Process Comparison Based on Selection-Projection Structures

Tobias Brockhoff(✉), Merih Seran Uysal, and Wil M.P. van der Aalst

Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany
{brockhoff,uysal,wvdaalst}@pads.rwth-aachen.de

Abstract. Insight into differences between different implementations of a process provides valuable information for improvement. Process comparison approaches leverage event data on process executions to provide such insight. However, state-of-the-art procedural methods are often limited to *local* differences considering activities executed within a limited number of steps (e.g., directly following activities). Thereby, detecting differences which, for instance, relate early steps of a process execution to its outcome remains challenging. In contrast, rule-based declarative approaches can detect *global* differences with respect to distant activities; yet they are limited by the *complexity of the rule templates employed*. Moreover, they are prone to yield *fragmented* diagnostics. If a subprocess occurs more frequently in one process variant, these approaches typically report each activity contained. In this work, we therefore propose a process comparison approach that detects *aggregated* likelihood differences for *global* control-flow patterns. To this end, we decompose the difference detection task into subprocesses induced by co-occurring activities. Using Earth Mover's Distance, we identify differences within individual subprocesses independent of predefined rule templates. We then aggregate and combine subprocesses which distinguish the process variants. By exploiting relations among subprocesses, we retrieve *maximal* differences affecting many activities. Reducing fragmentation caused by choice-induced frequency differences, we additionally *complement* these *maximal* differences. To compare the sensitivity of our difference detection method to existing approaches, we devise a quantitative evaluation framework. Moreover, we demonstrate the effectiveness of our method on a public, real-life event log. Ultimately, the evaluation shows that our method is accurate and capable of providing coherent, global diagnostics.

Keywords: Process Mining · Process Comparison · Process Variant Analysis · Business Process Intelligence

1 Introduction

Operational processes are at the core of companies' value chains making active process management crucial for a company's success. Often, multiple variants of the same process exist (e.g., implementations at different facilities). Process

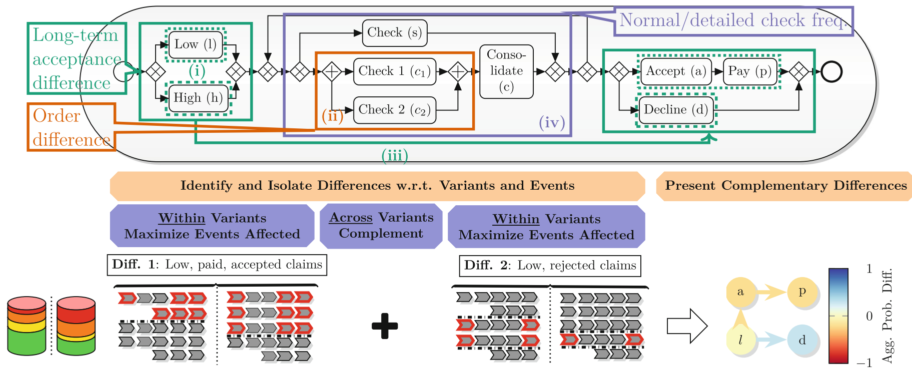


Fig. 1. Control-flow differences between process instances can be strongly entangled and concern distant parts of the process. In our approach, we decompose the difference detection in order to isolate activities and execution variants related to a difference. We then combine sets of complementary differences.

comparison methods aim to provide insight into differences between variants of a process denoting valuable information for process improvement. A common challenge is that there only exist event data of process executions where each event corresponds to a business transaction and is associated with a case (i.e., process instance), an activity, and a timestamp. Therefore, recent data-driven process comparison approaches take two event logs as input and return the (statistically significant) differences [4,7,14,19]. Yet, this can be challenging.

Consider the BPMN model of a claim management process shown in Fig. 1. Despite its simplicity, two instances of this process can differ in several ways. For example, they can differ with respect to (i) the frequency of low or high claims or (ii) in the probability of the detailed checks c_1 and c_2 being executed in a particular order. Moreover, these low-level differences might be further entangled. There can be additional dependencies between (iii) a claim's type and whether it is accepted, or (iv) the order differences of the checks is embedded in a frequency difference between normal and detailed checks. Besides, processes may differ in dimensions other than the (e.g., time). However, control flow has been the major concern of most existing works (see [20] for an overview), and we will focus on the former leaving extensions as future work.

State-of-the-art procedural approaches [4,14] are strong at identifying *local* differences—that is, differences concerning activities executed within a limited number of steps in the processes (e.g., directly following activities). However, detecting long-term differences such as the third differences illustrated in Fig. 1 is challenging. In contrast, the declarative approach proposed in [7] can detect differences for activities irrespective of their position in a case. However, it is limited to a fixed set of rule templates, and the implementation currently only supports activity pairs. Besides, reporting differences found for individual rules from a large corpus of rules is prone to yield fragmented diagnostics. For our example, considering the third—outcome-related—difference in Fig. 1, we obtain separate diagnostics showing that low claims are (1) accepted and (2) paid more frequently, but (3) declined less frequently. This motivates a process comparison approach that can provide *global* diagnostics that concern more than two activities. In particular, we want to detect differences that concern a *maximal* number of activities the individual cases (e.g., low claims are more likely to be accepted *and* paid). Besides, we want to incorporate *complementary* differences—that is, differences found for a distinct set of cases and caused by differing decision likelihoods (e.g., declining high claims more frequently)—to show the bigger picture. Ultimately, we obtain the *global*, *high-level* diagnostic shown in Fig. 1 by integrating all three diagnostics above.

In this work, we therefore propose an approach, which attempts to unify the strengths of both worlds, founded on the following research questions:

RQ(I) How can we reliably detect *global* control-flow differences?
RQ(II) How can we discover *maximal differences* that concern many activities?
RQ(III) How can we find differences across cases that complement each other?

To this end, we propose to analyze different sets of (possibly distant) activities. Thereby, we aim to isolate differences in terms of the cases and activities they concern improving the sensitivity of our approach to detect differences. Exploiting relations between the sets of activities, we aggregate sets of activities maximizing the number of activities concerned. Ultimately, we visualize differences in the process induced by each maximal set of activities. In doing so, we consider a context of complementary differences to show the bigger picture. The concept is shown on the bottom of Fig. 1 where we eventually identify a difference regarding the type and outcome of a claim. Conceptually, the approach is inspired by the variant and activity sliders implemented by most process mining tools. For process discovery, reducing the number of variants and activities reveals frequent control-flow patterns. In contrast, we consider not one but two process instances and attempt to find a configuration that shows a strong, coherent difference between the two instances.

