

RESEARCH ARTICLE

Quality Evaluation of Clustering-Based Process Model Discovery Using Event Log Similarity



Guodan Yin¹ , Huiling Li² , Cong Liu^{3,*}, Qinghua Zhang⁴ and Haijun Zhang⁵

¹*School of Culture and Communication, Zibo Polytechnic University, China*

²*School of Computer Science and Technology, Shandong University of Technology, China*

³*NOVA Information Management School, Nova University of Lisbon, Portugal*

⁴*Zibo Municipal Government Agencies Hospital, China*

⁵*Jinan Wave Data Technology Co., Ltd., China*

Abstract: Model discovery discovers process models from event logs, but directly discovered models are usually complex and hard to interpret. Trace clustering tackles this problem by splitting an event log into smaller sublogs, each of which can support the discovery of simpler models. In this setting, clustering quality means that the obtained sublogs are behaviorally coherent and sufficiently different to achieve meaningful process model discovery. Existing evaluation depends on *F-Measure*, but it is dependent on the chosen discovery algorithm, requires costly fitness and precision computation, etc. To overcome these disadvantages, in this paper, we propose a clustering evaluation approach as a surrogate complement to model-quality evaluation rather than an alternative to existing model-quality measures such as *fitness*, *precision*, and *F-Measure*. Specifically, this clustering evaluation approach assesses clustering quality on the inter-sublog event log similarity, rather than the discovered process models. These are instantiated with three state-of-the-art log-similarity metrics. Our experiments using four real-life event logs show that the obtained scores are strongly negatively correlated with the *F-Measure* in most of the cases and have reduced the evaluation time by a factor of 47,547 down to 63 ms. These results indicate that the proposed framework provides an efficient and discovery-relevant alternative for clustering evaluation in process model discovery.

Keywords: process mining, trace clustering, process model discovery, log similarity, quality evaluation

1. Introduction

Process mining [1–3] aims at discovering, monitoring, and improving real-world business processes by mining knowledge from event logs. An event log can consist of multiple traces, and each trace is seen as an ordered sequence of activities. An important task in process mining is model discovery, and model discovery algorithms have been extensively studied, such as Alpha Miner, Heuristic Miner, OER-Miner, and Inductive Miner. However, real-life processes are usually flexible and weakly structured, and directly applying a discovery algorithm may produce a so-called spaghetti model [4] in process mining, which is too complex and difficult to understand. To address this limitation, trace clustering decomposes an event log into several sublogs, so that each sublog can produce a simpler and more interpretable process model. In recent years, a wide range of approaches has

been developed for this purpose. The general idea is that a trace similarity measure is defined and the original log is partitioned so that traces within the same sublog are similar, whereas traces across different sublogs are as dissimilar as possible. This yields two benefits: (i) process models derived from sublogs can represent the behavior captured in the original log with greater clarity and fidelity; and (ii) each submodel remains easy to understand, thereby mitigating the spaghetti model problem.

A key challenge, however, is how to evaluate clustering quality for process discovery. In this context, clustering quality does not merely refer to generic cluster compactness or separation but to whether the resulting sublogs are behaviorally coherent and sufficiently distinct to support meaningful and interpretable process model discovery. At present, there is still a lack of an efficient and algorithm-independent method for this purpose.

Although numerous clustering quality metrics have been developed in data mining, many of them require domain information that is typically unavailable in real-life event logs, which limits their applicability to process discovery [5–7]. As a result,

*Corresponding author: Cong Liu, NOVA Information Management School, Nova University of Lisbon, Portugal. Email: cliu@novaims.unl.pt

the *F-Measure* has been adopted to evaluate clustering for model discovery [8, 9]. In practice, computing the *F-Measure* requires discovering a submodel for each sublog and calculating fitness and precision [10, 11] for all sublog-submodel pairs. This practice reveals a clear research gap. First, the evaluation results depend strongly on the chosen discovery algorithm because different algorithms may produce different models and thus different fitness and precision values. Second, the required conformance checking is computationally expensive, especially when many sublogs or clustering candidates must be evaluated.

This paper is based on the central hypothesis that high-quality trace clustering produces sublogs that are behaviorally distinct, and that such distinctiveness can be quantified through event log similarity. If this hypothesis holds, clustering quality can be assessed directly at the log level, without relying on discovered process models. On this idea, we propose a new trace-clustering evaluation framework to evaluate clustering quality according to the inter-sublog event log similarity. The framework is algorithm-independent of any discovery algorithm and free from the high computational overhead of repeated conformance checking. To demonstrate the application of the framework, we instantiate the framework with three state-of-the-art event log-similarity metrics.

The main contributions of this paper can be summarized as follows: (1) This paper formulates the problem of trace-clustering evaluation for process discovery as the need for an efficient and algorithm-independent evaluation method. (2) This paper proposes a novel evaluation framework that assesses clustering quality through inter-sublog event log similarity rather than through discovered process models. (3) This paper instantiates the framework with three state-of-the-art log-similarity metrics and validates its effectiveness and efficiency through experiments on real-life event logs.

The rest of this paper is structured as follows. Section 2 surveys related studies, and Section 3 provides the necessary preliminaries. Section 4 details the proposed log-similarity-based evaluation framework, followed by tool support in Section 5. Section 6 presents the experimental results, and Section 7 concludes the paper.

2. Related Work

This section reviews prior work on trace clustering and clustering quality evaluation in process mining. In process discovery, clustering is mainly used to decompose a complex event log into behaviorally coherent sublogs, so that simpler and more interpretable process models can be discovered from them. Existing studies have proposed various clustering strategies for this purpose, but the evaluation of clustering quality remains a major challenge.

2.1. Trace clustering

Most trace-clustering techniques are developed based on data mining clustering methods. A common approach transforms an event log into a vector-space representation, defines a distance between traces, and then applies a standard clustering algorithm. In addition to feature-based methods, control-flow-aware extensions have been proposed to incorporate contextual information about process behavior [12].

Greco et al. [13] pioneered trace clustering in process mining by representing each trace in a vector-space model and clustering the resulting vectors; for each sublog, they then discovered a simpler model using disjunctive workflow schemes. Bose

and van der Aalst [12] proposed control-flow-aware trace clustering by incorporating context attributes that characterize the behavioral structure of traces within an event log. First, a general edit-distance formulation for traces was introduced. Then, a context-aware clustering method was developed by constructing a vector-space representation for each trace, where features are extracted from conservative patterns or subsequences rather than basic activity profiles. Inspired by Web-usage mining [14], Ferreira et al. [15] introduced a notably different trace-clustering approach: the expectation-maximization algorithm is employed to learn first-order Markov models and perform model-based clustering of traces. The method was later evaluated using real server log data. This approach was subsequently validated on real server logs. In related work, De Koninck et al. [16] present two constrained trace-clustering techniques that leverage expert knowledge through instance-level constraints. Experiments on real-life data show that these techniques produce more justifiable clustering solutions without a substantial degradation in clustering quality. Fang et al. [17] introduced the concepts of benchmark logs and trace context trees to formalize the contextual semantics of event logs and support the classification of benchmark logs.

