Alignment Approximator: A ProM Plug-In to Approximate Conformance Statistics

Mohammadreza Fani Sani[1,2,*], Juan José Garza González[1], Sebastiaan J. van Zelst[3,1] and Wil M.P. van der Aalst[1,3]

[1] Process and Data Science Chair, RWTH Aachen University, Aachen, Germany
[2] Microsoft Development Center Copenhagen, Copilot Applied AI Team, Copenhagen, Denmark
[3] Fraunhofer FIT, Birlinghoven Castle, Sankt Augustin, Germany

Abstract

Conformance checking techniques compare process models with real execution data to assess their alignment. Alignments are valuable for calculating conformance statistics, but exact solutions can be computationally expensive for large event data sets. This paper presents an easy-to-use plug-in for the ProM process mining framework that approximates alignment values. The plug-in takes an event log and process model as input, providing an approximate alignment value along with bounds for the actual alignment. Diagnostic information on problematic activities is also provided. Three approaches are offered: subset selection, simulation, and log-to-log comparison. This plug-in enables efficient conformance assessment, overcoming computational challenges for large event data sets.

Keywords

Process Mining, Conformance Checking Approximation, Alignment, Subset Selection, Edit Distance, Simulation

1. Introduction

Conformance checking, a fundamental aspect of process mining, focuses on assessing the adherence of a designed or discovered process model to real process executions [1]. These techniques are valuable for identifying deviations and measuring the accuracy of process models in representing recorded event data. To handle concurrency and capture order-independent activity execution, conformance checking techniques rely on process modeling formalisms. Early approaches like "token-based replay" [2] often produced ambiguous or unpredictable results. Consequently, alignments were introduced to offer clearer explanations and quantification of deviations [3]. Alignments have rapidly become the standard conformance checking technique in practice [4]. However, computing alignments can be time-consuming, especially for complex process models and real-life datasets, making it challenging to apply them in practical settings using standard hardware.

In numerous applications, the computation of alignment values is required multiple times. For example, when seeking an appropriate process model for an event log, various process discovery algorithms with different settings are employed to discover multiple process models. The alignment techniques are then used to assess the fit between each process model and the event log. However, traditional alignment methods tend to be time-consuming, especially when dealing with large event data sets. This limitation makes it impractical to analyze numerous candidate process models within a limited timeframe. Therefore, by reducing the computation time of alignments, a greater number of candidate models can be considered for evaluation. Additionally, in many cases, obtaining precise alignment values is unnecessary, and having a quick approximation or a close lower/upper bound would suffice.

Proceedings of the Demonstration & Resources Track, Best BPM Dissertation Award, and Doctoral Consortium co-located with the 21st International Conference on Business Process Management, BPM 2023, Utrecht, The Netherlands, September 11-15, 2023
fanisani@pads.rwth-aachen.de (M. Fani Sani); j.garzagonzalez@celonis.com (J. J. Garza González); s.j.v.zelst@pads.rwth-aachen.de (S. J. van Zelst); wvdaalst@pads.rwth-aachen.de (W. M.P. van der Aalst)

Recently, various approaches for alignment approximation have been proposed [5, 6]. In our previous work [6], we exploit *subsets of the process model's behavior* for approximation, i.e., by using the subset of process behavior as a representative for the complete process model behavior. In this way, we are able to provide bounds for the approximated alignment value. Moreover, in [7], we show that it is possible to use simulation methods to generate the subset of model behavior. This approach lets us compute alignments (and also their approximation) for any process model independent of their notation. Therefore, by having some behaviors that are executable in the process model (e.g., using simulation), we are able to approximate its alignment value.

This demo paper presents an easy-to-use Alignment Approximator plug-in that utilizes three current approaches to approximate alignments, ensuring flexibility and accuracy. By offering bounds for the actual alignment, the plug-in provides users with valuable insights into the reliability of the approximated value. Additionally, users have the option to consider an event log as some possible behavior of the process model that can be achieved by simulation, enabling alignment computations for process models represented in various notations, including Petri nets. It should be noted that it is not required that the event log that represents the model contains all model's behavior as we approximate the alignment cost.