Our contributions are as follows: we propose a method to detect *global* control flow differences based on the analysis of different sets of activities. It does not rely on predefined rule templates and proves to be highly sensitive. Moreover, we propose a two-step method *maximizing* and *combining* distinctive activity sets to reduce fragmentation of diagnostics. Compared to recent procedural approaches, latter combination step allows visualizing a particular difference in the context of other differences which can foster new interpretations.

We discuss related work and introduce preliminary concepts in Sects. 2 and 3. Next, we propose our method in Sect. 4, which we quantitatively and qualitatively evaluate in Sect. 5. Finally, we give our conclusion in Sect. 6.

2 Related Work

On a high level, one can distinguish process comparison approaches that devise differences from a process model or directly from data. Former approaches either require models as input [2,10], first discover a model [6,18], or enhance models [9]. For an in-depth discussion, we refer to the survey in [20].

An early application of graph comparison to process models proposes to visually compare them with respect to a merged, specially layouted graph [2]. In [10], Küster et al. consider change operations in the SESE-decompositions of UML activity diagrams. Yet, the authors in [1] argue that one must consider event data instead of models to make process comparison actionable.

Kriglstein et al. [9] complement structural diagnostics on model elements by differences in their use. Suriadi et al. discover process models and compare them with respect to how well they fit the other variants [18]. A similar approach, focussing on the mutual replay results, is proposed in [6]. An overview of different model-based visualizations is given in [16].

Log-based methods devise differences directly from event logs. An approach using sequential pattern mining is proposed in [11]. Van Beest et al. represent the process variants with prime event structures [3], align them, and verbalize differences. However, the approach requires a concurrency oracle. In [4], statistically significant differences are considered. The authors create a shared Transition System (TS) and apply hypothesis tests to detect performance and frequency differences. Yet, there is a trade-off between the expressiveness of the TS and its size. Taymouri et al. [19] create directly-follows graphs (DFGs), so-called Mutual Finger Prints (MFPs), from cases which they consider distinctive for each variant. To this end, they extract location- and frequency-aware features from a discrete wavelet transformation and discover the most distinctive subset of features by training and evaluating classifiers. Finally, they select the cases containing these features. However, the computational complexity of the method is high. A declarative approach is proposed in [7]. The authors instantiate a set of rule templates and test for differences. A log-based approach that allows to consider perspectives beyond control flow and performance is proposed in [14]. Finally, an interactive process comparison framework, applying filters upfront, was proposed in [21] . In contrast, we consider filtering an essential part of process comparison itself.

Our decomposition is related to concept lattices in Formal Concept Analysis (FCA) [17,22]; yet, we explicitly consider activity order.

3 Preliminaries

We denote the powerset of a set X by $\mathcal{P}(X)$ and the bags over X by $\mathcal{B}(X)$. For example, $M = [a^5, b]$ is a bag of size $|M| = 6$ containing a five times. In an abuse of notation, we overload set operators for bags (e.g., $a \in M$ and $[a^3] \subset M$). A directed graph is a tuple $G = (V, E)$ of a set of vertices and a directed edge relation $E \subseteq V \times V$. For brevity, we use an infix notation to denote edges—for example, for $v_1, v_2 \in V$, the vertex v_1 is a predecessor of v_2 if $v_1 \; E \; v_2$ holds. The transitive reduction of G is the graph $G' = (V, E')$ with the fewest edges and the same pairwise reachability of vertices as G.

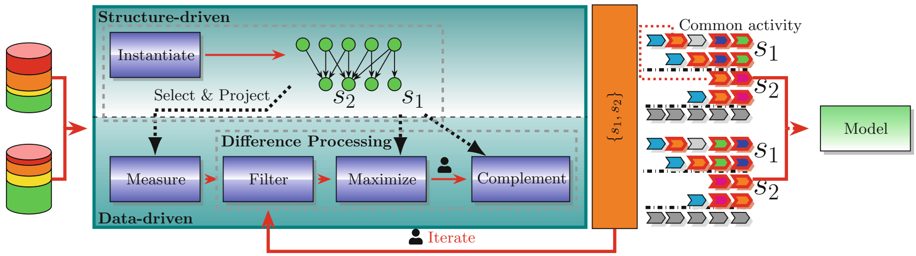


Fig. 2. Overview of the approach. We instantiate a measurement structure where vertices correspond to a subaspects of the process and edges relate the former. Using this structure, we measure differences and filter the initial results. Afterward, we maximize differences making them as specific as possible. Finally, we complement user-selected differences by related differences and use the associated variants and activities to visualize the result.

Event Data. Let $\mathcal{A}$ denote the universe of activity labels, and let $\Sigma \subseteq \mathcal{A}$ be a finite set of activities. A *trace* $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle \in \Sigma^*$ is a finite sequence of activities. The length of σ is $|\sigma| = n$. A trace $\sigma', |\sigma'| = m$ is a *subtrace* of σ, denoted by $\sigma' \sqsubseteq \sigma$, if there exist indices $1 \leq i_1 < \cdots < i_m \leq n$ such that $\sigma' = \langle \sigma_{i_1}, \ldots, \sigma_{i_m} \rangle$. Finally, we write $\{\sigma\} = \{\sigma_i \mid 1 \leq i \leq |\sigma|\}$ to denote the set of distinct activities in σ. An *event log* collects multiple executions of a process.

Definition 1 (Event Log). *Given a finite alphabet $\Sigma \subseteq \mathcal{A}$, an* event log $E \in \mathcal{B}(\Sigma^*)$ *is a finite bag of traces over Σ.*

For an event log $L \in \mathcal{B}(\Sigma^*)$, its empirical trace distribution is defined by the probability mass function $p_E \colon \Sigma^* \to [0,1], \sigma \mapsto \frac{E(\sigma)}{|E|}$. This is also called the *stochastic language* of E [12]. Let $\delta \colon \Sigma^* \times \Sigma^* \to [0,1]$ be a so-called *trace distance*—that is, a function that quantifies the dissimilarity between pairs of traces. Given event logs $L_1, L_2 \in \mathcal{B}(\Sigma^*)$, the Earth Mover's Distance (EMD) quantifies the dissimilarity of the associated stochastic languages [12]—that is, $\mathrm{emd}_\delta(p_{L_1}, p_{L_2}) \in [0,1]$. In the following, we assume that normalized edit distance (edt) is used as trace distance.