Overall, existing trace-clustering methods differ in how they represent traces and define similarity, but they share the same objective: to partition the original log into sublogs that can lead to more understandable process models. However, while clustering methods have been extensively studied, the question of how to evaluate clustering quality for process discovery remains insufficiently addressed.

2.2. Trace-clustering quality evaluation

Several quantitative metrics have been used to evaluate trace-clustering quality in process mining. For example, weighted fitness is adopted in the paper by Bose and van der Aalst [12], while overall model precision together with instance-level fitness is adopted in the paper by Fang et al. [17]. From the model-discovery perspective, the *F-Measure* [9] is a widely used metric, computed as the harmonic mean of fitness and precision [11], as given in Equation (1):

$$F - Measure(L, M) = \frac{2 \times fitness(L, M) \times precision(L, M)}{fitness(L, M) + precision(L, M)} \quad (1)$$

Here, $fitness(L, M)$ denotes the *fitness* of process model M with respect to event log L . A *fitness* of 1 means that all traces in L can be replayed by M , whereas lower values indicate that a larger share of observed behavior cannot be reproduced by M . $Precision(L, M)$ quantifies the proportion of behavior that can be repeated in the process model M but not seen in the event log L . A precision value of 1 indicates that all traces generated by M are included in L , whereas lower values indicate that M allows more behavior that is not observed in L .

Although the *F-Measure* is widely adopted, it has two important limitations for trace-clustering evaluation in process discovery. First, it is discovery-algorithm-dependent: because the score is computed from models discovered from sublogs, different discovery algorithms may yield different models and thus different evaluation results for the same clustering. As a result, the score reflects not only clustering quality alone but also the behavior of the chosen discovery technique. Second, it is computationally expensive: computing the *F-Measure* requires discovering process models for sublogs and performing repeated conformance

checking, while both fitness and precision are known to be costly to compute.

These limitations suggest the need for a different perspective on clustering evaluation. Rather than treating evaluation as a by-product of discovered models, this paper treats log similarity as a novel evaluation proxy for clustering quality. The key idea is that, if a clustering result is meaningful for process discovery, the resulting sublogs should be behaviorally distinct at the event log level. In this sense, log similarity is not introduced merely as another metric but as a surrogate for discovery-relevant clustering quality that avoids direct dependence on model discovery and conformance checking.

3. Preliminaries

Let S be a set and $|S|$ denotes the number of elements in set S . $B(S)$ is the collection of all multisets over set S . Let $f: X \rightarrow Y$ be a function. The set of all finite sequences over S is denoted by S^* .

Definition 1. (Events Traces and Event Logs) Let E be a set of events. A trace $\sigma \in E^*$ is a sequence of events. For any $1 \leq i \leq |\sigma|$, $\sigma(i)$ denotes the i -th event of σ . An event log is a multiset $L \in B(E^*)$.

For example, the event log $L_0 = [\langle a, c \rangle, \langle a, b, c \rangle, \langle b, c \rangle]$ contains 3 traces and 7 events.

Definition 2. (Events Traces and Event Logs) Let a and b be two activities occurring in a trace σ of an event log L . We say that a directly-follows to b , denoted by $\langle a, b \rangle$, if there exists an index i such that $\sigma(i) = a$, and $\sigma(i+1) = b$.

For instance, the event log L_0 contains 3 directly-follows relations $\langle a, c \rangle$, $\langle a, b \rangle$, $\langle b, c \rangle$.

Definition 3. (Events Traces and Event Logs) Let L be an event log. We define $M(L)$ as an $N \times N$ matrix, where N is the number of activities appearing in L . Each value/cell records the frequency of the directly-follows relation between two activities. If no directly-follows relation exists from a to b , then $M(L)_{\langle a, b \rangle} = 0$.

For example, the event log L_0 , its correspondence matrix $M(L_0)$ is shown in Table 1.

Definition 4. (Events Traces and Event Logs) Let $L \subseteq C$ be an event log. A trace cluster for L is any subset of L . A trace clustering $TC \subseteq P(L)$ is a collection of such clusters. Each trace must belong to at least one cluster, that is, $\cup TC = L$. Here, $P(L)$ denotes all possible sublogs in event log L .

In process discovery, trace clustering is used to partition a complex event log into sublogs that can support simpler and more interpretable discovered models. Accordingly, the quality of a trace clustering should be assessed at the clustering-structure level, rather than only through the quality of process models discovered afterward.

Definition 5. (Events Traces and Event Logs) Let $TC = \{L_1, L_2, \dots, L_k\}$ be a trace clustering of an event log L . We define clus-

tering quality from two complementary aspects: (1) inter-sublog separation, which measures how behaviorally distinct different sublogs are, and (2) intra-sublog cohesion, which measures how behaviorally coherent the traces within the same sublog are.

A clustering with high quality should ideally exhibit high inter-sublog separation and, optionally, high intra-sublog cohesion.

In this paper, we focus primarily on inter-sublog separation. The reason is that, from a process discovery perspective, if two sublogs remain highly similar, then the clustering has not effectively separated different behavioral patterns in the original log. In that case, discovering separate process models from these sublogs provides limited additional interpretability. In contrast, low similarity between sublogs indicates that the clustering captures distinct behavioral structures, which is more likely to yield simpler and more meaningful submodels.

It is important to distinguish evaluation of clustering structure from evaluation of discovered models. The former evaluates whether a clustering meaningfully separates the event log into behaviorally distinct sublogs. The latter evaluates the quality of process models discovered from these sublogs, for example by fitness, precision, or *F-Measure*. Our framework addresses the former and uses log-level similarity as an evaluation proxy, rather than evaluating discovered models directly.

4. Clustering-Based Evaluation

This section first outlines traditional trace-clustering methods and motivates our log-similarity evaluation. Then, the proposed framework is described in depth.

4.1. An approach overview

As illustrated in Figure 1, traditional trace-clustering evaluation is typically performed in three phases. First, the original event log is clustered to obtain a set of sublogs. Then, a process model is discovered from each sublog using some discovery algorithm. Finally, clustering quality is quantified by computing the *F-Measure* over all pairs of sublog and model discovered. However, *F-Measure*-based evaluation has two limitations: (i) they are sensitive to the discovery algorithm used, and (ii) fitness and precision have to be computed, which entails high computation costs.

