Structural and Behavioral Biases in Process Comparison Using Models and Logs

Anna Kalenkova✉, Artem Polyvyanyy, and Marcello La Rosa

School of Computing and Information Systems
The University of Melbourne, VIC, 3010, Australia
{anna.kalenkova;artem.polyvyanyy;marcello.larosa}@unimelb.edu.au

Abstract. Process models automatically discovered from event logs represent business process behavior in a compact graphical way. To compare process variants, e.g., to explore how the system's behavior changes over time or between customer segments, analysts tend to visually compare conceptual process models discovered from different "slices" of the event log, solely relying on the structure of these models. However, the structural distance between two process models does not always reflect the behavioral distance between the underlying event logs and thus structural comparison should be applied with care. This paper aims to investigate relations between structural and behavioral process distances and explain when structural distance between two discovered process models can be used to assess the behavioral distance between the corresponding event logs.
Keywords: Process mining · Variant analysis · Structural distance · BPMN

1 Introduction

Process mining [1] is a branch science at the intersection of data science and process management, focused on the analysis of event logs extracted from enterprise systems. Key process mining capabilities include the *(1) automated discovery* of process models from event logs, *(2) performance mining* to identify friction points in process performance, *(3) conformance checking* to find discrepancies between actual and modeled process behavior, and *(4) variant analysis* to assess different variants of the same business process and identify root causes for their differences.

Automated process discovery algorithms generalize the behavior recorded in event logs by constructing conceptual process models that represent this behavior.

Conformance checking offers a variety of methods for *model-to-log* (M2L) comparison, when the process model corresponds to a prescribed process behavior and the log corresponds to the actual behavior of the process. Most of these approaches provide a single number to quantify the model and log similarity and are not supported by visual analytics, i.e., discrepancies of the processes are not explicitly visualized.

Variant analysis techniques [18], on the other hand, focus on *log-to-log* (L2L) and *model-to-model* (M2M) comparison. These techniques are used to compare variants of the same process, e.g., purchasing processes for different customer types. Specialized methods for L2L comparison were proposed in [4] and [5]. The approach [4] provides results in the form of natural language expressions, while technique [5] visualizes discrepancies on transition systems, that are either too abstract to represent real-world data or cannot be properly compared when detailed. Some of the conformance checking techniques [2,9,15,16] can be also used in variant analysis and applied for L2L comparison. However, none of the these techniques provide any visual information highlighting the discrepancies of event logs.

In contrast to the L2L comparison techniques, that lack visual analysis, state-of-the-art model-to-model (M2M) comparison methods are implemented in a variety of tools [8,10,11,13,19] and are primarily intended to visualize differences in conceptual process models. These methods explicitly highlight models' discrepancies.

With state-of-the-art M2M comparison tools and process discovery techniques, process analysts can discover models from event logs and then compare them applying one of the M2M methods, instantly visualizing differences in process variants. Fig. 1 presents a prospective schema for variant analysis when an event log L is split into two sub-logs: L_1 and L_2, then from these sub-logs process models M_1 and M_2 are discovered, and after that these models are matched using one of the M2M comparison techniques. Elements that should be deleted from M_1 and added to match it with M_2 are highlighted in red and green, respectively.

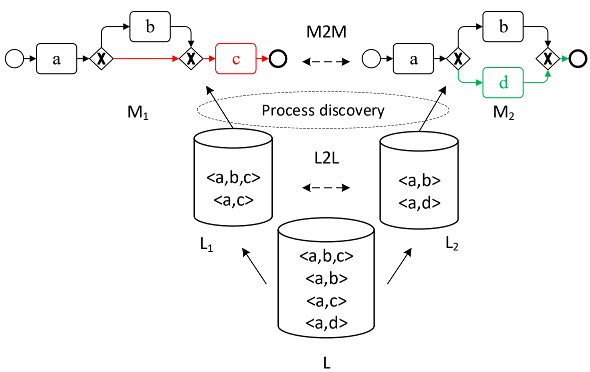


Fig. 1: Using M2M comparison in variant analysis.

