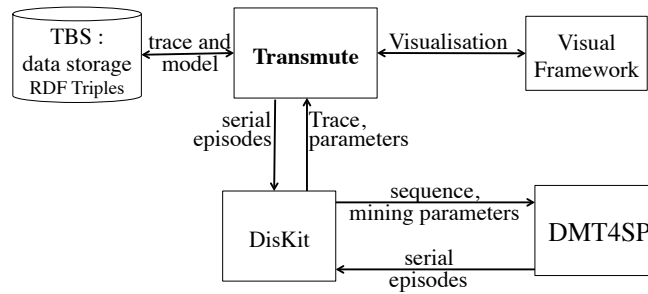


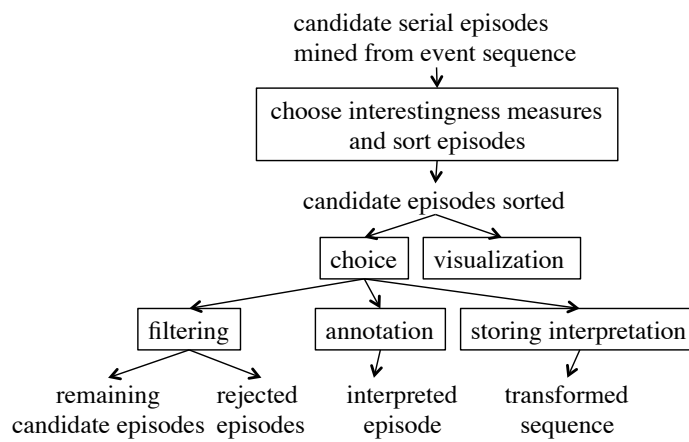
Fig. 1. The interface of TRANSMUTE.

or sequence transformation in order to manipulate them (abstraction, filtering, *etc.*). Sequences and their associated model are stored in a triplestore as RDF triples. The transformations will not be detailed more in this paper.

The main important operations involved in the interpretation process are shown in Figure 3. The process is repetitive, as many times as the user wants to sort, select or annotate episodes and if there are remaining episode to examine. The interpretation process begins with the serial episodes outputted by the miner. These are called candidate episodes. The user chooses interestingness measures to sort the candidate episodes. Then he can select an episode to visualize its occurrences in the displayed trace and navigate through the different occurrences. When an episode is selected, a filtering process is triggered. It consists in temporarily filtering all the occurrences of the unselected candidate episodes having at least one event in common with the occurrences of the selected episode, and then filtering the episodes whose support, recalculated by taking into account the filtered occurrences, is under the threshold initially set by the user. The episodes thus eliminated remain visible to the user but they are faded in the interface and can not be selected any more. The user can interpret a selected episode by adding an annotation which is displayed with an associated icon in a new trace. The user can record the result of his work in a *transformed* sequence that stores all the annotations resulting from the interpretation. The transformed sequence is built from the original sequence by replacing every occurrence of the selected episode by a new type of event



**Fig. 2.** The architecture of TRANSMUTE.



**Fig. 3.** Overview of the main steps of the interpretation process.

resulting from the annotation. The next section describes formally the main features of TRANSMUTE.

## 4 Interactive interpretation of serial episodes

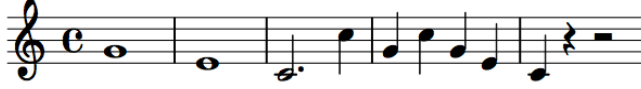
Sequences are studied in the context of a classical KD process made up of the main steps: pre-processing (selection of the trace, transformation), mining, and post-processing (visualization and interpretation of candidate episodes). Hereafter, we first describe briefly the basic definitions of the concepts involved in the process and we further develop the post-processing step. The concepts are illustrated in the domain of musical analysis where a typical task consists in analysing a musical score described as a sequence of notes associated with a duration and a time-stamp in order to find recurrent melodic patterns.

## 4.1 Definitions

During the pre-processing, a sequence is chosen by the analyst and transformed in the format of the analyser.

**Definition 1 (Sequence).** A sequence  $S$  is a set of event occurrences – or events for short –, each one having a type and a date. An event is a couple  $(e_i, t_i)$  where  $e_i \in E$  is a type of event and  $E$  is the set of types of events, and  $t_i \in \mathbb{N}$  is a time stamp associated with  $e_i$ .