We overcome these limitations by presenting a trace-clustering evaluation framework that can quantify the similarity between the sublogs in an efficient and robust way. As shown in Figure 2, the framework consists of two phases:

Phase 1 (Trace clustering): A selected trace-clustering method (e.g., ActiTrac [18]) is applied to the original event log to partition it into a set of sublogs; and

Phase 2 (Log-similarity-based quality evaluation): A high-quality clustering produces sublogs with high within-cluster similarity and low between-cluster similarity. Accordingly, clustering quality is assessed by quantifying similarity among the resulting sublogs, where higher intra-sublog similarity and lower inter-sublog similarity indicate better performance. This log-similarity assessment provides an algorithm-agnostic, efficient basis for comparing trace-clustering techniques.

Table 1
The correspondence matrix of L_0

	a	b	c
a	0	1	1
b	0	0	2
c	0	0	0

Figure 1
Traditional trace-clustering evaluation

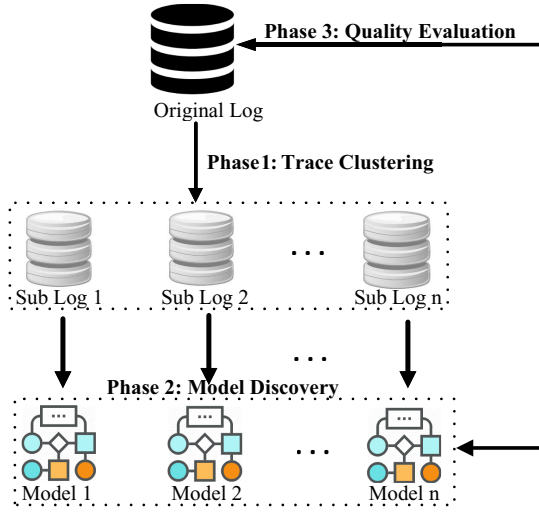
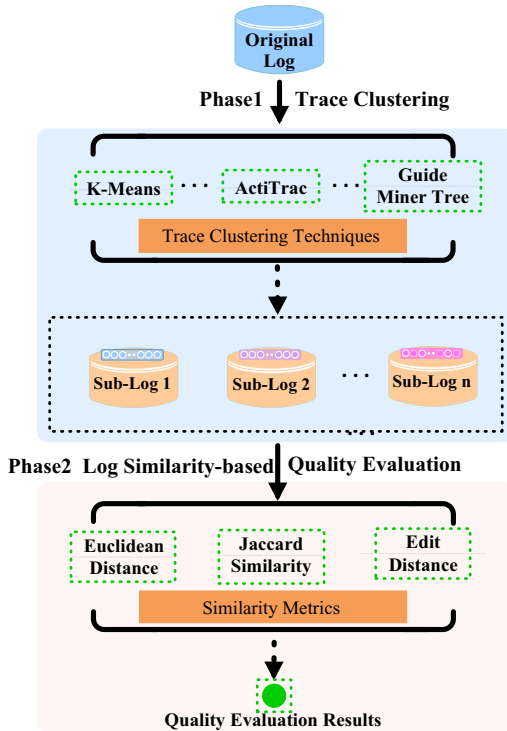


Figure 2
The novel trace-clustering evaluation framework



4.2. Log-similarity-based quality evaluation

To showcase the effectiveness and practical applicability of the proposed framework, it is instantiated with three state-of-the-art log-similarity metrics, which are reviewed below.

It is important to note that the log-similarity metrics used in this paper are intended as evaluation surrogates rather than replacements for model-quality measures. They do not directly assess the quality of process models discovered from sublogs. Instead, they provide a log-level proxy for clustering quality from a process discovery perspective, enabling a more efficient and discovery-algorithm-independent evaluation.

4.2.1. Euclidean-distance similarity

Euclidean-distance similarity [19] is adopted as a simple and interpretable baseline. Given two logs, their correspondence matrices are first constructed. Let X and Y denote corresponding row vectors of the two matrices. The row-wise Euclidean distance is computed via Equations (2) and (3) and is then normalized to a $[0, 1]$ similarity score for each row. The log-level similarity is obtained by averaging these row-wise scores. A smaller Euclidean-distance similarity indicates a lower similarity between the two logs.

$$d(X, Y) = \sqrt{\sum (x_i - y_i)^2} \quad (2)$$

$$\text{sim}_{\text{Euclidean}}(X, Y) = \frac{d(X, Y)}{1 + d(X, Y)} \quad (3)$$

Given two event logs $L_1 = [\langle a, b, d \rangle, \langle a, c \rangle, \langle a, c, d \rangle]$ and $L_2 = [\langle a, b, c \rangle, \langle b, c, d \rangle, \langle a, c \rangle]$, their correspondence matrices are shown in Tables 2 and 3.

The row vectors of the correspondence matrix of L_1 and L_2 are $X_1 = [0, 1, 2, 0]$, $X_2 = [0, 0, 0, 1]$, $X_3 = [0, 0, 0, 1]$, $X_4 = [0, 0, 0, 0]$, and $Y_1 = [0, 1, 1, 0]$, $Y_2 = [0, 0, 2, 0]$, $Y_3 = [0, 0, 0, 1]$, and $Y_4 = [0, 0, 0, 0]$, respectively. These vectors are used to compute the Euclidean similarity values via Equations (2) and (3). For example, the Euclidean-distance similarity between L_1 and L_2 is 0.298.

4.2.2. Jaccard similarity

Jaccard similarity [20] quantifies the overlap between two sets. For event logs, each correspondence matrix is first converted into row-wise binary vectors by mapping nonzero entries to 1. Then, the Jaccard similarity is computed for each pair of corresponding row vectors and aggregated across rows using the mean. For row vectors X and Y , the Jaccard similarity is calculated by Equation(4).

$$\text{sim}_{\text{jaccard}}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad (4)$$

In Equation (4), $|X \cap Y|$ represents the number of identical elements in two vectors, and $|X \cup Y|$ represents the number of all different elements in two vectors. Higher Jaccard values indicate greater similarity between logs, whereas lower values indicate less

Table 2
The correspondence matrix of L_1

	a	b	c	d
a	0	1	2	0
b	0	0	0	1
c	0	0	0	1
d	0	0	0	0

Table 3
The correspondence matrix of L_2

	a	b	c	d
a	0	1	1	0
b	0	0	2	0
c	0	0	0	1
d	0	0	0	0

similarity. In this example, the Jaccard similarity between L_1 and L_2 is 0.75.

4.2.3. Edit-distance similarity

The edit distance between two traces is defined as the smallest number of basic operations—insertions, deletions, or substitutions—needed to convert one trace into the other [12]. Let $S \in A^*$ and $T \in B^*$ be two traces with length $|S|$ and $|T|$, respectively. The edit distance is computed via standard dynamic programming.