The proposed plug-in encompasses three distinct approaches for approximating alignments:

- Subset Selection: This approach allows users to select a subset of model behavior based on customizable parameters, effectively approximating the alignment value with by computing alignment of limited number of variants in the event log.
- Simulation: Leveraging process model simulation, this approach estimates the alignment value by configuring simulation parameters to achieve optimal results. This method does not need to compute any alignment.
- Log-to-Log Comparison: By comparing the behavior captured in two event logs, this approach approximates the alignment value. It is particularly useful when evaluating alignment without the requirement of discovering a process model, and when reliable variants of traces are available.

These three approaches enable users to approximate alignment values efficiently while also providing diagnostic information.

The remainder of this paper is structured as follows. In Section 2, we discuss the approximation methods at an abstract level. Moreover, Section 3 explains how to use the developed tool. Furthermore, Section 4 briefly describes the maturity of Alignment Approximator. Finally, Section 5 concludes the paper presents some directions for extending the implementation.

2. Alignment Approximation Using a Subset of Model Behaviors

The general idea of the used sampling method is presented in Fig. 1. The general idea is instead of using the whole process model that is a set of sequences over a set of activities $\mathcal{A}$ (i.e., $\{\}\subseteq\mathcal{M}\subseteq\mathcal{A}^*$), we proposed to use a subset of process model (i.e., $\{\}\subseteq\mathcal{M}'\subseteq\mathcal{M}$). As it shows in [8] the edit distance function quantifies alignment costs. The edit distance function $\triangle : \mathcal{A}^*\times\mathcal{A}^*\rightarrow\mathbb{N}_{\geq 0}$ receives two sequences and returns the number of edits (i.e., inserts and deletes) to convert one trace to the other one. It is shown in [7] that by using the edit distance function, we are also able to detect some deviated behavior that is used to provide some diagnostic information. Considering this notation, we can approximate the alignment cost of process models with any notation as far as we have some of the traces that are executable by the process model, e.g., by simulation.

To generate the process model subset, we consider three approaches. In *subset selection*, we can select some of the variants in the event log and compute their alignments using the classical alignment method [3]. We have different options to choose variants, like considering their *frequency*, *length*,

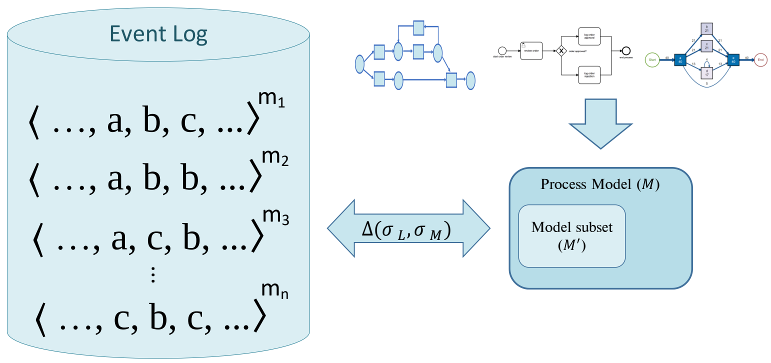

Figure 1: The schematic view of the proposed conformance approximation methods. We consider a process model as a set of sequence of activities. Therefore, we are able to consider all many of the available process model notations as far as having some of the executable traces of that model. In this way, even an event log can be considered as a process model.

and *similarity* to other variants and selecting them *randomly*. The drawback of this method is we should describe the given process model with the Petri net notation. This approach and how we can compute bounds for the actual alignment cost are presented in [6, 8].

The second approach to generate the subset of model traces is *simulation*. This simulation could be done *randomly* or be *guided* by the behavior and their probabilities in the event log [7]. Moreover, the number of simulated by the user can be set by the user. In addition, the user can decide if he wants to remove repetitive patterns in alignment approximation or not. Note that the proposed guided simulation method currently works for process models presented with the Petri net notation.

In the last approach, i.e., *log-to-log comparison*, we consider a process model as an event log that is a multiset over a set of sequences of activities. In other words, let $\mathcal{A}$ denotes a set of activities, we define an event log as $L \in \mathcal{B}(\mathcal{A}^*)$. In this regard, we first find the set of unique variants in the event log, i.e., $\overline{L}$. As in this approach, we directly use some possible traces of the process model, even if there is no reference process model and just some of the correct behaviors of the process (e.g., some of the valid variants) are known, the proposed method is able to approximate the conformance value. Using this approach, we are able to compute the alignment of process models with any notations until there is a subset of their simulation. It is also possible to give the process model behaviors as a CSV file and later convert it to an event log.