The transformation of the input data into an event sequence is a simple bijective syntactic transformation to conform to the miner’s data format <sup>4</sup>. In the musical analysis domain, every music note name and rest of the score corresponds to an event type of the event sequence. The note values (whole notes, quarter note, etc.) are used to compute a time stamp associated with each event of the sequence. We can notice that depending on the application domain, the events of a sequence may not always be characterized by a duration. This is the case for numerical traces for example. Let us consider the following extract of a musical score :



This musical piece can be described by the following input data <sup>5</sup>:

Note name G E C C G C G E C

Duration 4 4 3 1 1 1 1 1 1

time-stamp 0 4 8 11 12 13 14 15 16

From this trace, the following sequence is built with  $E = \{C, E, G\}$ :

$S = \{(G, 0), (E, 4), (C, 8), (C, 11), (G, 12), (C, 13), (G, 14), (E, 15), (C, 16)\}$

In the following step, the analyst provides the parameters to guide the mining. In our implementation, we use DMT4SP.

**Definition 2 (Serial episode, occurrence).** A serial episode  $p = (e_1, e_2, \dots, e_n)$ ,  $e_i \in E$  is a sequence of types of events of length  $l_p = n$ . An occurrence  $o_p$  of the episode  $p$  is a set of events  $o_p = \{(e'_i, t'_i)\}_{i=1,n}$  such as  $(e'_i, t'_i)_{i=1,n} \in S$  and  $\forall i, e_i = e'_i$ .  $O_p = \{o_p^j\}_j$  is the set of occurrences of the episode  $p$  in  $S$ .

**Definition 3 (Frequency, support).** We call frequency or support of an episode  $p$  the number of occurrences of this episode in  $S$ . We denote it  $\sigma(p) = |O_p|$ . The mining returns a set  $P$  of frequent episodes such as  $P = \{p_i\}, \forall i, \sigma(p_i) \geq \sigma_{min}$ , where  $\sigma_{min}$  is the minimal support chosen by the analyst.

In the example, if  $\sigma_{min} = 2$ , the occurrences of the episode  $(G, E, C)$  are:

$op_1 = \{(G, 0), (E, 4), (C, 8)\}$  and  $op_2 = \{(G, 14), (E, 15), (C, 16)\}$ .

<sup>4</sup> Event types are represented with integer values

<sup>5</sup> Here, the notes have 4, 3 and 1 beats, and the corresponding note values are respectively whole note, dotted half note and quarter note.



$$\sigma((G, E, C)) = 2 \geq \sigma_{min}.$$

Additional constraints may be specified but outside the scope of this article and will not be detailed: time constraints (gap min, window size, and in a limited way a constraint similar to gap max), syntactic constraints (min/max size, prefix, last element), episode closure, sub-episode inclusion/exclusion, event type filtering.

## 4.2 Post-processing and interactive interpretation

The serial episodes outputted by the miner are displayed in the Transmute application with appropriate chosen icons (Figure 1).

### Interestingness measures

To help the user sort and choose episodes, some interestingness measures are computed. First, the closure property of episodes is important because it leads to a more compact representation and limits the number of episodes generated. An episode is closed if it is not included in a longer episode having the same support<sup>6</sup>. The user can choose this option to somewhat limit the results. In Figure 1, the *closed* column contains marks in front of closed episodes.

In addition to the closed episodes, we propose some interestingness measures below in order to measure the combinatorial redundancy and the *compactness* of episodes. The first measure is *event coverage* and takes into account the number of event of the sequence. The second measure is the *temporal coverage* and takes into account the event durations.