Although this approach seems feasible and readily available, it raises the following research questions:

RQ 1: *Can we use structural M2M comparison as a proxy for the behavioral L2L comparison?* In other words, *do distances between process models correlate with the distances between the event logs?*

RQ 2: *Does the correlation between L2L and corresponding M2M distances depend on the algorithm used to discover the models from logs?*

RQ 3: *What is the role of representational biases, i.e., process modeling languages and distance metrics, in applying M2M comparison for the L2L analysis?*

To answer these questions we analyzed event logs of real-world systems. In order not to depend on an application domain, we spit the event logs into sub-logs (temporal slices) using time frames. This type of variant analysis is also known as *concept drift* analysis [6]. It assesses how the process changes over time. We discover process models from the sub-logs and then relate structural distances between these process models and behavioral distances between the corresponding sub-logs. We then answer the research questions and make conclusions and recommendations regarding applying the M2M techniques for the L2L analysis.

2 Motivating Example

In this section, we consider simple event logs and process models discovered from these event logs. We then relate the differences of the event logs to the differences of the corresponding process models. Consider event log $L_1 = \{\langle a, b, c, d, e\rangle, \langle a, d, e\rangle\}$. This event log contains two traces, each of the traces is a sequence of events. Fig. 2 presents a BPMN model M_1 discovered from L_1 by applying Split miner [3] or Inductive miner [12]. Note that for the event log L_1, these two discovery algorithms produce the same model M_1.

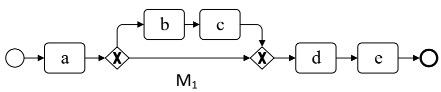


Fig. 2: M_1 discovered from L_1 by Split and Inductive miner algorithms.

Event log $L_2 = \{\langle a, b, c, d, e\rangle, \langle a, d, e\rangle\ \langle a, b, e\rangle\}$ includes the same set of traces as event log L_1 but also contains trace $\langle a, b, e\rangle$. For event log L_2 different process discovery algorithms produce different process models. Fig. 3 presents two process models M_1 and M_{2split} discovered by Split miner from event logs L_1 and L_2, respectively. To transform M_1 to M_{2split}, we remove two red arcs and add two green gateways and five green arcs. In this case, the model is modified by adding an alternative path.

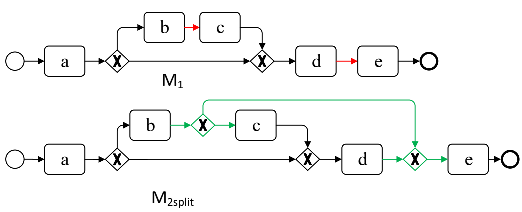


Fig. 3: M_1 and M_{2split} models discovered by Split miner from L_1 and L_2, respectively.

Two models M_1 and M_{2ind} discovered by Inductive miner from event logs L_1 and L_2 are presented in Fig. 4. The differences between these two models are also highlighted in red and green. In contrast to M_{2split}, model M_{2ind} significantly differs from M_1. One needs to remove four arcs, add four gateways and ten arcs to match model M_1[1]. That means, we need to reorganize the entire structure of the model. However, L_1 and L_2 differ in only one trace. The limitation of Inductive miner is that it cannot construct sequence flows going from one block of constructs to another, allowing only regular hierarchical structures of embedded sub-processes (sequence, choice,

[1] Although the model can be simplified (some gateways can be merged), we analyze BPMN models as they are provided by the discovery algorithms.

loop, parallel executions). This example demonstrates that different process discovery algorithms relate L2L and M2M distances differently.