4 Selection-Projection-Based Difference Discovery

We propose a two-stage process comparison approach that separates the detection and aggregation of differences. Figure 2 provides an overview and demonstrates how we distinguish between a structure-driven and data-driven aspect of our method. On the structural side, we create a lattice of commonly co-occurring sets of activities and use them to select and project control-flow variants of the process. This results in pairs of filtered sub-event logs extracted from the original event logs, representing data on subprocess executions. For example, in Fig. 2, the subprocess s_1 is associated with three activities, highlighted red, from two variants in each event log. By *measuring* the differences between each pair of sub-event logs using EMD, we then identify sets of activities for which the two process variants significantly differ in frequency or activity execution order. To consolidate differences related to the same variants, we *aggregate* them based on structural relations between subprocesses. This reduces fragmentation of frequency differences with respect to entire subprocesses such as the frequency difference regarding low and accepted as well as low and paid claims in Fig. 1. Additionally, we propose integrating complementary differences that are closely related but involve a different set of control-flow variants. For instance, in Fig. 2, we complement s_1 by s_2 since latter subprocess focuses on a distinct set of variants while sharing one activity. Ultimately, we create a model for the differential analysis of the retrieved *complementary* and *maximal* subprocesses. In the following, we illustrate our approach on the following two—left and right ($l|r$)—event logs generated from the process shown in Fig. 1:

$$\begin{aligned} L^{l|r,\mathrm{ex}} = [&\langle l, c_1, c_2, c, \mathrm{a}, \mathrm{p}\rangle^{42|50}, \langle l, c_1, c_2, c, \mathrm{d}\rangle^{9|3}, \langle l, c_2, c_1, c, \mathrm{d}\rangle^{9|7}, \\ &\langle h, c_1, c_2, c, \mathrm{a}, \mathrm{p}\rangle^{28|20}, \langle h, c_1, c_2, c, \mathrm{d}\rangle^{6|7}, \langle h, c_2, c_1, c, \mathrm{d}\rangle^{6|13}] \end{aligned} \tag{1}$$

Selection-Projection Structure As depicted on the left of Fig. 2, one can think of an event log as a list of cases or, focusing on the control flow, variants. To isolate control-flow differences, we can "horizontally" select entries (i.e., variants) and "vertically" remove activities (i.e., activity projection). While for process discovery one typically focuses on frequent variants and activities, we suggest choosing and projecting traces based on frequently co-occurring activities for process comparison. To this end, let $L \in \mathcal{B}(\Sigma^*)$ denote an event log over an alphabet $\Sigma \subseteq \mathcal{A}$ and $\Sigma' \subseteq \Sigma$ be set of activities. *Activity projection* $\pi^{\mathrm{act}}_{\Sigma'} \colon \Sigma^* \to \Sigma^*$ projects traces on Σ'. For a trace $\sigma \in \Sigma^*$, $\pi^{\mathrm{act}}_{\Sigma'}(\sigma)$ is the longest subsequence of σ over Σ'—that is, $\pi^{\mathrm{act}}_{\Sigma'}(\sigma) = \arg\max_{\tau \in (\Sigma')^*, \tau \sqsubseteq \sigma} |\tau|$. Similarly, *activity-based trace selection* $\mu^{\mathrm{act}}_{\Sigma'} \colon \Sigma^* \to \Sigma^*$ keeps a trace σ if all activities in Σ' occur—that is, $\mu^{\mathrm{act}}_{\Sigma'}(\sigma) = \sigma$ if $\Sigma' \subseteq \{\sigma\}$; otherwise, $\mu^{\mathrm{act}}_{\Sigma'}(\sigma) = \langle\rangle$. Assuming that event logs initially do not contain empty traces, we do not model discarded traces by a dedicated symbol to unify the notation. Using an element-wise application, we can concatenate the functions and apply them to event logs to isolate control-flow aspects of a process For example, the event log $L^{l,\mathrm{ex}}_{\pi^{\mathrm{act}}_{\{l,\mathrm{d}\}} \circ \mu^{\mathrm{act}}_{\{l,\mathrm{d}\}}} = \left[\langle l, \mathrm{d}\rangle^{18}, \langle\rangle^{82}\right]$ focuses on the frequency and activity order of low and declined claims.

While can we consider different sets of activities to study different control-flow aspects, we can also use them to further process the retrieved differences. In doing so, we exploit that activity sets are naturally related in terms of specialization. To this end, consider Fig. 3 that shows the effect of selecting variants (y-axis) and projecting the selected variants on potentially different sets of activities (x-axis). In particular, we consider the case where we use the same set of activities to select and project (highlighted in orange). *The behavior* based on which we select the variant is the *same* behavior that we extract from the former. Besides, this also establishes a specialization relation regarding the data extracted for activity sets $s_1, s_2 \in \mathcal{P}(\mathcal{A}), s_1 \subset s_2$. For s_2, we consider a subset

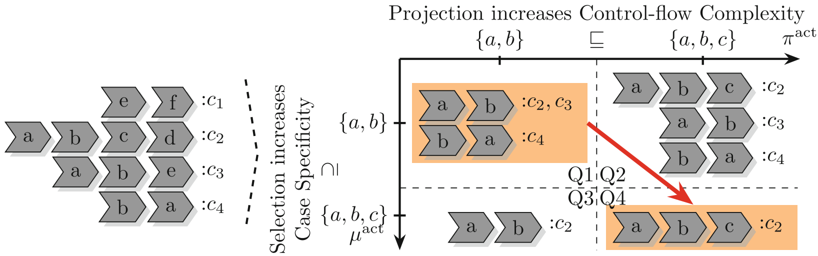


Fig. 3. By extending the activity set, we extract subtraces with increasingly complex control flow from a more specific (i.e., smaller) subset of traces. Here, we eventually isolate a single variant.

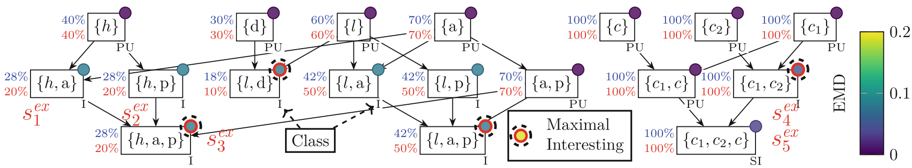


Fig. 4. SPS for our running example. Each vertex is a set of frequently co-occurring activities. It is annotated by the activity set's occurrence probability in the left and right log, its associated EMD, and class. Dashed red circle show the *maximal interesting* activity sets (comp. Definition 3). (Color figure online)

(i.e., fewer) of the variants, and the control flow becomes more complex—that is, we consider a superset of the events. In contrast, disentangling the selection and projection can blur the localization (e.g., a difference measured for Q2 might optionally involve c), or it might result in diagnostics that are less general than expected. For instance, differences found for Q3 do not necessarily concern a and b in general. Exploiting the specialization relation, we define the Selection-projection Structure (SPS)—the backbone of our approach.