**Definition 4 (Event coverage).** *The event coverage  $EC_p$  is the set of distinct events of the occurrences of an episode  $p$ .  $EC_p = \bigcup_{i=1}^{\sigma(p)} o_p^i$ . The event coverage indicator  $ECl_p$  of an episode is the number of distinct events :  $ECl_p = |EC_p|$ .*

**Definition 5 (Event spreading).** *The time interval  $Int(s)$  of a sequence  $s$  is the time interval between the first and the last event of the sequence.  $Int(s) = [t_{min}, t_{max}]$ ,  $t_{min} = \min(t_i)$ ,  $t_{max} = \max(t_i)$ ,  $\forall (e_i, t_i) \in s$ . The spreading  $ES_p$  of an episode  $p$  of length  $n$  is the set of events of  $S$  having a time-stamp included in the time intervals  $Int(o_p^i)$  of the occurrences of  $p$ .  $ES_p = \{(e_k, t_k) \in S, \forall Int(o_p^i) = [t_{min}^i, t_{max}^i], t_{min}^i \leq t_k \leq t_{max}^i\}$ . The event spreading indicator is the number of events whose time-stamp is included in the time intervals of the occurrences of  $p$ :  $ESI = |ES_p|$*

**Definition 6 (Noise).** *The noise  $N_p$  of an episode is the number of events not belonging to the occurrences of a given episode and inserted in the temporal interval of a episode:  $N_p = ES_p \setminus EC_p$*

*The noise indicator  $NI_p$  is the number of events not belonging to the occurrences of a given episode and inserted in the temporal interval of a episode:*

$$NI_p = ESI_p - ECl_p$$

<sup>6</sup> The computation of the closure property in TRANSMUTE is based on the number of occurrences of a serial episode. It is not detailed in this paper.

As explained before, in some situations when the events are associated with a duration, it is possible to compute temporal indicators of the same nature than the previous event indicators but that focus on the duration of the events. This is the case in the domain of musical analysis where each note has a duration.

**Definition 7 (Temporal coverage).** *Temporal coverage  $TC_p$  of an episode  $p \in P$  is the total duration of the events of the event coverage. We note  $d(o)$  the duration associated with the event  $o$ . Let  $p$  an episode of event coverage  $EC_p, \forall o_i \in EC_m, TC_p = \sum_i d(o_i)$ .*

In the previous example, let consider the episodes  $p_1 = (G, C)$  and  $p_2 = (G, E, C)$ .  
 $EC_{p_1} = \{(G, 0), (C, 8), (G, 12), (C, 13), (G, 14), (C, 16)\}, ECI_{p_1} = 6,$   
 $ES_{p_1} = \{(G, 0), (E, 4), (C, 8), (G, 12), (C, 13), (G, 14), (E, 15), (C, 16)\},$   
 $ESI_{p_1} = 8,$   
 $N = \{(E, 4), (E, 15)\}, NI_{p_1} = 2$  and  $TC_{p_1} = 11.$   
 $EC_{p_2} = \{(G, 0), (E, 4), (C, 8), (G, 14), (E, 15), (C, 16)\}, ECI_{p_2} = 6,$   
 $ES_{p_2} = \{(G, 0), (E, 4), (C, 8), (G, 14), (E, 15), (C, 16)\}, ESI_{p_2} = 6,$   
 $N_{p_2} = \emptyset, NI_{p_2} = 0$  and  $TC_{p_2} = 14.$

The event coverage indicator aims at giving an estimate of the importance of an episode in the sequence. When many episodes differ only with a very few number of events, this indicator helps decide which episode seems to be the best to choose. The event coverage aims at finding all the episode occurrences the events own to and consequently the episodes in competition with a given chosen episode. These definitions enable us to describe the principles of interactive filtering.

### Interactive filtering

Once episodes are sorted and displayed in the interface of TRANSMUTE, the analyst can select them to highlight their occurrences in the trace, and choose the most relevant ones. When an episode is selected by the expert, all the occurrences of the episode are framed in the interface and all the occurrences of the other episodes whose occurrences have at least one event in common with the occurrences of selected episode are eliminated, the support is re-calculated consequently, and the episodes having an insufficient support are temporarily discarded from the available selection. This process constitutes the *filtering* operation. As a consequence of the filtering process, there is a progressive decrease of the number of results that the analyst has to investigate, which facilitates the next choices because there are less episodes to consider. In the previous example, if the episode  $(G, E, C)$  is chosen, it may become obsolete to consider other episodes such as  $(G, E)$  or  $(E, C)$  because some of their occurrences may include events appearing in  $(G, E, C)$ . The filtering operation addresses this observation: all the redundant episodes given the chosen episode are discarded which enables the analyst to focus more quickly on other episodes. As the interpretation step is iterative, during each iteration, the selection of an episode is followed by a filtering step which as a consequence filters the remaining episodes and the list of remaining episodes progressively decreases. Filtering plays an important role when providing assistance to the analyst during the interpretation phase. This step relies on the notion of event coverage for searching episodes to delete.