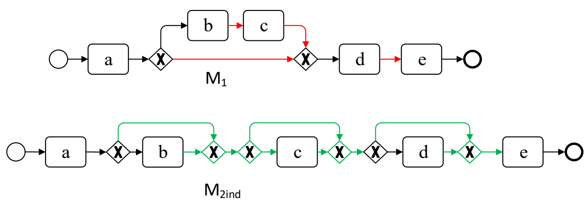


Fig. 4: M_1 and M_{2ind} discovered by Inductive miner from L_1 and L_2, respectively.

3 Structural and Behavioral Process Distances

In this section, we define process distances that are later used for the analysis of correlations between structural and behavioral characteristics of processes.

Each distance measure is considered together with a set of process models or event logs and forms a *metric space* $(\mathcal{M}, \Delta)$, where $\mathcal{M}$ is the set of models or event logs (represented structurally as graphs or behaviorally as languages, i.e., sets of execution sequences), and $\Delta : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is a distance function, such that:

1. *Identity of indiscernible*: $\forall M \in \mathcal{M} : \Delta(M, M) = 0$; $\forall M_1, M_2 \in \mathcal{M}$: if $M_1 \neq M_2$, then $\Delta(M_1, M_2) > 0$;
2. *Symmetry*: $\forall M_1, M_2 \in \mathcal{M} : \Delta(M_1, M_2) = \Delta(M_2, M_1)$;
3. *Triangle inequality*: $\forall M_1, M_2, M_3 \in \mathcal{M} : \Delta(M_1, M_2) \leq \Delta(M_1, M_3) + \Delta(M_3, M_2)$.

3.1 Structural process distance

The structural distance between two process models can be defined using the *graph edit distance* [7], i.e., the minimum number of atomic operations (insertions and deletions) that transform one process model to another. We will consider BPMN models containing only core elements (tasks, exclusive and parallel gateways, start and end events). The specifics of BPMN process models are that their nodes are typed, and tasks are labeled. Thus, we only match nodes if they are of the same type and their labels coincide.

Once the minimum number of insertion and deletion operations is defined, we calculate the structural distance between two process models M_1 and M_2 as:

$$\Delta_{struct}(M_1, M_2) = 1 - \frac{sim(M_1, M_2)}{sim(M_1, M_2) + \mathit{diff}(M_1, M_2)}, \quad (1)$$

where $sim(M_1, M_2)$ is the number of matching elements and $\mathit{diff}(M_1, M_2)$ is the number of mismatching elements in M_1 and M_2. The structural distance is 0 for perfectly matching models and 1 for completely different models. The higher the distance value, the more models differ.

Recall the models from the motivating example (Section 2) that were discovered by Split (Fig. 3) and Inductive (Fig. 4) miners. The structural distance between the models discovered by Split miner $\Delta_{struct}(M_1, M_{2split}) = 0.360$ is less than the distance between the models discovered by Inductive miner $\Delta_{struct}(M_1, M_{2ind}) = 0.563$.

Besides quantifying the structural distance between process models, the proposed distance measure also defines a metric space, i.e., the three properties are satisfied.

3.2 Behavioral process distances

The behavior of a process model, as well as an event log, is represented by a language, i.e., a sets of label sequences, it encodes. The distance between two languages L_1 and L_2 can be calculated as:

$$\Delta_{beh}(L_1, L_2) = 1 - \frac{ent\bullet(L_1 \cap L_2)}{ent\bullet(L_1 \cup L_2)}, \qquad (2)$$

where $ent\bullet$ is the *short-circuit* entropy measure that estimates the language cardinality (the number of sequences the language contains) [16]. This behavioral distance measure gives a number between 0 and 1, which is 0 for identical languages, when $L_1 \cap L_2 = L_1 \cup L_2$, and 1, when the languages have the empty intersection.

For event logs L_1 and L_2 from the motivating example (Section 2), the behavioral distance is estimated as: $\Delta_{beh}(L_1, L_2) = 0.416$. The behavioral distances between the corresponding process models discovered by Split miner and Inductive miner are $\Delta_{beh}(L_{M_1}, L_{M_{2split}}) = 0.416$ and $\Delta_{beh}(L_{M_1}, L_{M_{2ind}}) = 0.708$, respectively, where $L_{M_1}, L_{M_{2split}}, L_{M_{2ind}}$ are the languages accepted by these models. Similar to the structural distance, this behavioral distance forms a metric space.