Consider $S1 = \langle a, c, d, e, f \rangle$ and $T1 = \langle a, b, d, f, g \rangle$, where $|S1| = |T1| = 5$. The shared activities are a, d, f ; the activities unique to S are c, e , and those unique to $T1$ are b, g . The edit distance $d_{edit}(S1, T1)$ is the minimum number of insertions, deletions, and substitutions needed to transform $S1$ into $T1$ [12]. One optimal edit sequence that transforms $S1$ into $T1$ is to substitute c to b , delete e , and insert g . Therefore, $d_{edit}(S1, T1) = 3$. To obtain a similarity score in $[0, 1]$ (larger values indicate higher similarity), the distance is normalized as in Equation (5).

$$sim_{edit}(S, T) = 1 - \frac{d_{edit}(S, T)}{\max\{|S|, |T|\}} \quad (5)$$

Applying Equation (5) to the example gives $sim_{edit}(S1, T1) = 1 - 3/5 = 0.4$. Other normalizations are possible, for example,

$$sim_{edit}^{(sum)}(S, T) = 1 - \frac{2 \times d_{edit}(S, T)}{|S| + |T|} \quad (6)$$

which is equivalent to Equation (5) when $|S| = |T|$. Unless otherwise noted, we use (5).

5. Tool Support

The open-source ProM 6 framework provides a pluggable environment for process mining and related tasks. It can be extended by adding plug-ins and currently offers more than 1,600 plug-ins. The framework is freely downloadable from Prom6¹.

The proposed trace-clustering evaluation framework has been implemented as an independent ProM 6 plugin, named log-similarity-based evaluation method¹. The plugin takes two event logs as input, applies a user-configured similarity metric, and outputs a single similarity value. Two screenshots of the tool are shown in Figures 3 and 4.

The proposed tool supports scalable experimentation by automating the evaluation workflow and enabling batch processing of large numbers of clustering results across different datasets and parameter settings. Since it evaluates clustering quality directly at the log level, without repeatedly invoking process discovery and conformance checking, it substantially reduces computational overhead and facilitates efficient, reproducible, and high-throughput experimental comparison.

6. Experimental Results

This section first describes the experimental environment and datasets. Then, clustering quality is evaluated using the proposed log-similarity metrics. Next, the conventional *F-Measure* evaluation is compared with the log-similarity-based evaluation. Finally, the runtime performance of the evaluated metrics is reported. The

Figure 3
A snapshot of the evaluation tool

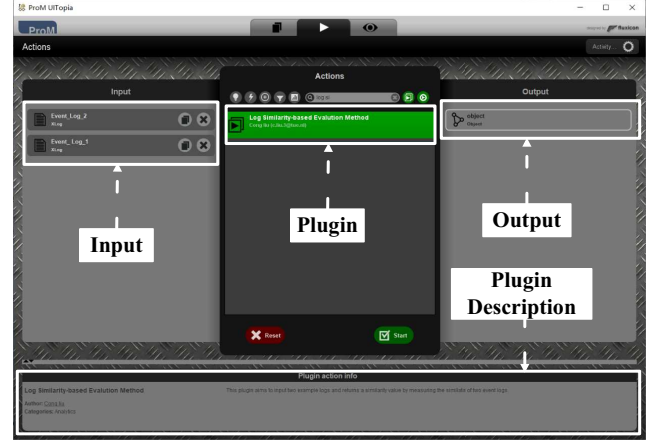
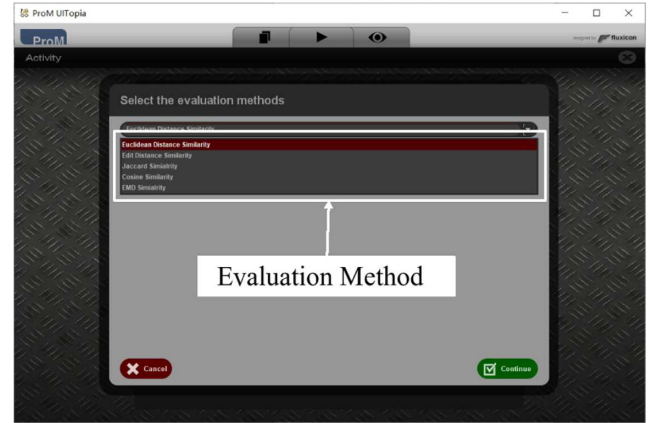


Figure 4
Similarity metric configuration



reported runtime was averaged over five independent runs. The experiments were conducted on a laptop with a 2.40 GHz CPU and 16 GB of allocated RAM, using Windows 8.1 and Java SE 1.7.0_67 (64-bit).

6.1. Experimental setting

The proposed trace-clustering evaluation method is evaluated on four public real-life event, as shown in the Data Availability Statement. The basic statistics of these logs are reported in Table 4. In addition to conventional size-related statistics, this paper also reports several descriptors of behavioral complexity, including the number of variants (#Var) and the average trace length (#Atl). The descriptor provides a more informative characterization of the logs by quantifying the richness and dispersion of trace behaviors, which are important for interpreting clustering quality from a process discovery perspective.

Three trace-clustering methods—ActiTrac, Guide Miner Tree, and k-means [21]—are evaluated with the number of cluster counts $K \in \{3, 4, 5, 6\}$. This range is selected to represent a small-to-medium number of clusters, which is commonly used in exploratory trace clustering for process discovery. This range provides enough variation to observe how clustering quality changes as the log is partitioned more finely, while avoiding excessively

¹<https://www.promtools.org/prom6>

Table 4
Summary of the experiment logs

Event Log	#Trace	#Event	#Act	#Var	#Atl
Cb2500	2500	50194	38	1605	20
NASA	2566	73638	94	2513	29
BPI2012A	13087	73022	53	32	20
BPI20153	1409	59681	383	1349	42

fragmented sublogs that may produce unstable or difficult-to-interpret process models. For model discovery, the Inductive Miner, a state-of-the-art technique, is adopted.

For the baseline model-based evaluation, Inductive Miner is used as the discovery algorithm. This choice is motivated by its representativeness, stability, and wide adoption in process mining. Moreover, using a single discovery algorithm provides a consistent and reproducible basis for computing *F-Measure* across all clustering results. It should be noted, however, that the proposed evaluation framework itself is algorithm-independent: it evaluates clustering quality directly at the log level and does not rely on any specific process discovery algorithm. In this sense, Inductive Miner is used only to support comparison with the conventional *F-Measure*-based evaluation, rather than as a required component of the proposed method.