**Definition 8 (Filtering).** Let  $p_s$  an episode selected by the expert, having an event coverage defined as  $EC_s$ . Let  $p_i \in P, p_i \neq p_s$  an episode, the set of occurrences of  $p_i$  invalidated by the selection of  $p_s$  is:  $O(p_i|p_s) = \{\forall o_i \in O(p_i), o_i \cap EC_s \neq \emptyset\}$ . We denote  $P_{p_s} \subset P$  the set of episodes of  $P$  invalidated by the choice of  $p_s$  by the analyst:  $P_{p_s} = \{p_i \in P, p_i \neq p_s, \text{ such as } \sigma(O(p_i|p_s)) < \sigma_{min}\}$ . When the episode  $p_s$  has been selected, the set of remaining episodes to consider during the next iteration is:  $P \setminus P_{p_s}$ .

In the example, let's consider the following episodes:

$p_1 = (G, E, C), p_2 = (G, E)$  and  $p_3 = (G, C)$ .  
 $CE_{p_1} = \{(G, 0), (E, 4), (C, 8), (G, 14), (E, 15), (C, 16)\}$ ,

The occurrences of  $p_2$  and  $p_3$  are:

$o_{p_2}^1 = \{(G, 0), (E, 4)\}, o_{p_2}^2 = \{(G, 14), (E, 15)\}$ ,  
 $o_{p_3}^1 = \{(G, 0), (C, 8)\}, o_{p_3}^2 = \{(G, 12), (C, 13)\}, o_{p_3}^3 = \{(G, 14), (C, 16)\}$ .

If the expert chooses the episode  $p_1$ ,  $o_{p_2}^1$  and  $o_{p_2}^2$  are deleted, as well as  $o_{p_3}^1$  and  $o_{p_3}^3$ . The supports of  $p_2$  and  $p_3$  respectively become 0 and  $1 < \sigma_{min}$ . The filtering after the selection of  $p_1$  therefore eliminates  $p_2$  and  $p_3$ . In this example, the event types of the episodes  $p_2$  and  $p_3$  are included in the event types of  $p_1$ , but it does not mean that all the events of the occurrences of  $p_2$  and  $p_3$  are included in the events of the occurrences of  $p_1$  (for example  $o_{p_3}^2 = \{(G, 12), (C, 13)\}$ ). In other domains, for example TAMAGO-COURS previously mentioned, there are two important episodes:  $(S, A, F)$  and  $(A, F)$ , and the deletion of  $(S, A, F)$  do not remove the  $(A, F)$  episode because there are a lot of  $(A, F)$  occurrences without the  $S$  event before.

The filtering initially starts with the set  $P$  of all the episodes that have been mined. In each iteration, the set  $P$  is progressively cleaned around a selected episode as if it eliminates the neighbouring episodes in the search space, often redundant episodes.

## 5 Experiments

In order to verify the effectiveness of the interpretation strategy, we propose to apply it on real musical pieces in the field of melodic analysis. We have studied three music pieces. For each of them, a music expert gave before the experiment a list of relevant episodes. In the following, we call these episodes the *expert episodes*. We launched the miner with parameters chosen in order to guarantee that all the expert episodes were recalled in the mining results ( $\sigma_{min} = 2$ ). The results are listed in the table below:

Title of the music piece	number of mined episodes	number of expert episodes
1. Reichert, Tarentelle	3 853	29
2. Ibert, Entr'acte	12 947	20
3. Debussy, Syrinx	59 786	11

Then we propose to assess the efficiency of the process by measuring the effort it takes the analyst to find all the expert episodes according to different strategies. The effort is defined as the total number of episodes the expert has to examine before finding every expert episode. The effort of the expert is estimated with the rank of each expert episode in the candidate episodes, if we assume that episodes are examined in sequence by the expert from the beginning of the mining results. The lower the rank,

the lower the number of episodes to examine before reaching an expert episode and consequently the effort, showing the efficiency of the strategy. The effort is measured in several circumstances where the episodes are sorted using different criteria: natural order of the mining (no sorting), frequency, event coverage and temporal coverage. A second experiment uses the noise as a first sorting criteria, and the four previous criteria as second sorting criteria.