The behavioral distance measure, see eq. (2), is restrictive and assesses only the share of common sequences of two languages. To also consider common subsequences we use a so-called partial behavioral distance that is calculated for the "diluted" languages (for the details see [15]) that extend the initial languages by allowing any number of label skips. Consider language $L = \{\langle a, b, c\rangle, \langle a, d\rangle\}$. The corresponding "diluted" language $L' = \{\langle a, b, c\rangle, \langle a, b\rangle, \langle a, c\rangle, \langle b, c\rangle, \langle a, d\rangle, \langle a\rangle, \langle b\rangle, \langle c\rangle, \langle d\rangle, \langle\rangle\}$ contains all the subsequences of L that can be obtained by skipping some labels in the words of L. The partial behavioral distance between languages L_1 and L_2 is defined as:

$$\Delta'_{beh}(L_1, L_2) = 1 - \frac{ent\bullet(L'_1 \cap L'_2)}{ent\bullet(L'_1 \cup L'_2)}, \qquad (3)$$

where L'_1 and L'_2 are the "diluted" versions of languages L_1 and L_2, respectively. This measure assesses the share of common subsequences in two languages.

Although the behavioral distance, see eq. (2), forms a metric space, the partial behavioral distance, see eq. (3), does not satisfy the *Identity of indiscernible* rule. Consider two languages: $L_1 = \{\langle a, b\rangle\}$ and $L_2 = \{\langle a, b\rangle, \langle a\rangle\}$. Note that $L_1 \neq L_2$, and at the same time, $L'_1 = L'_2 = \{\langle a, b\rangle, \langle a\rangle, \langle b\rangle, \langle\rangle\}$, and hence, $\Delta'_{beh}(L_1, L_2) = 0$. However, the partial behavioral distance satisfies the other space metric properties, i.e., *Symmetry* and *Triangle inequality*.

4 Evaluation

This section analyzes real-world event logs by relating their behavioral distances to the structural distances between BPMN models discovered from these event logs. The distances between process models were calculated with BPMNDiffViz tool[2] [8] – an

[2] https://bitbucket.org/sivanov68/bpmndiffviz/src/master/.

open-source tool for structural BPMN model comparison. To overcome the computational costs, we applied the greedy approximation algorithm that, according to [17], gives the minimal edit distance in most of the cases for BPMN models discovered from event logs. Behavioral distances were estimated using Entropia[3] [14] – an open-source and publicly available tool for measuring the quality of discovered process models.

We have analyzed seven real-world publicly available BPI (Business Process Intelligence) Challenge event logs[4]. Table 1 presents characteristics of these event logs, including the numbers of traces, events, and their occurrences. Note that one trace can occur multiple times in a real-life event log. The table also contains event logs' notations that will be used later to denote the corresponding distances.

Event log	Name	Notation	# Traces	# Trace Occur.	# Events	# Ev. Occur.
1	Domestic Declarations'20	●	99	10,500	17	56,437
2	International Declarations'20	●	753	6,449	34	72,151
3	Prepaid Travel Cost'20	●	202	2,099	29	18,246
4	Travel Permit Data'20	●	1,478	7,065	51	86,581
5	Request For Payment'20	●	89	6,886	19	36,796
6	Application Receipt Phase	●	116	1,434	27	8,557
7	Road Traffic	●	231	150,370	11	561,470

Table 1: Characteristics of the real-world event logs.