6.2. Evaluation results comparison

Clustering aims to divide data into groups such that similarity is high within each cluster and low across different clusters [22]. In the considered setting, lower log similarity among sublogs indicates higher clustering quality. By contrast, a higher *F-Measure* indicates better clustering outcomes from the model-discovery perspective. Therefore, the log-similarity metric is negatively correlated with the traditional *F-Measure*.

The quality evaluation results for the four experimental logs are shown in Tables 5–8. Across all cases, the traditional *F-Measure* is negatively correlated with the three log-similarity-based metrics. For example, on the NASA log, the *F-Measure* of the ActiTrac clustering approach decreases as the number of clusters K increases, whereas the log-similarity metric increases, exhibiting the expected inverse trend.

The effect of the number of clusters K can be understood as the result of two competing mechanisms. From the clustering-structure perspective, increasing K usually produces finer-grained sublogs and tends to strengthen behavioral separation across sublogs. From the model-discovery perspective, however, a larger K also makes sublogs smaller and more fragmented, which may reduce behavioral support and make the discovered models less robust. This explains why, on logs such as Cb2500 and NASA, the *F-Measure* generally decreases as K increases, whereas the log-level surrogate metrics often move in the opposite direction. The NASA results of ActiTrac provide a representative example of this phenomenon.

The comparison across clustering algorithms reveals additional differences. ActiTrac and Guide Miner Tree tend to show smoother changes across K , suggesting that these process-aware methods produce relatively stable decompositions of the original log. In contrast, k-means is more volatile on BPI2012A and especially on BPI20153. This suggests that centroid-based partitioning in a vector-space representation may sometimes separate traces in the chosen feature space without producing equally meaningful sublogs from a process discovery perspective. In other words, representation bias can be amplified when the behavioral structure of the log is complex or weakly captured by the clustering features.

The failure cases deserve separate discussion. The clearest exception occurs on BPI20153, where the k-means results do not consistently follow the expected inverse relation between the surrogate metrics and the *F-Measure*. As already noted in the current results, this log has high variant complexity, and the deviation is attributed to both the difficulty of the dataset and the limited interpretability of existing trace-clustering methods on it. In such cases, small changes in cluster boundaries may have a large effect on the discovered models, while the log-level surrogate metrics mainly respond to distributional separation between sublogs. Similar local deviations can also happen when the clustering algorithm is biased toward the underlying representation or when simplified trace representations do not fully preserve ordering and control-flow dependencies. Such findings indeed highlight that log similarity should be seen as a surrogate for discovery-oriented clustering quality, rather than a replacement for model-quality estimation.

The quantitative measure for the relationship between the *F-Measure* and the three log-similarity metrics is further expressed

Table 5
Quality evaluation results of Cb2500 log

Clustering Method	K	Quality Evaluation Metric			
		F-Measure	Euclidean	Edit	Jaccard
ActiTrac	3	0.472	0.002	0.53	0.444
	4	0.429	0.007	0.948	0.449
	5	0.372	0.006	0.954	0.451
	6	0.352	0.0132	0.958	0.45
Guide Miner Tree	3	0.51	0.0064	0.527	0.453
	4	0.451	0.0138	0.527	0.45
	5	0.382	0.01	0.528	0.45
	6	0.325	0.026	0.526	0.451
K-means	3	0.4905	0.005	0.5268	0.443
	4	0.4803	0.007	0.528	0.443
	5	0.4795	0.01	0.5284	0.444
	6	0.412	0.018	0.5252	0.438

Table 6
Quality evaluation results of NASA log

Clustering Method	K	Quality Evaluation Metric			
		F-Measure	Euclidean	Edit	Jaccard
ActiTrac	3	0.094	0.202	0.939	0.482
	4	0.074	0.248	0.948	0.483
	5	0.076	0.264	0.954	0.484
	6	0.067	0.287	0.958	0.485
Guide Miner Tree	3	0.296	0.089	0.931	0.46
	4	0.241	0.1021	0.937	0.45
	5	0.237	0.0897	0.937	0.456
	6	0.235	0.09	0.927	0.457
K-means	3	0.26	0.09	0.927	0.448
	4	0.257	0.0901	0.934	0.452
	5	0.228	0.09	0.936	0.458
	6	0.19	0.091	0.937	0.452

Table 7
Quality evaluation results of BPI2012A log

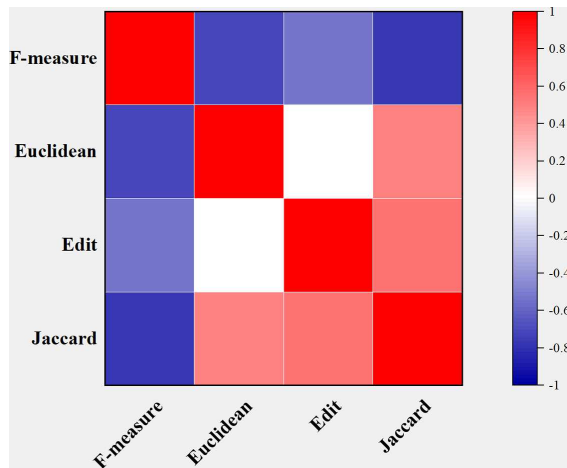
Clustering Method	K	Quality Evaluation Metric			
		F-Measure	Euclidean	Edit	Jaccard
ActiTrac	3	0.203	0.119	0.102	0.4872
	4	0.174	0.155	0.154	0.4886
	5	0.157	0.181	0.178	0.4886
	6	0.143	0.233	0.197	0.4879
Guide Miner Tree	3	0.216	0.728	0.071	0.4566
	4	0.178	0.547	0.09	0.4673
	5	0.16	0.579	0.095	0.465
	6	0.162	0.582	0.113	0.4653
K-means	3	0.143	0.221	0.304	0.4378
	4	0.205	0.585	0.063	0.4442
	5	0.164	0.351	0.083	0.4479
	6	0.159	0.293	0.106	0.4504

Table 8
Quality evaluation results of BPI20153 log

Clustering Method	K	Quality Evaluation Metric			
		F-Measure	Euclidean	Edit	Jaccard
ActiTrac	3	0.0164	0.1528	0.9612	0.472
	4	0.0169	0.1651	0.9406	0.472
	5	0.0155	0.1995	0.942	0.473
	6	0.0146	0.1885	0.9452	0.467
Guide Miner Tree	3	0.0228	0.199	0.988	0.496
	4	0.026	0.195	0.988	0.496
	5	0.0291	0.193	0.986	0.494
	6	0.0299	0.207	0.986	0.493
K-means	3	0.0456	0.2423	0.9899	0.4964
	4	0.0258	0.2588	0.99	0.4958
	5	0.0915	0.2825	0.9912	0.494
	6	0.0103	0.3011	0.9916	0.4942

with the Spearman rank correlation [23] (Tables 5–8), which are summarized in Figure 5. All three log-similarity metrics are negatively correlated with the *F-Measure*. The Spearman correlations between the *F-Measure* and Euclidean, Edit, and Jaccard are -0.70, -0.54, and -0.77, respectively, with p-values of 3.08e-8, 7.45e-5, and 1.58e-10, respectively. Among them, Jaccard shows the strongest negative correlation, followed by Euclidean and Edit. These findings further support the reasonableness of the proposed metrics.