Definition 2 (SPS). *Let $L^l, L^r \in \mathcal{B}(A^*), \langle\rangle \notin L^l, \langle\rangle \notin L^r$ denote event logs over an alphabet $A \subseteq \mathcal{A}$. A Selection-projection Structure (SPS) $\mathrm{sps}_{L^l,L^r} = (S, <, m^{\mathrm{sps}}_{L^l,L^r})$ for L^l and L^r is a triple of a set of activity sets $S \subseteq \mathcal{P}(A)$, an edge set $< \subseteq S \times S$ such that $(S, <)$ is the transitive reduction of the graph $(S, \{(s_1, s_2) \in S \times S \mid s_1 \subset s_2\})$, and an EMD-based vertex measurement function*

$$m^{\mathrm{sps}}_{L^l,L^r} : S \to [0,1], s \mapsto \mathrm{emd}_{\mathrm{edt}}(p_{L^l_{\pi^{\mathrm{act}}_s \circ \mu^{\mathrm{act}}_s}}, p_{L^r_{\pi^{\mathrm{act}}_s \circ \mu^{\mathrm{act}}_s}}). \tag{2}$$

Figure 4 shows an SPS for our running example comprising sets of *frequently co-occurring* activities. We measure a non-zero difference for nine vertices in total, yet multiple vertices refer to the same difference. For example, the sets $\{h, \mathrm{a}\}$, $\{h, \mathrm{p}\}$, and $\{h, \mathrm{a}, \mathrm{p}\}$ all indicate a difference regarding the acceptance (a) and payment (p) of high claims (h).

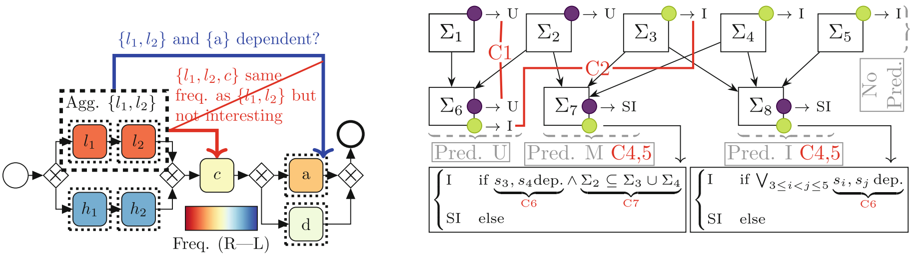
Maximization. In an SPS, many vertices may witness the same difference which can result in fragmented diagnostics. Yet, the specialization relation between vertices in the SPS gives us a means to *maximize* differences: return the more specific vertex, if it perfectly extends the differences measured for its predecessors. However, this requires an accurate characterization of the mechanisms behind the difference scores (e.g., frequency difference, order differences, or even combinations of the former). We therefore propose a relaxed difference maximization approach based on the vertices' EMD values and leave other implementation for future work. If two related activity sets have similar difference scores, they are likely to witness the same difference, and we return the more specific one (i.e., superset). However, Fig. 5 exemplifies limitations of this naive approach. To this end, consider two process instances differing in two ways: in the right process instance (1) more low claims are registered (i.e., l_1, l_2), and (2) more claims are accepted. The color of a vertex shows its frequency which is directly related to its EMD value. The activities l_1 and l_2 witness the same frequency difference and aggregating them shows a difference for a coherent branch of the process. In contrast, we argue that $\{l_1, l_2, c\}$ is not an interesting difference, despite it has the same support as $\{l_1, l_2\}$. It comprises the consolidation step c which is always executed and therefore not interesting. Moreover, we measure a strong difference for $\{l_1, l_2, \mathrm{a}\}$. Nevertheless, the choices related to $\{l_1, l_2\}$ and $\{\mathrm{a}\}$ might be independent making the set less interesting. Based on these considerations, we bottom-up classify the SPS vertices as *interesting* (I), *uninteresting* (U), and *sub-interesting* (SI) and retrieve the most specific, interesting differences.

Definition 3 (SPS Difference Classification). *Let $L^l, L^r \in \mathcal{B}(A^*), \langle\rangle \notin L^l, \langle\rangle \notin L^r$ denote event logs over an alphabet $A \subseteq \mathcal{A}$; $\tau^i_m, \tau^i_i \in [0,1]$ be thresholds; and $\mathrm{sps}_{L^l,L^r} = (S, <, m^{\mathrm{sps}}_{L^l,L^r})$ be an SPS. The SPS difference class of a vertex is given by $\bar{\kappa} := \kappa_{\mathrm{sps}_{L^l,L^r}, \tau^i_m, \tau^i_i} : S \to \{\mathrm{U}, \mathrm{SI}, \mathrm{I}\}$,*

$$\bar{\kappa}(s) = \begin{cases} \mathrm{U} \;\; \textit{if} \;\; \bar{m}(s) < \tau^i_m \land \forall s_1\,(s_1 < s \to \bar{\kappa}(s_1) = \mathrm{U}) & \textit{(C1 All U)} \\ \mathrm{I} \;\; \textit{if} \;\; \bar{m}(s) \geq \tau^i_m \land \{[\forall s_1\,(s_1 < s \to \bar{\kappa}(s_1) = \mathrm{U})] & \textit{(C2 - Pred. U)} \\ \qquad \lor [\exists^{=1} s_1\,(s_1 < s) \land \exists^{=1} s_1\,(s_1 < s \land \bar{\kappa}(s_1) = \mathrm{I})] & \textit{(C3 - One Pred. I)} \\ \qquad \lor [\nexists s_1 \bar{\kappa}(s_1) = \mathrm{SI} & \textit{(C4 - No pred. SI)} \\ \qquad \land \exists s_1, s_2 (s_1 < s \land s_2 < s \land \bar{\kappa}(s_1) = \mathrm{I} & \textit{(C5 - Pred. I)} \\ \qquad \land\, \bar{\kappa}(s_2) = \mathrm{I} \land \phi_{L^l,L^r}(s_1, s_2) > \tau^i_i & \textit{(C6 - Not indep.)} \\ \qquad \land \bigcup\limits_{s_1 < s, \bar{\kappa}(s_1) = \mathrm{U}} s_1 \subseteq \bigcup\limits_{s_1 < s, \bar{\kappa}(s_1) = \mathrm{I}} s_1)]\} & \textit{(C7 - Prove I)} \\ \mathrm{SI} \;\; \textit{else} & \end{cases} \tag{3}$$