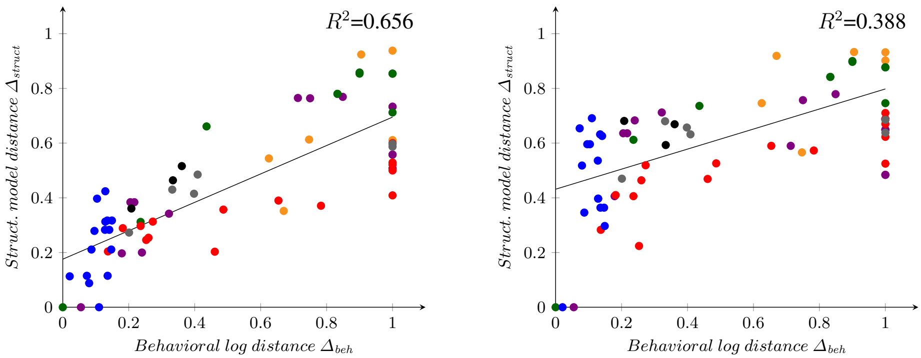
To not depend on a particular application domain, we have split each of the event logs into six sub-logs that correspond to six equal time frames. Each sub-log includes all the traces that contain events belonging to the corresponding time frame.

We then applied two process discovery algorithms, namely Split miner [3] and Inductive miner [12], to construct BPMN models from the given 42 sub-logs. Eight models produced by Split miner were not sound, i.e., could not be described as finite automata models, so their behavior could not be quantified, and hence, the corresponding sub-logs were excluded from the evaluation.

All possible pairs of the sub-logs of the same event log and corresponding discovered models were compared. Fig. 5 relates the behavioral sub-log distances to the structural BPMN model distances. As observed from the two plots for the models discovered by Split miner (Fig. 5a) and Inductive miner (Fig. 5b), the distances between the sub-logs are more correlated with the structural distances between the corresponding models discovered by Split miner.

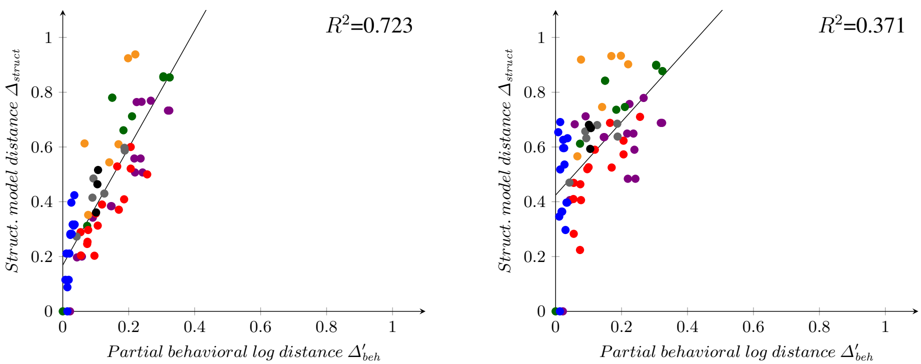
While the structural distances between the models discovered by Split miner (Fig. 6a) show good correlation with the partial behavioral distances between the corresponding sub-logs ($R^2 = 0.723$), for the models discovered by Inductive miner (Fig. 6b) the correlation is less obvious ($R^2 = 0.371$). However, in both cases the M2M comparison only approximates L2L distances. Especially this can be seen for the sub-logs of the *Road Traffic* event log, i.e., behavioral distances between similar sub-logs cannot

[3] https://github.com/jbpt/codebase/tree/master/jbpt-pm/entropia.
[4] https://data.4tu.nl/



(a) Models discovered by Split miner. (b) Models discovered by Inductive miner.

Fig. 5: Behavioral distances between the sub-logs and structural distances between the discovered models.



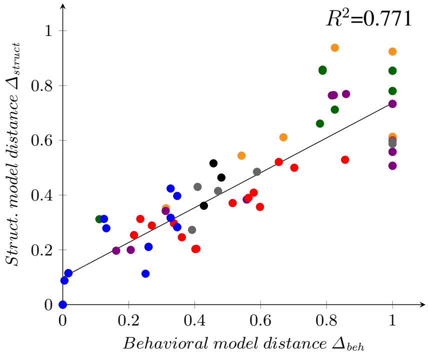
(a) Models discovered by Split miner. (b) Models discovered by Inductive miner.