Figure 5
Correlation results



Taken together, the results suggest that log-level surrogates are particularly suitable for coarse-to-medium-grained ranking of clustering candidates. They provide a useful first-stage evaluator for identifying promising clusterings, after which a smaller number of shortlisted candidates can be further validated by model-based criteria such as the *F-Measure* when maximum evaluation fidelity is required.

6.3. Time performance comparison

The runtime comparison between the traditional *F-Measure*-based evaluation and the proposed log-similarity-based evaluation is shown in Figure 6. For the *F-Measure*, the reported time includes (i) discovering a model from each sublog using the Inductive Miner and (ii) computing fitness and precision to obtain the *F-Measure*. By contrast, the log-similarity-based evaluation measures only inter-sublog similarity, without requiring model discovery or conformance checking.

As shown in Figure 6, the log-similarity approach is substantially faster than the *F-Measure* baseline. For example, on Cb2500 log, using ActiTrac with $K = 6$ reduces the runtime from 47,547 ms to 63 ms when using a similarity metric (e.g., cosine), yielding an approximately 745× speedup. The three similarity metrics exhibit comparable runtimes. Overall, these results indicate that the proposed log-similarity evaluation provides a much more efficient assessment of clustering quality, while remaining reliable from a model-discovery perspective.

This runtime gap can be further explained from a qualitative complexity perspective. Let K denote the number of sublogs. The *F-Measure*-based baseline requires repeated process discovery and repeated conformance checking, that is, at least K model-discovery runs plus fitness/precision computation, and in the

all-pairs setting the conformance-related workload grows quadratically with the number of sublogs. By contrast, the proposed method only requires pairwise sublog comparison, which can be expressed qualitatively as $O(K^2 C_{sim})$, where C_{sim} denotes the cost of one log-level comparison. Although both approaches may include pairwise terms, the practical constants are fundamentally different because log-level comparison is much cheaper than the combination of process discovery and conformance checking.

These results also reveal an important trade-off between computational efficiency and evaluation fidelity. The proposed method achieves its efficiency advantage by avoiding model discovery and conformance checking, but it therefore evaluates clustering quality only indirectly. It cannot directly capture model replay-ability, overgeneralization, or structural soundness. Nevertheless, the overall correlation trends in Figure 5 and the substantial speedup in Figure 6 indicate that the method offers a favorable balance between efficiency and fidelity: it is accurate enough for comparative screening, while being much cheaper to apply repeatedly.

This balance is especially important for large-scale or iterative process mining studies. In such studies, many values of K , multiple clustering algorithms, feature encodings, or parameter configurations must often be compared across several logs. Under these conditions, repeated *F-Measure* computation becomes a bottleneck. The proposed surrogate-based evaluation is therefore particularly suitable as a scalable front-end tool: it can quickly narrow the candidate space, reserve expensive model-based evaluation for a small subset of promising solutions, and thus support more efficient and reproducible experimental workflows.

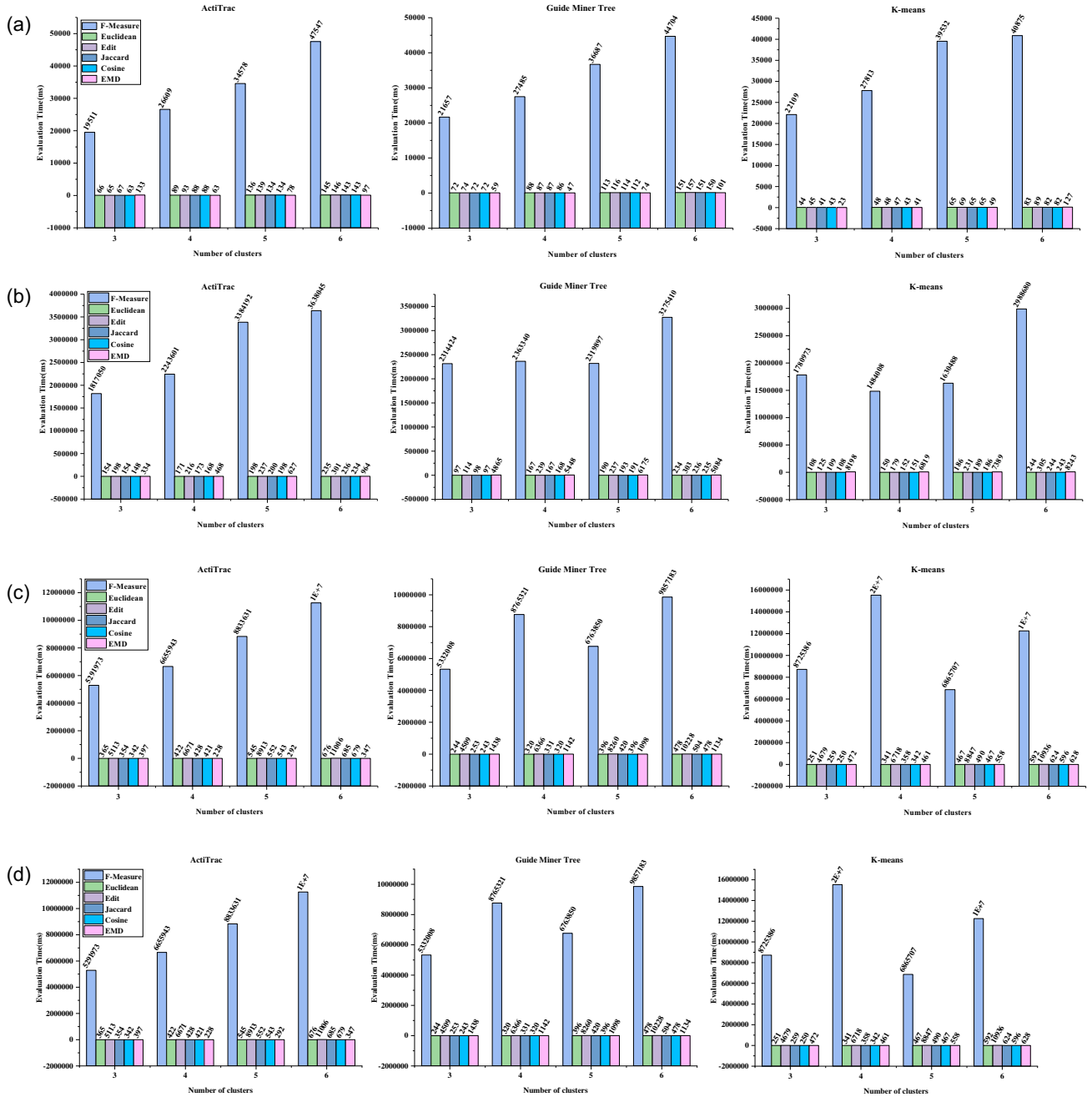
6.4. Discussion

The experimental results suggest that the proposed log-similarity-based evaluation is best understood as a replacement for the *F-Measure* only in specific stages of the evaluation workflow, rather than as a universal substitute.