for $s \in S$, $\bar{m} := m^{\mathrm{sps}}_{L^l,L^r}$, and ϕ denoting the phi-coefficient of the occurrence of two given activity sets. Given a domination factor $\tau^i_d > 1$, a vertex $s \in S$ is maximal interesting if and only if

$$\bar{\kappa}(s) = \mathrm{I} \land \forall s_1 \left(s < s_1 \land \bar{\kappa}(s_1) = \mathrm{I} \to m^{\mathrm{sps}}_{L^l,L^r}(s) > \tau^i_d m^{\mathrm{sps}}_{L^l,L^r}(s_1)\right). \tag{4}$$



(a) Fundamental Considerations (b) Classifying SPS-vertices

Fig. 5. Maximizing differences in the SPS graph.

Figure 5b illustrates the conditions. The EMD value determines the class of a vertex without predecessors (note that the "for all" statements in C1,2 hold). Otherwise, we consider its predecessors: if all predecessors are *uninteresting*, we again consider the EMD value (C2). For example, in Fig. 4, the vertex $\{c_1, c_2\}$ becomes interesting due to the activity order difference. If all predecessors are *interesting*, we not only consider EMD but also whether there are dependent predecessors (C 6). Similar conditions apply if the predecessors are *either interesting or uninteresting*. However, the *uninteresting* predecessors indicate that we might aggregate an uninteresting subaspect (comp. c in Fig. 5a). Therefore, we additionally require that each activity is interesting with respect to at least one subaspect (C 7). For example, we do not observe a difference regarding the acceptance subprocess $\{\mathrm{a}, \mathrm{p}\}$, but in the context of high claims—namely, $\{h, \mathrm{a}\}$ and $\{h, \mathrm{p}\}$—acceptance is more likely in the left log. A vertex with *interesting* predecessors that cannot be proved *interesting* is labeled *sub-interesting*. We found evidence that we cannot further maximize an *interesting* predecessor, and, in case of doubt, we follow Occam's razor and opt for the simpler difference.

Eventually, we report the *maximal interesting vertices*—that is, interesting vertices that do not have an interesting successor with a similar EMD value. In Fig. 4, the scores of these vertices are highlighted in red using, for example, the thresholds $\tau_m^i = 0.01, \tau_i^i = 0.2, \tau_d^i = 1.2$.

Complementary Differences. Conceptually, each *maximal interesting vertex* captures differences with respect to *observed* activities. Thinking of BPMN, this covers sequences of activities, order resolution of concurrency, and loop repetitions. However, a single set of observed activities cannot entirely explain choices. A simple difference in the likelihood of choosing between two activities a and b, results in significant EMD values for $\{a\}$ and $\{b\}$ (but not for $\{a, b\}$). Yet, showing both sets at the same time would paint a clearer picture. Note that the declarative method proposed in [7] faces a similar challenge. We therefore propose to search additional, complementary differences. Let $\mathrm{sps}_{L^l,L^r} = (S, <, m^{\mathrm{sps}}_{L^l,L^r})$ be an SPS for event logs L^l and L^r and $S^{\mathrm{int}}_{\mathrm{sps}_{L^l,L^r}}$ denote the set of maximal interesting differences. We propose a three-step filter pipeline where we assess whether two vertices $s_1, s_2 \in S^{\text{int}}_{\text{sps}_{L^l,L^r}}$ occur complementary by the Jaccard index of their co-occurrences in the event logs—namely, $J^{\text{occ}}_{L^l,L^r}(s, s_1) := \min_{x=l,r} \frac{|L^x_{\mu^{\text{act}}_{s_1 \wedge s_2}}|}{|L^x_{\mu^{\text{act}}_{s_1 \vee s_2}}|}$. A low score indicates that, in (at least) one process instances, cases that comprise one activity set do not comprise the other. Given a vertex $s \in S^{\text{int}}_{\text{sps}_{L^l,L^r}}$ and a bound on the vertices' co-occurrence $\tau^c_{jacc} \in [0, 1]$, the filtering steps are:

(1) remove structurally related differences
$S^{\text{int}\,1}_{\text{sps}_{L^l,L^r}} := S^{\text{int}}_{\text{sps}_{L^l,L^r}} \setminus \left\{ s_1 \in S^{\text{int}}_{\text{sps}_{L^l,L^r}} \,\middle|\, s_1 \subseteq s \vee s \subseteq s_1 \right\}$;
(2) remove co-occurring vertices
$S^{\text{int}\,2}_{\text{sps}_{L^l,L^r}} := S^{\text{int}\,1}_{\text{sps}_{L^l,L^r}} \setminus \left\{ s_1 \in S^{\text{int}\,1}_{\text{sps}_{L^l,L^r}} \,\middle|\, J^{\text{occ}}_{L^l,L^r}(s, s_1) > \tau^c_{jacc} \right\}$;
(3) sort vertices $s_1 \in S^{\text{int}\,2}_{\text{sps}_{L^l,L^r}}$ by increasing vertex similarity $\frac{s \cap s_1}{s \cup s_1}$, decreasing co-occurrence $J^{\text{occ}}_{L^l,L^r}(s, s_1)$, and similar EMD $|m^{\text{sps}}_{L^l,L^r}(s) - m^{\text{sps}}_{L^l,L^r}(s_1)|$.

In the first step, we exploit the structural relation to discard vertices that are naturally not complementary.