Fig. 6: Partial behavioral distances between the sub-logs and structural distances between the corresponding models.

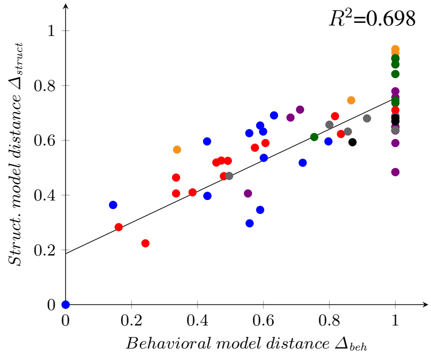
be differentiated using the M2M comparison. At the same time, M2M analysis allows distinguishing the cases when event logs are similar or distinct.

Split miner and Inductive miner produce different types of BPMN models: while models discovered by Inductive miner are hierarchical, models constructed by Split miner represent a wider class of arbitrarily structured BPMN diagrams. To draw more detailed conclusions, we need to understand the role of representational bias.

Fig. 7 and Fig. 8 relate distances and partial distances between languages of the analyzed process models to the structural distances between these models. Structural distances between the process models discovered by Split miner (Fig. 7a, Fig. 8a) correlate with the corresponding model behavioral distances, similarly to how they correlate with the distances between the corresponding sub-logs (similar R^2 values). However, structural distances between the models discovered by Inductive miner (Fig. 7b, Fig. 8b) reflect differences in the behavior of these models better than in the behavior of the corresponding sub-logs. In the next section, we discuss and explain this phenomenon.

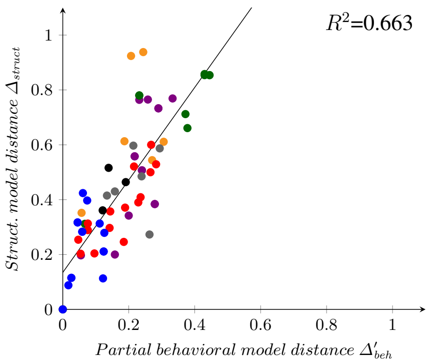


(a) Models discovered by Split miner.

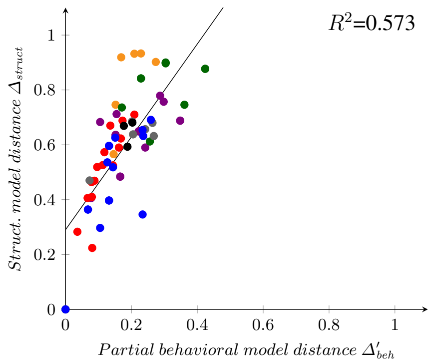


(b) Models discovered by Inductive Miner.

Fig. 7: Behavioral and structural distances between the discovered models.



(a) Models discovered by Split Miner.



(b) Models discovered by Inductive miner.

Fig. 8: Partial behavioral and structural distances between the discovered models.

5 Discussion

Although structural distances between process models produced by Inductive miner correlate with the behavioral model distances, there is less correlation with the distances of corresponding event logs. Which means we need to have a closer look at the relations between behavioral M2M distances and corresponding L2L distances.

Fig. 9 shows a metric space $(\mathcal{M}, \Delta)$ for languages that represent behaviors of process models and event logs. Consider models M_1 and M_2 discovered from event logs L_1 and L_2, and accepting languages L_{M_1} and L_{M_2}. According to the *Triangle inequality*, $\Delta(L_{M_1}, L_{M_2}) \leq \Delta(L_1, L_2) + \Delta(L_{M_1}, L_1) + \Delta(L_{M_2}, L_2)$ and $\Delta(L_{M_1}, L_{M_2}) \geq \Delta(L_1, L_2) - \Delta(L_{M_1}, L_1) - \Delta(L_{M_2}, L_2)$, i.e., when approximating $\Delta(L_1, L_2)$, distance $\Delta(L_{M_1}, L_{M_2})$, in the worst case, includes $\Delta(L_{M_1}, L_1)$ and $\Delta(L_{M_2}, L_2)$.