Using all three approaches, the plug-in returns an approximation for the alignment cost and upper and lower bounds for the actual alignment. Note, it returns these fitness values that are directly computed based on the alignment costs. Moreover, the plug-in provides some diagnostic information about the problematic activities and how many times each activity has sync/async moves.

3. Alignment Approximator Tool

We have developed the Alignment Aproximator tool as a plug-in in the ProM [9] framework to increase its integration with other process mining plug-ins. The ProM framework is one of the most widely used open-source process mining platforms with several process mining algorithms. Providing our tool in this framework lets users easily apply it among other process mining methods. This tool is accessible via https://svn.win.tue.nl/repos/prom/Packages/LogFiltering/Trunk/. For example, it is possible to simulate a BPMN model and use the simulated log as an input of the Alignment Aproximator tool.

As mentioned earlier, there are three methods available for generating the subset of a process model. These methods involve: 1) aligning a process model with selected variants from the event log (subset selection), 2) simulating the behavior of the process model (simulation), and 3) considering a

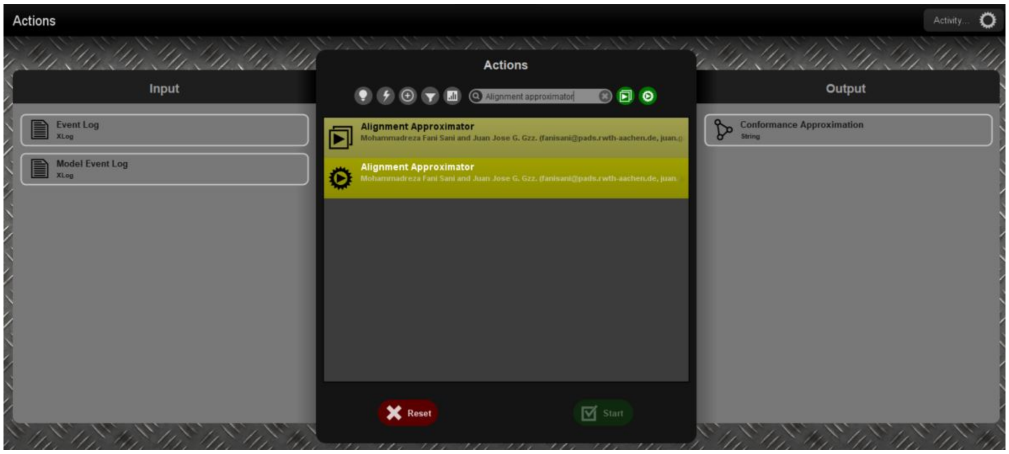

Figure 2: The inputs and outputs of the Alignment Aproximator tool. The tool can be used in two ways: by providing one event log and a Petri net, or by providing two event logs.

Upper Bound:	0.983425882102646
Approximation:	0.9828658747757373
Lower Bound:	0.9823058674488285

(a) Alignment cost approximation

Activities	Synchronous Moves	Asynchronous Moves	Synchronous Ratio	Asynchronous Ratio
Payment	69630	7971	89.73	10.27
Insert Date Appeal to Prefecture	121	4067	2.89	97.11
Receive Result Appeal from Prefecture	121	878	12.11	87.89
Notify Result Appeal to Offender	121	775	13.50	86.50
Send Appeal to Prefecture	121	4020	2.92	97.08
Appeal to Judge	121	434	21.80	78.20
Add penalty	79217	643	99.19	0.81
Insert Fine Notification	79739	121	99.85	0.15
Send for Credit Collection	59013	0	100.00	0.00
Create Fine	150370	0	100.00	0.00
Send Fine	103504	604	99.42	0.58

(b) Diagnostic information

Figure 3: An example of the information provided by the Alignment Approximator tool. We can have the approximated alignment value and the upper and lower bound for it. Moreover we can use diagnostic information to find out which activities are more problematic.

process model as a set of activity sequences (log-to-log comparison).