Visualization. Given a set of complementary activity sets $S^{\text{co}} \subseteq S$ for an SPS $\text{sps}_{L^l,L^r} = (S, <, m^{\text{sps}}_{L^l,L^r})$, we illustrate differences using a directly follows-based visualization. Like the approach proposed in [19], we focus on the traces that constitute the difference, yet we show both logs in the same graph to facilitate the visual alignment. To create the DFG, we first generalize the trace selection to multi-activity set conditions. Given activity sets $\Sigma_i, i = 1, \ldots, n$, $\mu^{\text{act}}_{\Sigma_1 \vee \cdots \vee \Sigma_n}$ ($\mu^{\text{act}}_{\Sigma_1 \wedge \cdots \wedge \Sigma_n}$) keeps a trace if any (all) of the individual trace selections $\mu^{\text{act}}_{\Sigma_i}, i = 1, \ldots, n$ do. Selecting all relevant traces and projecting on the involved activities, we obtain two pairs of event logs:

$$L^{x,\text{co}} := L^x_{\pi^{\text{act}}_{\bigcup_{s \in S^{\text{co}}} s} \circ \mu^{\text{act}}_{\bigvee_{s \in S^{\text{co}}} s}}, L^{x,\text{co},\neq\langle\rangle} := [\sigma \in L^{x,\text{co}} |\, \sigma \neq \langle\rangle], x = l, r. \quad (5)$$

Note that the second pair of event logs only contains relevant traces. Since the initial event logs may have different sizes, we scale by the size of the logs. For $L^{x,\text{co}}$ ($L^{x,\text{co},\neq\langle\rangle}$), $x = l, r$, each DFG edge thereby shows the expected (conditional) number of occurrences in the log (given that the trace is relevant). Conditioning allows to analyze qualitative differences within the subprocess defined by the selected variants irrespective of its global frequency. We refer to these graphs as Trace-probability DFGs (TP-DFGs) as the edge values are aggregated trace likelihoods. Figure 6a shows a TP-DFG on real-life data where we added information on the considered activity sets to an artificial start vertex.

5 Evaluation

We evaluate the Java implementation of our approach (SPS)[1] with respect to the research questions, and compare it to Bolt's approach (TS-PC) [4] and Cecconi's

[1] https://github.com/tbr-git/procmin-apps.

method (DecPC) [7]. We quantitatively evaluate RQ(I) and assess RQ(II) and RQ(III) in a case study. In the quantitative evaluation, we also consider EMD ($\mathrm{emd}_{\mathrm{edt}}$) between the complete logs as we only require a score.

For SPS, we consider the 10,000 most frequent activity sets. For TS-PC, we use the default p-value of 0.05 with the default abstraction (TS-PC-d)—i.e., 1-set history abstraction—and 2-sequence history abstraction as in [4] (TS-PC-s). For DecPC, we evaluate the default parameters (DecPC-d) and the significantly differing parameterization used for the evaluation presented in [7](DecPC-s).

5.1 Case Study

We conduct a real-life case study to compare and evaluate the considered approaches with respect to RQ(II) and RQ(III). To this end, we consider the well-known Road Traffic Fine Management log [13], split into a left and right log containing low (< 50€) and high fines (≥ 50€) [7,19], which was also investigated in [4,19]. We analyze the logs with respect to two evaluation questions:

EQ(I) Are there execution patterns with respect to the control flow that are more likely in either variant?
EQ(II) Are there qualitative differences in the execution of certain subprocesses?

EQ(I) is related to RQ(I) and RQ(II), and RQ(III) is the foundation to assess EQ(II). We identify a subprocess by co-occurring activities where retrieving complementary difference accommodates for choices. Eventually, the conditioned TP-DFG shows qualitative differences. In contrast to DecPC, which currently only supports binary rules (e.g., if a occurs, b is more likely to occur), we implicitly condition on multiple activity sets. We run our process comparison method in an automatic mode. We consider the five strongest *maximal differences* as seeds and complement each by (at most) three additional activity sets. In doing so, we skip seeds that were already shown as complementary differences. To discover the maximal interesting SPS-vertices and complementary differences, we use the thresholds $\tau_m^i = 0.001$, $\tau_d^i = 0.9$, $\tau_s^i = 1.2$, $\tau_i^i = 0.05$, and $\tau_{jacc}^c = 0.2$. An analysis of the sensitivity of our method to these parameters as well as additional results can be found online[2].

Results. Table 1 summarizes the results obtained by our method and DecPC. Table 1a shows four sets of complementary activity sets together with each activity set's (conditional) probability. Besides, we depict the conditioned TP-DFG associated with the second set of complementary differences in Fig. 6a and the TS obtained using TS-PC in Fig. 6b.

Considering EQ(I), all approaches show that the activities SF, IFN, and AP, related to additional fining, more frequently occur for high claims (D(1)). However, DecPC splits it among multiple differences, and it remains up to the analyst to see the bigger picture. Similarly, there are more high-fine cases where an additional penalty (AP) is added *and* the fine is collected (SCC, D(2)) or

[2] https://doi.org/10.6084/m9.figshare.c.7167954.v1.

Table 1. Summary of results obtained for the RTFM log

(a) Discovered complementary SPS-vertex sets

No.	Activities	Prob. To. L	Prob. To. R	Cond. L	Cond. R	Log
1	$\{SF, IFN, AP\}$ (D(1))	0.12	0.69	1	1	
3	$\{P\}$ (D(4))	0.66	0.33	0.88	0.38	(0.75|0.87)
	$\{SF, IFN, AP, SCC\}$	0.09	0.51	0.12	0.58	
	$\{SF, IFN, IDA2P, AP, SA2P\}$ (D(5))	0	0.06	0.1	0.07	
	$\{SF, AP, A2J\}$ (D(5))	0	0.01	0	0.01	
4	$\{SF, P, IFN, AP, SCC\}$	0	0.02	0.41	0.21	(0.01|0.09)
	$\{SF, IFN, AP, IDA2P, SA2P\}$ (D(5))	0	0.06	0.56	0.68	
	$\{IFN, AP, A2J\}$ (D(5))	0	0.01	0.02	0.14	
	$\{P, IDA2P\}$ (D(5))	0	0.01	0.08	0.09	

(b) Top 10 distinguishing rules [7]

No.	Rule	LLH diff.
1	P then SF (D(6))	24.8%
2	P then AP (D(6))	19.57%
3	P then IFN (D(6))	19.57%
4	AP always occurs (D(1)(2))	19.41%
5	IFN always occurs	19.41%
6	CF then AP before CF	19.41%
7	CF then IFN before CF	19.41%
8	SF always occurs (D(1)(2))	19.19%
9	CF then SF before CF (D(1)(2))	19.19%
10	SCC always occurs (D(2))	14.47%