Once episodes are sorted, we note the rank of the expert episodes for each sorting criteria, which, as explained before, indicates the effort required by the expert to find each episode. We report the results in the two tables below. In the first table, the effort is first measured without filtering. In the second table, the effort is measured with filtering applied each time an expert episode is found and considered as selected. Intuitively, it is clear that the deletion of episodes each time an expert episode is selected should lead to a decrease of the number of episodes and should have an immediate effect on the new ranks of remaining expert episodes. Nevertheless, experts episodes may be wrongly eliminated because of the filtering process. We measure this bias through the recall measure.

With no filtering, the effort of the expert is measured with the rank of the last expert episode and is given in Table 1 for the three pieces, according to the used criteria ( $\nearrow$  indicates and ascending sorting, and  $\searrow$  a descending sorting). With no filtering, the

piece	mining episodes	expert episodes	without sorting	Noise $\nearrow$ and					
				$\sigma \searrow$	$ECI \searrow$	$TC \searrow$	$\sigma \searrow$	$ECI \searrow$	$TC \searrow$
1	3 853	29	3 838	3 838	3 797	2 735	240	369	<b>233</b>
2	12 947	20	12 818	12 829	12 591	<b>9 516</b>	12 667	12 672	12 668
3	59 786	11	57 424	57 935	34 886	<b>32 847</b>	52 928	52 906	52 906

**Table 1.** Effort of the expert without filtering.

best results are obtained with a sorting by ascending noise and descending temporal coverage for piece 1, by descending temporal coverage for pieces 2 and 3. The recall is 100% as no episode is deleted in this experiment. Compared to no sorting at all, the reduction of the expert's effort is respectively 94%, 26%, and 43%. For all the criteria, the use of measures has enabled an improvement as it always enables to reduce the effort of the expert. Meanwhile, this result is not completely satisfactory because the effort required from the expert remains too important.

When filtering is introduced, the effort of the expert is computed differently. As the candidate episodes are sorted after each selection of an expert episode, the expert must resume the episode review from the beginning. Therefore, the effort of the expert equals the sum of the ranks of the different expert episodes after each filtering. The results are reported in Table 2. The best results are obtained with a sorting with ascending noise and descending temporal coverage for piece 1, descending temporal coverage for piece 2 and event coverage for piece 3. We can note that, without sorting, review of episodes with revision requires a much higher number of interactions than the initial number of episodes, which is possible because after each choice review starts again from the

beginning. We can note that the variations obtained between the different pieces do not make it possible to definitively conclude on the utility of the interestingness measures. This implies that it is important to propose a range of measures to the user. However, the filtering process associated with a measurement always leads to a significant gain.

music piece	mined episodes	experts episodes	without sorting	$\sigma \searrow$	$ECI \searrow$	$TC \searrow$	Noise ↗ and		
							$\sigma \searrow$	$ECI \searrow$	$TC \searrow$
1	3 853	29	1442	175	138	66	60	68	<b>45</b>
2	12 947	20	3056	638	317	<b>231</b>	619	481	481
3	59 786	11	204 851	55 956	<b>9274</b>	9 772	12 792	11 482	11 531

**Table 2.** Effort of the expert with filtering.

The diminution of the effort obtained with the filtering strategy with regard to a processing without sorting and without filtering (table 3) is significant and shows the efficiency of the filtering strategy combined with the interestingness measures. Nevertheless, the recall for the expert episodes is 82% for piece 1, 100% for piece 3 and 33% for piece 2 when the sorting does not use the noise and 62% when noise is introduced. The introduction of the noise favoured the episodes composed of closer notes and has improved the recall, which remains however perfectible.