The provided plug-in receives two inputs that are event log and process model. Users can give a (simulated) event log or a Petri net to have a process model. A snapshot of this plug-in and its inputs are presented in Fig. 2 If the user provides a Petri net for the input process model, the plug-in provides two possibilities to the user to approximate the alignment value, i.e., simulation and subset selection. If the user prefers the subset selection method, he/she can select different selection strategies, e.g., clustering, frequency, and similarity. By selecting the simulation approach, users can adjust the simulation by different methods, e.g., the number of simulated traces and the type of summarization. For more information about how to adjust the settings, please refer to [10]. Furthermore, if the user gives an event log as a process model, there would not be any further option, and the result will be shown to the user.

The output of this plug-in is an approximation of the alignment, i.e., the approximated fitness, upper and lower bounds for the actual fitness, and the number of synchronous and asynchronous moves for different activities. Two snapshots of the output of the Alignment Approximator tool are presented in Fig. 3a and Fig. 3b.

A video that describes how to use this tool is presented in https://youtu.be/eJBBuNhFmC4. Moreover, in https://github.com/fanisanim/AlignmentApproximator, we provide a comprehensive guide on how to effectively utilize the Alignment Approximator tool.

4. Maturity of the tool

The proposed tool integrates multiple approximation algorithms, making it user-friendly and accessible to end-users. These algorithms have been extensively applied to approximate event logs from diverse real-world datasets, showcasing their ability to enhance the performance of alignment computations [7, 6, 8]. Furthermore, we have conducted tests using real event logs to evaluate the tool's new capabilities, such as log-to-log comparison, and have observed significant improvements in alignment computation efficiency. These findings reinforce the tool's maturity and its potential to deliver enhanced performance for alignment approximation.

5. Conclusion

This demo paper presents a ProM plug-in that approximates alignments and provides bounds for alignment costs, along with diagnostic information about synchronous and asynchronous moves in activities. Developed within the ProM platform, it offers three different methods for alignment approximation: subset selection, simulation, and log-to-log comparison. The log-to-log comparison approach extends the tool's applicability to process models with different notations by enabling simulation. Additionally, we provide a video and a tutorial that offer step-by-step instructions on how to use the tool effectively. To advance this research, we aim to provide a method that offers acceptable approximation error bounds and adjusts method settings based on event log characteristics. Moreover, we want to show the diagnostic information of the alignment on process models that helps analysts detect the process's problematic part.

Acknowledgements

Funded under the Excellence Strategy of the Federal Government and the Länder. We also thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

[1] W. M. P. van der Aalst, Process Mining - Data Science in Action, Second Edition, Springer Berlin Heidelberg, 2016. doi:10.1007/978-3-662-49851-4.

[2] A. Rozinat, W. M. P. van der Aalst, Conformance checking of processes based on monitoring real behavior, Information Systems 33 (2008) 64–95.

[3] A. Adriansyah, J. Munoz-Gama, J. Carmona, B. van Dongen, W. M. P. van der Aalst, Alignment based Precision Checking, in: International Conference on Business Process Management, Springer, 2012, pp. 137–149.

[4] S. J. van Zelst, A. Bolt, M. Hassani, B. F. van Dongen, W. M. P. van der Aalst, Online conformance checking: relating event streams to process models using prefix-alignments, International Journal of Data Science and Analytics (2017) 1–16.

[5] M. Bauer, H. van der Aa, M. Weidlich, Sampling and approximation techniques for efficient process conformance checking, Information Systems (2020) 101666.

[6] M. Fani Sani, S. J. van Zelst, W. M. P. van der Aalst, Conformance checking approximation using subset selection and edit distance, in: International Conference on Advanced Information Systems Engineering, Springer, 2020, pp. 234–251.

[7] M. Fani Sani, J. J. G. Gonzalez, S. J. van Zelst, W. M. P. van der Aalst, Conformance checking approximation using simulation, in: 2020 2nd International Conference on Process Mining (ICPM), IEEE, 2020, pp. 105–112.

[8] M. Fani Sani, M. Kabierski, S. J. van Zelst, W. M. P. van der Aalst, Model independent error bound estimation for conformance checking approximation, arXiv e-prints (2021) arXiv–2103.

[9] W. M. P. van der Aalst, B. van Dongen, C. W. Günther, A. Rozinat, E. Verbeek, T. Weijters, Prom: The process mining toolkit, BPM (Demos) 489 (2009).

[10] J. J. G. González, Conformance checking approximation through a process model simulation, in: Master thesis, RWTH-Aachen University, 2020. doi:10.13140/RG.2.2.23813.06887.