The aim of process discovery algorithms is to minimize $\Delta(L_{M_i}, L_i)$, $i \in \{1, 2\}$, distances and construct models that are behaviorally close to the event logs. An ideal discovery algorithm would construct models, such that $\Delta(L_{M_i}, L_i) = 0$, and hence $\Delta(L_{M_1}, L_{M_2}) = \Delta(L_1, L_2)$. However, this is not always possible and, first of all, because of the representational bias. Even if a discovery algorithm makes the best possible attempt to construct a model M that is behaviorally similar to the event log L, the distance $\Delta(L_M, L)$ is bounded by $\Delta_{min}(\mathcal{M}, \Delta)$, i.e., $\Delta(L_M, L) \geq \Delta_{min}(\mathcal{M}, \Delta)$, where $\Delta_{min}(\mathcal{M}, \Delta)$ is the minimal distance between behaviors in the metric space.

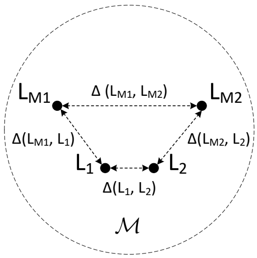


Fig. 9: Behavioral distances between models and logs.

The metric space of block structured BPMN models discovered by Inductive miner is sparser than the space of BPMN models with arbitrary structures constructed by Split miner because it forms a proper subset of the models discovered by Split miner, hence, the value of $\Delta_{min}(\mathcal{M}, \Delta)$ for Split miner is less than for Inductive miner, so, it is easier for Split miner to find a model that is behaviorally closer to a given event log. However, Split miner can produce unsound models that cannot be represented as finite automata models and their behavior cannot be quantified. Apart from the structuredness, there can be other constraints on the sets of models that influence the density of the metric space. For instance, both Split miner and Inductive miner construct process models with uniquely labeled tasks.

These observations provide answers to the research questions: RQ 1. In some cases, M2M comparison can be used as a proxy for L2L comparison. However, M2M analysis only approximates L2L distances; RQ 2. The discovery algorithm plays the pivotal role in relating M2M and L2L distances, because of its two main characteristics: the representational bias and the ability to construct models that behaviorally close to the event logs (this characteristic depends on the representational bias, but also incorporates the quality of the algorithm itself); RQ 3. The role of representational bias is crucial, and together with different metrics it affects correlations between M2M and L2L distances.

These results explain when the M2M comparison can be used for the L2L analysis. Namely, the analyst should consider two characteristics of the discovery algorithm that affect correlations between M2M and L2L distances: (1) *accuracy*, i.e., the distance between an event log and the discovered process model; (2) *representational bias*, i.e., the process modeling language. These results provide a tool for analysing process discovery algorithms, i.e., to assess applicability of a new process discovery algorithm for the comparison of event logs using M2M analysis, one should consider its accuracy and representational bias and then relate to those of other discovery algorithms.

6 Conclusions and Future Work

This paper bridges the gap between behavioral and structural comparison techniques in process mining. It explains the cases when behavioral differences in processes can be learned from the structural differences in the corresponding process models. The results are supported by experiments on real-life event logs. In future work, we plan to conduct large scale experiments and consider further process discovery algorithms and behavioral distance measures. We also plan to run an empirical study that will involve participants assessing structural and behavioral distances between models and logs.

Acknowledgments. This work was partly supported by the Australian Research Council Discovery Project DP180102839.