Piece		$\sigma \searrow$	$ECI \searrow$	$TC \searrow$	Noise ↗ and		
					$\sigma \searrow$	$ECI \searrow$	$TC \searrow$
1	Diminution	95%	96%	98%	98%	98%	98%
	Recall	82%			82%		
2	Diminution	95%	98%	98%	95%	96%	
	Recall	33%			62%		
3	Diminution	3%	84%	83%	78%	80%	
	Recall	100%			100%		

**Table 3.** Recall rate of expert episodes and diminution of the effort for the strategy with filtering vs without filtering and without sorting.

The order in which the episodes are chosen has an impact on the episodes eliminated by the filtering, hence the importance of the choice of the sorting criteria and the possibility for the user to undo episode selection. These results show that it is important to have several measures to take into account the various characteristics of the data. For example, in the musical domain, it is important to observe the pieces before the analysis, to take into account their specificities and then, to choose the appropriate measures. Subjective measures have not been introduced in this work but would be probably helpful to complete these objective measures.

## 6 Discussion

In the literature, interactive approaches to knowledge discovery rely most of the time on visualizing data and mining results along different perspectives, and interactions with the user aim at changing the point of view on the data and filtering the results in order to focus on a subset of the data, according to the mantra "*preview first, zoom / filter, details on request*" [8].

Our approach differs in the particular way the interactions with the user are taken into account. It is based on the notion of coverage reflecting the relative importance of an episode in the analyzed trace. Coverage is exploited in two very closely related ways to assist interpretation: It is used firstly as a interestingness measure to rank mining results, and secondly to filter episode as a result of a selection, acting a bit like a filter in the pattern space "around" a selected episode. The coverage is expressed in several ways: in number of events and in total duration, which makes it possible to characterize events associated or not with a duration. The noise measurement is also based on this principle because it derives from the coverage and can be compared with the compactness measure in [14]. Meanwhile, the choice of an interestingness measure and the filtering process are independent. Currently the measures proposed in TRANSMUTE are predefined, and a perspective is to enable the user to choose them among a set.

Our approach can also be compared with to *minimum description length* principle [4] but with a rather different goal. In the MDL approach, the aim is to select a set of patterns that best compress the data. This principle is found in an underlying way with the use of coverage in our approach, that expresses the relative importance of an episode in a sequence, and thus serves to order and filter them. However, the aim of our approach is not to compress data, but rather to explicit significant episodes and to record the interpretation of the expert with the analyzed trace. In addition, the MDL approach has limitations on the one hand in computing complexity and, on the other hand, difficulties when applied to sequential data [6]. From this perspective, an *interactive* MDL approach is promising because it gives the initiative to the user in order to fully take advantage of his knowledge, and together with the exploitation of computational capabilities to assist him with useful filtering mechanisms to eliminate results that have become obsolete as a result of previous choices. Finally, the MDL approach is only applicable if the analyzed data is suitable for compression. This may often not be the case, for example for traces collected from technology enhanced learning environments, where further experimentation are still needed to evaluate the effectiveness of the approach. TRANSMUTE has been used to analyze the traces of the serious game TAMAGOCOURS and the first results are encouraging, but a comprehensive evaluation remains to be conducted to conclude the effectiveness of the interpretation strategy.

Experiments showed that the effectiveness of the coverage to focus quickly on interesting episodes vary a lot between the various musical pieces. It is therefore essential to propose several measures to allow the user making several trials. Currently, the measures implemented in the TRANSMUTE tool are predefined, and a perspective to this work is to allow the user to choose other objective or subjective interestingness measures, but also measures that express domain knowledge. We previously experimented a domain-dependent interestingness measure, such as the beginning musical sentence boundary. Most of the time in music, all the pieces start at the same moment in a mea-

sure (first beat or last half of the last beat, *etc.*). Once implemented, this property can be taken into account in the form of a domain-specific constraint which makes it possible to filter a very large number of redundant and irrelevant patterns.

The proposed evaluation is based on episodes provided by a domain expert, the expert episodes, but the experiment consisted in automatically simulating the selection of expert episodes. It gives a first idea of the gain in ranking episodes, and consequently in expert effort that can hypothetically be expected from this approach. A first qualitative evaluation was carried out with a group of five non-specialist users. They were asked to retrieve a set of given expert patterns from a score extract and then to complete a questionnaire to assess the usability of the prototype. They were provided a user guide and the mining parameters, and the evaluation focused exclusively on the interpretation. Users have all been able to easily identify the expert patterns and found TRANSMUTE easy to use. Nevertheless, this evaluation was carried out on a small sequence with a reduced number of expert patterns and there still needs to confront a user with much larger data in order to evaluate the time saved and the help provided by the assistance to interpretation.