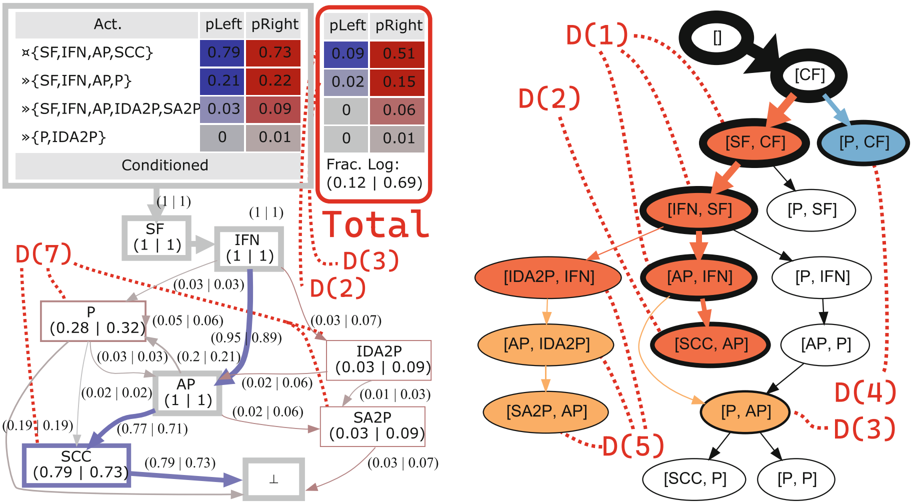
paid (P, D(3)). In contrast, a payment is made for the majority of low-fine cases (D(4)). While our approach clearly shows this difference, it is not included in the top-ten differences obtained by DecPC. TS-PC even shows that low fines are more likely to be paid immediately after creation. Finally, SPS and TS-PC indicate that traffic offenders who receive a high fine are more likely to appeal (A2J, SA2P, IDA2P, D(5)).

Considering RQ(III), the retrieved complementary differences define coherent subaspects. While payments ($\{P\}$) might semantically complement the first difference, D(3) shows that these often occur together. The second set of differences covers the outcomes of cases with an additional penalty—namely, payment, credit collection, or appeal. Similarly, the third set gathers the general outcomes. Finally, the fourth set is again concerned with additional fining, yet, the seed are cases including credit collections despite payments were made.

Finally, we assess EQ(II) using the conditioned TP-DFG. First, DecPC finds that if payment was made, high fines are more likely to contain an additional penalty (D(6)). Note the difference to the diagnostic: even though a fine was added, payments are more likely. In fact, the order of the activities in the rule is the opposite of their intuitive order. In contrast, the conditioned TP-DFG in Fig. 6a shows only small differences considering the outcome of cases with an additional fine. While credit collection is slightly more likely for low-fine cases, high-fine cases more frequently result in a payment or in an appeal (D(7)).

5.2 Quantitative Evaluation on Scoring Concept Drift

To the best of our knowledge, there exist no frameworks that evaluate the sensitivity of process comparison approaches. Thus, we propose to quantitatively evaluate approaches based on their ability to distinguish pairs of event logs that were extracted from a stable process and those that were extracted from differing processes. To this end, we use the values returned by approaches to quantify the severity of a detected difference. For SPS, we consider the largest EMD among the maximal interesting vertices. For TS-PC, we extract the largest Cohen's d value, and DecPC returns the difference in the confidence of distinguishing rules.



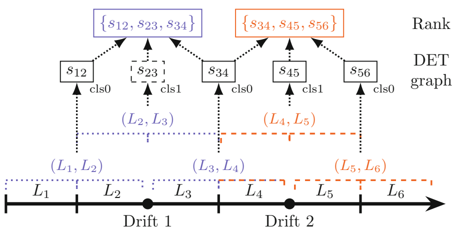
(a) Conditioned TP-DFG for second most-significant difference

(b) Transition system TS-PC

Fig. 6. Graph-based difference visualization using (a) our proposed method and (b) TS-PC. Due to the conditioning enabled by the variant selection, Subfigure (a) shows that low fines are less likely to be paid given that an additional penalty (AP) was added. In contrast, Subfigure (a) shows that additional penalties and credit collection (SCC) as well as appealing are more likely for high fines. For both graphs, additional frequency filtering has been applied.

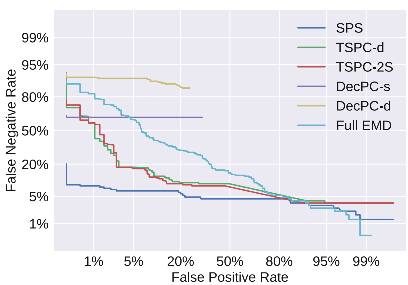
Regarding the classification, we consider this value a "confidence" score assuming a score of zero if no difference was detected. This is also how a user might intuitively interpret the number. In particular, we create process comparison tasks from collections of artificial event logs [5,8,15] created for Process Concept Drift Detection where each event log contains sudden changes at known positions. Figure 7a shows the extraction of five process comparison tasks where each log contains 50% of the cases of a stable period, and each pair of event logs is either extracted from same period or comprises across-drift event logs.

Finally, we evaluate process comparison approaches in two ways: first, we consider the Detection Error Tradeoff (DET) graph. Thereby, we assess how reliably high scores indicate strong differences—independent of the underlying process. Second, we analyze how drifts a ranked with respect to the process before and after. In contrast to the DET graph, this only considers similar processes. We distinguish the cases: no difference was detected, a method correctly scores the across-drift task higher than the other two tasks, and a remainder class. Note that our approach and EMD always return a score, and the first case therefore never applies. The top part of Fig. 7a shows both approaches.

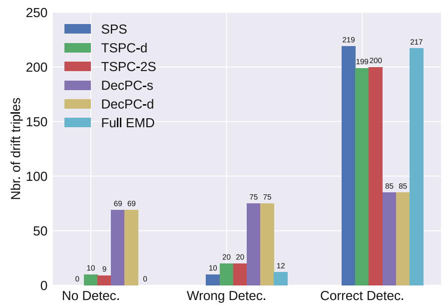


(a) Creating classification tasks from concept drift

(b) Time process comparison



(c) DET graph (lower is better)



(d) Ranking drift with respect to pre-/post-drift logs

Fig. 7. Classifying event logs whether they contain a drift.

Results. Figure 7 depicts the results for the extracted 609 classification tasks. The DET graph in Fig. 7c shows that SPS and TS-PC perform best in the global drift classification task. Compared to SPS, the false positive rate of TS-PC increases earlier meaning that TS-PC scores certain across-drift tasks lower than log pairs extracted from a stable process. EMD performs worse than the former two methods, yet it might become superior for complicated differences. For a high false positive rate, the curve falls beneath the curves for SPS and TS-PC. Finally, DecPC shows an interesting pattern. It assigns the highest scores to across-drift tasks; yet, at some point, there are many false positives. Eventually, the false negative rate drops to zero for a false positive rate larger than 20%. This can be explained by a large number of drift and non-drift process comparison tasks where no difference is detected. Such tasks receive the score zero.