References

1. van der Aalst, W.: Process Mining: Data Science in Action. Springer (2016)
2. Augusto, A., Armas-Cervantes, A., Conforti, R., Dumas, M., La Rosa, M., Reissner, D.: Abstract-and-compare: A family of scalable precision measures for automated process discovery. In: Business Process Management. pp. 158–175. Springer, Cham (2018)
3. Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Polyvyanyy, A.: Split miner: Automated discovery of accurate and simple business process models from event logs. Knowl. Inf. Syst. 59(2), 251–284 (2019)
4. van Beest, N., Dumas, M., García-Bañuelos, L., La Rosa, M.: Log delta analysis: Interpretable differencing of business process event logs. In: BPM. pp. 386–405. Springer (2015)
5. Bolt, A., de Leoni, M., van der Aalst, W.: A visual approach to spot statistically-significant differences in event logs based on process metrics. In: Advanced Information Systems Engineering. pp. 151–166. Springer International Publishing, Cham (2016)
6. Bose, R., van der Aalst, W., Žliobaitė, I., Pechenizkiy, M.: Dealing with concept drifts in process mining. IEEE Trans. on Neural Networks and Learning Syst. 25(1), 154–171 (2014)
7. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Anal. Appl. 13(1), 113–129 (Feb 2010)
8. Ivanov, S., Kalenkova, A., van der Aalst, W.: BPMNDiffViz: A tool for BPMN models comparison. In: BPM Demo Session, BPM. CEUR-WS, vol. 1418, pp. 35–39 (2015)
9. Kalenkova, A., Polyvyanyy, A.: A spectrum of entropy-based precision and recall measurements between partially matching designed and observed processes. In: Service-Oriented Computing. pp. 337–354. Springer International Publishing, Cham (2020)
10. La Rosa, M., Dumas, M., Uba, R., Dijkman, R.: Business process model merging: An approach to business process consolidation. ACM Trans. Softw. Eng. Methodol. 22(2) (2013)
11. La Rosa, M., Reijers, H., van der Aalst, W., Dijkman, R., Mendling, J., Dumas, M., García-Bañuelos, L.: Apromore: An advanced process model repository. Expert Systems with Applications 38(6), 7029–7040 (2011)
12. Leemans, S., Fahland, D., van der Aalst, W.: Discovering Block-Structured Process Models from Incomplete Event Logs. In: ATPN'2014, LNCS, vol. 8489, pp. 91–110. Springer (2014)
13. Pietsch, P., Wenzel, S.: Comparison of BPMN2 diagrams. In: Business Process Model and Notation. pp. 83–97. Springer, Berlin, Heidelberg (2012)
14. Polyvyanyy, A., Alkhammash, H., Di Ciccio, C., García-Bañuelos, L., Kalenkova, A., Leemans, S., Mendling, J., Moffat, A., Weidlich, M.: Entropia: A family of entropy-based conformance checking measures for process mining. In: ICPM Doctoral Consortium and Tool Demonstration Track. CEUR Workshop Proceedings, vol. 2703, pp. 39–42 (2020)
15. Polyvyanyy, A., Kalenkova, A.A.: Monotone conformance checking for partially matching designed and observed processes. In: ICPM. pp. 81–88. IEEE (2019)
16. Polyvyanyy, A., Solti, A., Weidlich, M., Di Ciccio, C., Mendling, J.: Monotone precision and recall measures for comparing executions and specifications of dynamic systems. ACM Trans. Softw. Eng. Methodol. 29(3) (2020)
17. Skobtsov, A., Kalenkova, A.: Efficient algorithms for finding differences between process models. In: 2019 Ivannikov Ispras Open Conference (ISPRAS). pp. 60–66 (2019)
18. Taymouri, F., La Rosa, M., Dumas, M., Maggi, F.: Business process variant analysis: Survey and classification. Knowledge-Based Systems 211, 106557 (2021)
19. Treude, C., Berlik, S., Wenzel, S., Kelter, U.: Difference computation of large models. In: ESEC/FSE. p. 295–304. ACM, NY (2007)