On the one hand, the three surrogate metrics are overall negatively correlated with the *F-Measure* across the four logs, and the runtime results show a substantial efficiency advantage. On the other hand, the proposed approach should not be interpreted as a complete replacement for the *F-Measure* when the objective is final, model-level validation. The log-similarity metrics evaluate clustering quality at the log level and therefore do not directly assess replay-ability, precision, overgeneralization, or structural soundness of the process models discovered from the sublogs. This limitation becomes particularly important in difficult cases where the surrogate trend deviates from the expected inverse relation with the *F-Measure*. In the current experiments, the clearest example is k-means on BPI20153, where the behavior of the surrogate metrics does not fully align with the model-based evaluation. Such cases indicate that, for highly complex logs or unstable clustering settings, the proposed method should be used as an efficient proxy for preliminary assessment, while the *F-Measure* should still be retained for final confirmation of a small number of shortlisted candidates.

Therefore, the practical recommendation of this study is twofold. For exploratory experimentation, parameter tuning, repeated comparison across multiple values of K , or multi-log benchmarking, the proposed approach can effectively replace the *F-Measure* in order to reduce evaluation cost. However, for final reporting, high-stakes comparison, or any setting where the quality of the discovered process models must be verified directly, the *F-Measure* should remain part of the evaluation pipeline. In

Figure 6
Evaluation time of three trace-clustering methods on different logs: (a) Cb2500; (b) NASA; (c) BPI2012A; (d) BPI20153



this sense, the proposed method is best viewed not as a universal substitute but as a scalable and reliable first-stage evaluator for discovery-oriented trace clustering.

7. Conclusion

A trace-clustering evaluation framework is introduced to quantify similarity among sublogs. Compared with existing evaluation approaches, the proposed approach provides high-quality assessments while enabling more efficient experimental evaluation on clustered sublogs. It serves as a complementary surrogate for model-quality assessment rather than as a replacement for established model-quality measures. Moreover, the proposed method framework can be employed as an effective pre-screening and

ranking tool. For cases requiring more precise evaluation, the selected clustering results can be further examined through model-based verification, thereby balancing computational efficiency and analytical accuracy.

At the same time, the current work has several limitations. First, the proposed method evaluates clustering quality through a log-level surrogate and therefore does not directly assess the replay-ability, precision, or structural soundness of the process models discovered from sublogs. Second, the current framework focuses mainly on inter-sublog separation, while intra-sublog cohesion and finer-grained semantic distinctions are not yet fully incorporated. Third, when event logs are highly complex, clustering boundaries are unstable, or trace representations do not sufficiently preserve control-flow information, local deviations

from model-based evaluation criteria such as the *F-Measure* may still occur.

Several directions can be investigated in future work. First, the proposed evaluation framework can be extended to additional real-life domains, such as medical treatment processes [24], to further examine its generalizability. Second, semantics-aware similarity measures can be implemented so that event log comparison considers not only structural differences but also activity meaning and contextual information. Third, embedding-based log representations could provide a richer basis for comparison of traces and sublogs, in particular for logs with high behavioral differences or weakly structured patterns. Fourth, a hybrid evaluation approach mixing fast log-level surrogate evaluations with model-level quality metrics may provide a less demanding trade-off between time-efficiency and evaluation quality. And fifth, considering more trace-clustering methods [25] and more evaluation measures [26] can further reinforce the robustness analyses of our approach.

Funding Support

This work was partly supported by the National Natural Science Foundation of China under Grant 62472264 and 52574256, the Natural Science Distinguished Youth Foundation of Shandong Province under Grant ZR2025QA13, and the national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project – UID/04152/2025 – Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS and UID/PRR/04152/2025.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in 4TU.ResearchData at <https://doi.org/10.4121/uuid:aa01a720-4616-43e9-af67-370942019f48>, <https://doi.org/10.4121/uuid:60383406-fbcd-441f-aa5e-4ec763426b76>, <https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f>, and <https://doi.org/10.4121/uuid:ed445cdd-27d5-4d77-a1f7-59fe7360cfbe>.

Author Contribution Statement

Guodan Yin: Data curation, Writing – original draft. **Huilin Li:** Formal analysis, Writing – original draft, Visualization. **Cong Liu:** Software, Validation, Writing – review & editing, Project administration, Funding acquisition. **Qinghua Zhang:** Methodology, Data curation, Investigation. **Haijun Zhang:** Conceptualization, Resources, Supervision.