The TRANSMUTE prototype that implements this approach has limitations due to its interface, and to a lack of optimisation. TRANSMUTE is limited since it is able to process a single small trace at a time, and does not allow to process too many episodes (a few thousands of events and episodes) and work still remain to be done for leveraging this issue. Meanwhile a separate optimized version of the filtering process has been implemented and shows reasonable execution time for a few million episodes. The KD process implemented in the DISKIT module is however able to process large traces and huge numbers of episodes (a few millions episodes). Moreover, a recent version allows processing several traces, the filtering operation can process a few millions episodes and enables to undo user's episode selections. The next step is to develop a new interface to display and analyze multiple traces.

## 7 Conclusion and future work

We have presented an approach for interactive and iterative interpretation of serial episodes. This approach rests upon interestingness measures to sort episodes, an interactive visualization tool where the expert can immediately see the impact of his actions on the sequence and a dynamic filtering process following an episode selection. This approach is reified in the TRANSMUTE web-based prototype where the user can annotate the selected episode by creating new labeled events and storing a transformed trace into the data store to save the interpretation work. The experiments show a significant improvement in the ranking of episodes, which augurs an easier identification of interesting episodes.

## References

1. W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus. Knowledge discovery in databases: An overview. *AI Magazine*, 13(3):57–70, 1992.
2. Andreas Holzinger. Human-computer interaction and knowledge discovery (hci-kdd): What is the benefit of bringing those two fields to work together? In *Availability, Reliability, and security in Information Systems and HCI*, pages 319–328. Springer, 2013.
3. Matthijs van Leeuwen. Interactive data exploration using pattern mining. In *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, pages 169–182. Springer, 2014.
4. J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465 – 471, 1978.
5. Jilles Vreeken, Matthijs Leeuwen, and Arno Siebes. Krimp: mining itemsets that compress. *Data Mining and Knowledge Discovery*, 23(1):169–214, 2011.
6. Hoang Thanh Lam, Fabian Mörchen, Dmitriy Fradkin, and Toon Calders. Mining compressing sequential patterns. *Statistical Analysis and Data Mining*, 7(1):34–52, 2014.
7. Enrico Bertini and Denis Lalanne. Surveying the complementary role of automatic data analysis and visualization in knowledge discovery. In *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration*, pages 12–20. ACM, 2009.
8. Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.
9. David Gotz, Fei Wang, and Adam Perer. A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data. *Journal of biomedical informatics*, 48:148–159, 2014.
10. Frederic Stahl, Bogdan Gabrys, Mohamed Medhat Gaber, and Monika Berendsen. An overview of interactive visual data mining techniques for knowledge discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(4):239–256, 2013.
11. Vladimir Dzyuba, Matthijs Van Leeuwen, Siegfried Nijssen, and Luc De Raedt. Active preference learning for ranking patterns. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 532–539. IEEE, 2013.
12. Tijn De Bie. Subjective interestingness in exploratory data mining. In *International Symposium on Intelligent Data Analysis*, pages 19–31. Springer, 2013.
13. Mario Boley, Michael Mampaey, Bo Kang, Pavel Tokmakov, and Stefan Wrobel. One click mining: Interactive local pattern discovery through implicit preference and performance learning. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, pages 27–35. ACM, 2013.
14. Nikolaj Tatti. Discovering episodes with compact minimal windows. *Data Min. Knowl. Discov.*, 28(4):1046–1077, July 2014.
15. Mirco Nanni and Christophe Rigotti. Extracting trees of quantitative serial episodes. In Sašo Džeroski and Jan Struyf, editors, *Knowledge Discovery in Inductive Databases: 5th International Workshop, KDID 2006 Berlin, Germany, September 18, 2006 Revised Selected and Invited Papers*, pages 170–188, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
16. Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259–289, 1997.