The local ranking of drifts shown in Fig. 7d confirms the prior findings. SPS correctly ranks most process comparison tasks triples performing slightly better than TS-PC. Interestingly, EMD performs very well in this local context. While SPS aims to isolate differences before it scores them, EMD always considers the entire trace. Thereby, the cost contribution of a difference dependents on the trace's length. Consequently, EMD performs worse than SPS on the global drift classification task. Finally, the statistical test used by DecPC is, on the one hand, conservative resulting in many cases where no difference is detected at all. On the other hand, DecPC also ranks a couple task triples incorrectly.

Considering the time required for the comparison (i.e., excluding log loading times), Fig. 7b shows that DecPC is by far the slowest. The other approaches usually finish within seconds.

Discussion. The quantitative evaluation shows that our method outperforms existing approaches in the proposed difference classification. However, this evaluation is still based on artificial and does not assess the usefulness of the discovered differences. Besides, even though high EMD values in the SPS framework indicate strong differences, the individual value can be difficult to interpret; it may subsume different differences. Considering the performance, the EMD problem for each individual SPS vertex is usually simpler making it possible to quickly measure many vertices. Yet our implementation is highly concurrent, while the other approaches are single threaded. In the real-life case study, our approach aggregates similar differences better than DecPC. Compared to TS-PC, TS-PC manages to show almost all differences in a single graph, which is possible for such a relatively simple log. In contrast, our approach could detect an additional difference by specifically focusing on complementary variants.

6 Conclusion

We propose a process comparison approach that detects global and complex control-flow differences in event data extracted from an information system. To this end, we leverage event data projections to facilitate the comparison and to isolate differences. We then extend the isolated differences to cover larger fractions of the associated cases. Moreover, we propose an approach to complement a given difference by considering additional activities and cases. Explicitly identifying variants that induce a difference also gives a context for a refined analysis. We demonstrate the applicability of the method in a case study as well as quantitatively assess its sensitivity in a newly devised process comparison evaluation approach. For future work, we plan to improve the detection and presentation of conditional differences as well to incorporate other dimensions (e.g., time). Furthermore, we intend to explicitly test differences for statistical significance to improve the confidence in attained results.

Acknowledgments. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-EXC-2023 Internet of Production-390621612. We also thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

1. van der Aalst, W.M.P., de Medeiros, A.K.A., Weijters, A.J.M.M.: Process equivalence: comparing two process models based on observed behavior. In: BPM, pp. 129–144 (2006)

2. Andrews, K., Wohlfahrt, M., Wurzinger, G.: Visual graph comparison. In: 13th IV, pp. 62–67 (2009)
3. van Beest, N.R.T.P., Dumas, M., García-Bañuelos, L., La Rosa, M.: Log delta analysis: interpretable differencing of business process event logs. In: BPM, pp. 386–405 (2015)
4. Bolt, A., de Leoni, M., van der Aalst, W.M.P.: A visual approach to spot statistically-significant differences in event logs based on process metrics. In: CAiSE, pp. 151–166 (2016)
5. Bose, R.P.J.C., van der Aalst, W.M.P., Žliobaitė, I., Pechenizkiy, M.: Handling concept drift in process mining. In: CAiSE, pp. 391–405 (2011)
6. Buijs, J.C.A.M., Reijers, H.A.: Comparing business process variants using models and event logs. In: BPMDS, pp. 154–168 (2014)
7. Cecconi, A., Augusto, A., Di Ciccio, C.: Detection of statistically significant differences between process variants through declarative rules. In: BPM Forum, pp. 73–91 (2021)
8. Ceravolo, P., Tavares, G.M., Junior, S.B., Damiani, E.: Evaluation goals for online process mining: a concept drift perspective. In: SERVICES, pp. 27–27 (2022)
9. Kriglstein, S., Wallner, G., Rinderle-Ma, S.: A visualization approach for difference analysis of process models and instance traffic. In: BPM, pp. 219–226 (2013)
10. Küster, J.M., Gerth, C., Förster, A., Engels, G.: Detecting and resolving process model differences in the absence of a change log. In: BPM, pp. 244–260 (2008)
11. Lakshmanan, G.T., Rozsnyai, S., Wang, F.: Investigating clinical care pathways correlated with outcomes. In: BPM, pp. 323–338 (2013)
12. Leemans, S.J.J., Syring, A.F., van der Aalst, W.M.P.: Earth movers' stochastic conformance checking. In: BPM Forum, pp. 127–143 (2019)
13. de Leoni, M., Mannhardt, F.: Road traffic fine management process (2015)
14. Nguyen, H., Dumas, M., La Rosa, M., ter Hofstede, A.H.M.: Multi-perspective comparison of business process variants based on event logs. In: Conceptual Modeling, pp. 449–459 (2018)
15. Ostovar, A., Maaradji, A., La Rosa, M., ter Hofstede, A.H.M., van Dongen, B.F.V.: Detecting drift from event streams of unpredictable business processes. In: Conceptual Modeling, pp. 330–346 (2016)
16. Pini, A., Brown, R., Wynn, M.T.: Process visualization techniques for multi-perspective process comparisons. In: AP-BPM, pp. 183–197 (2015)
17. Poelmans, J., Dedene, G., Verheyden, G., Van der Mussele, H., Viaene, S., Peters, E.: Combining business process and data discovery techniques for analyzing and improving integrated care pathways. In: ICDM, pp. 505–517 (2010)
18. Suriadi, S., Mans, R.S., Wynn, M.T., Partington, A., Karnon, J.: Measuring patient flow variations: a cross-organisational process mining approach. In: AP-BPM, pp. 43–58 (2014)
19. Taymouri, F., La Rosa, M., Carmona, J.: Business process variant analysis based on mutual fingerprints of event logs. In: CAiSE, pp. 299–318 (2020)
20. Taymouri, F., Rosa, M.L., Dumas, M., Maggi, F.M.: Business process variant analysis: survey and classification. Knowl.-Based Syst. **211**, 106557 (2021)
21. Vidgof, M., Djurica, D., Bala, S., Mendling, J.: Interactive log-delta analysis using multi-range filtering. SoSyM **21**, 847–868 (2022)
22. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Ordered Sets, pp. 445–470 (1982)