References

- [1] Vitale, F., Pegoraro, M., van der Aalst, W. M. P., & Mazzocca, N. (2025). Control-flow anomaly detection by process mining-based feature extraction and dimensionality reduction. *Knowledge-Based Systems*, 310, 112970. <https://doi.org/10.1016/j.knosys.2025.112970>
- [2] Chen, K., Abtahi, F., Carrero, J.-J., Fernandez-Llatas, C., & Seoane, F. (2023). Process mining and data mining applications in the domain of chronic diseases: A systematic review. *Artificial Intelligence in Medicine*, 144, 102645. <https://doi.org/10.1016/j.artmed.2023.102645>
- [3] Munoz-Gama, J., Martin, N., Fernandez-Llatas, C., Johnson, O. A., Sepúlveda, M., Helm, E., . . . , & Zerbato, F. (2022). Process mining for healthcare: Characteristics and challenges. *Journal of Biomedical Informatics*, 127, 103994. <https://doi.org/10.1016/j.jbi.2022.103994>
- [4] Amling, J., Slany, E., Dormagen, C., Kretschmann, M., & Scheele, S. (2025, July 9–11). Bridging the interpretability gap in process mining: A comprehensive approach combining explainable clustering and generative AI. In R. Guidotti, U. Schmid, & L. Longo (Eds.), *Explainable artificial intelligence: Third World Conference, xAI 2025 Proceedings, Part II* (2577, pp. 78–103). Istanbul, Turkey: Springer. https://doi.org/10.1007/978-3-032-08324-1_4
- [5] Schuster, D., van Zelst, S. J., & van der Aalst, W. M. P. (2022). Utilizing domain knowledge in data-driven process discovery: A literature review. *Computers in Industry*, 137, 103612. <https://doi.org/10.1016/j.compind.2022.103612>
- [6] Ullmann, T., Hennig, C., & Boulesteix, A.-L. (2022). Validation of cluster analysis results on validation data: A systematic framework. *WIREs Data Mining and Knowledge Discovery*, 12(3), e1444. <https://doi.org/10.1002/widm.1444>
- [7] Sommers, D., Sidorova, N., & van Dongen, B. (2025). A ground truth approach for assessing process mining techniques. *Process Science*, 2(1), 1. <https://doi.org/10.1007/s44311-025-00006-8>
- [8] Faizan, M., Zuhairi, M. F., & Ismail, S. (2021). Process discovery enhancement with trace clustering and profiling. *Annals of Emerging Technologies in Computing*, 5(4), 1–13. <https://doi.org/10.33166/AETiC.2021.04.001>
- [9] Lavazza, L., & Morasca, S. (2023). Common problems with the usage of F-measure and accuracy metrics in medical research. In *IEEE Access*, 11, 51515–51526. <https://doi.org/10.1109/ACCESS.2023.3278996>
- [10] Zhang, Z., Johnson, C., Venkatasubramanian, N., & Ren, S. (2022). Process scenario discovery from event logs based on activity and timing information. *Journal of Systems Architecture*, 125, 102435. <https://doi.org/10.1016/j.sysarc.2022.102435>
- [11] Zhang, D. (2024). Individual intervention and assessment of students' physical fitness based on the “three precision” applet and mixed strategy optimised CNN networks. *EAI Endorsed Transactions on Pervasive Health and Technology*, 10, 1–11. <https://doi.org/10.4108/eetpht.10.5852>
- [12] Bose, R. P. J. C., & van der Aalst, W. M. P. (2009). Context aware trace clustering: Towards improving process mining results. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, 401–412. <https://doi.org/10.1137/1.9781611972795.35>
- [13] Greco, G., Guzzo, A., Pontieri, L., & Sacca, D. (2006). Discovering expressive process models by clustering log traces. In *IEEE Transactions on Knowledge and Data Engineering*, 18(8), 1010–1027. <https://doi.org/10.1109/TKDE.2006.123>
- [14] Canay, Ö., & Kocabağ, Ü. (2023). An innovative data collection method to eliminate the preprocessing phase in web

- usage mining. *Engineering Science and Technology, an International Journal*, 40, 101360. <https://doi.org/10.1016/j.jestch.2023.101360>
- [15] Ferreira, D., Zacarias, M., Malheiros, M., & Ferreira, P. (2007). Approaching process mining with sequence clustering: Experiments and findings. In G. Alonso, P. Dadam, & M. Rosemann (Eds.), *Business Process Management: 5th International Conference*, 360–374. https://doi.org/10.1007/978-3-540-75183-0_26
- [16] De Koninck, P., Nelissen, K., Vanden Broucke, S., Baesens, B., Snoeck, M., & De Weerd, J. (2021). Expert-driven trace clustering with instance-level constraints. *Knowledge and Information Systems*, 63(5), 1197–1220. <https://doi.org/10.1007/s10115-021-01548-6>
- [17] Fang, H., Liu, W., Wang, W., & Zhang, S. (2023). Discovery of process variants based on trace context tree. *Connection Science*, 35(1), 2190499. <https://doi.org/10.1080/09540091.2023.2194578>
- [18] Senevirathna, P., Pires, D. E. V., & Capurro, D. (2025). Uncovering digital overdiagnosis–quantification and mitigation using clinical trajectories: Heparin-induced thrombocytopenia use case. *Journal of Biomedical Informatics*, 169, 104876. <https://doi.org/10.1016/j.jbi.2025.104876>
- [19] Bakhtiyarov, B., Jabiyeva, A., Mutallimova, A., Novruzova, R., & Khudaverdiyeva, M. (2026). Adaptive Gaussian-based kernel k-means: Scalable adaptive kernel-based clustering for big data. *Journal of Computational and Cognitive Engineering*, 5(2), 258–273. <https://doi.org/10.47852/bonviewjccce52026511>
- [20] Tang, M., Kaymaz, Y., Logeman, B. L., Eichhorn, S., Liang, Z. S., Dulac, C., & Sackton, T. B. (2021). Evaluating single-cell cluster stability using the Jaccard similarity index. *Bioinformatics*, 37(15), 2212–2214. <https://doi.org/10.1093/bioinformatics/btaa956>
- [21] Shang, R., Ara, B., Zada, I., Nazir, S., Ullah, Z., & Khan, S. U. (2021). Analysis of simple K-mean and parallel K-mean clustering for software products and organizational performance using education sector dataset. *Scientific Programming*, 2021(1), 9988318. <https://doi.org/10.1155/2021/9988318>
- [22] Khan, M. K., Ahmed, S. M., Sarker, S., & Khan, M. H. A. (2021). K-cosine-medoids clustering algorithm. In *2021 5th International Conference on Electrical Information and Communication Technology*, 1–5. <https://doi.org/10.1109/EICT54103.2021.9733540>
- [23] Yu, H., & Hutson, A. D. (2024). A robust Spearman correlation coefficient permutation test. *Communications in Statistics–Theory and Methods*, 53(6), 2141–2153. <https://doi.org/10.1080/03610926.2022.2121144>
- [24] Su, X., Liu, C., Lu, F., Cheng, L., Zeng, Q., & Zhou, J. (2025). Enhancing healthcare process model discovery through duplicate task identification. In *2025 IEEE International Conference on Web Services*, 477–483. <https://doi.org/10.1109/ICWS67624.2025.00067>
- [25] Zhang, F., Liu, D., & Liu, C. (2021). Difficulty-based SPOC video clustering using video-watching data. *IEICE Transactions on Information and Systems*, E104.D(3), 430–440. <https://doi.org/10.1587/transinf.2020EDP7106>
- [26] Liu, C. (2021). Discovery and quality evaluation of software component behavioral models. In *IEEE Transactions on Automation Science and Engineering*, 18(4), 1538–1549. <https://doi.org/10.1109/TASE.2020.3008897>

How to Cite: Yin, G., Li, H., Liu, C., Zhang, Q., & Zhang, H. (2026). Quality Evaluation of Clustering-Based Process Model Discovery Using Event Log Similarity. *Journal of Computational and Cognitive Engineering*. <https://doi.org/10.47852/bonviewJCCE62029268>