A Novel Trace Clustering Technique Based on Constrained Trace Alignment

Pan Wang[1(✉)], Wen'an Tan[1,2], Anqiong Tang[2], and Kai Hu[3]

[1] Nanjing University of Aeronautics and Astronautics, Nanjing, People's Republic of China
pwang@nuaa.edu.cn
[2] Shanghai Polytechnic University, Shanghai, People's Republic of China
{watan,aqtang}@sspu.edu.cn
[3] Beihang University, Beijing, People's Republic of China
hukai@buaa.edu.cn

Abstract. Whenever traditional process discovery techniques are confronted with complex and flexible environments, equipping all the traces with just one single model might lead to a spaghetti-like process description. Trace clustering which splits the logs into clusters and applies discovery algorithm per cluster has affirmed to be a versatile solution for that. Nevertheless, most trace clustering techniques are not precise enough due to the indiscriminate treatment on the activities captured in traces. As a result, the impacts of some important activities are reduced and some typical information may be distorted or even lost during comparison. In this paper, we propose a novel trace clustering technique that based on constrained traces alignment and then adapt two appropriate clustering strategies into process mining perspective. And experiments on real-life event logs show that our technique has compelling outperformance in terms of process models complexity and comprehensibility.

Keywords: Constrained Trace Clustering · Trace clustering · Process mining
Business process management · Constrained Trace Alignment

1 Introduction

Process discovery is one of the most crucial process mining tasks that entails the construction of process models from event logs of information systems [1]. The most arduous challenge for process discovery is tackling the problem that discovery algorithms are unable to generate accurate and comprehensible process models out of event logs stemming from highly flexible environments.

Trace Clustering is an efficient solution, which *clusters the traces such that each of the resulting clusters corresponds to coherent sets of cases that can each be adequately represented by a process model* [3]. Figure 1 shows the basic procedure for trace clustering.

Q. Zu and B. Hu (Eds.): HCC 2017, LNCS 10745, pp. 53–63, 2018.
https://doi.org/10.1007/978-3-319-74521-3_7

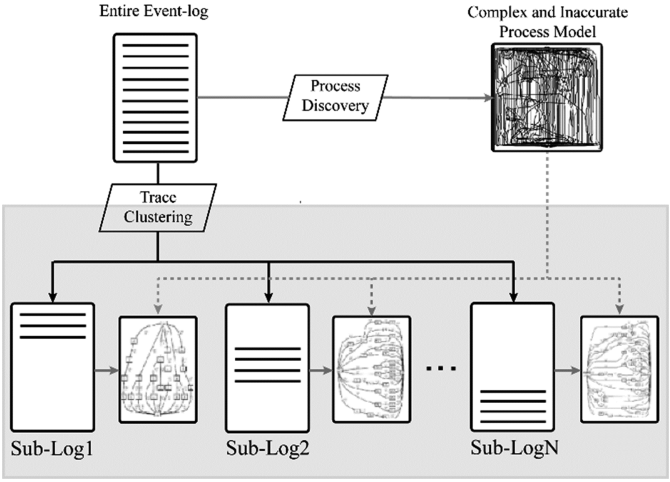

Fig. 1. Illustration of the basic procedure of trace clustering in process mining.

Nevertheless, most currently available trace clustering techniques are not precise enough due to the indiscriminate treatment on the activities captured in traces. As a result, the impacts of some important activities are reduced and some typical process information may be distorted or even lost during comparison.

To address the drawback, this paper presents a novel similarity measurement based on constrained traces alignment. First, some typical causal sequences that reflect the "backbone" of process are identified. Then, these sequences are exploited as constraints to guarantee the priority of important activities in traces. Subsequently, we suggest two clustering strategies that agree with the process mining perspective. The agglomerative hierarchical clustering (AHC) was selected for its embedded flexibility on abstraction level to provide us an overall insight into the complex process. And the spectral clustering has a good recommendation about the number of clusters corresponding to the generic abstraction level.

In brief, this work contributes by proposing a novel constrained trace similarity measurement to guarantee the priority of important process episodes and subsequently adapting two appropriate clustering techniques into process mining perspective. In addition, experiments on real-life logs prove the improvements achieved by our method relative to six existing methods.

The rest of the paper is organized as follows: Sect. 2 provides a brief overview of related works. Next, Sect. 3 introduces our novel constrained trace similarity measurement and the process-adaptive clustering strategies we selected. And Sect. 4 discusses the experiment results. Finally, Sect. 5 draws conclusions and spells out directions for future work.

2 Related Work

The main distinction between trace clustering techniques is the clustering bias (distance/similarity measures) they proposed. Existing approaches in literature can be classified into two major categories.

2.1 Distance-Based Trace Clustering

2.1.1 Vector-Based Trace Clustering

Vector-based trace clustering approaches transform traces into a vector space. Then, clustering can be achieved combining different distance metrics in the vector space. Greco et al. [8] were pioneers in study of clustering log traces within the process mining domain. They make trace clusters through the vector space over the activities and their transitions to discover expressive process models at the first attempt. They also introduced a notion of disjunctive workflow schemas (DWS) for divisive trace clustering [5]. Song et al. [13] elaborated on constructing so-called profiles associated with multiple trace perspectives as the feature vector.

2.1.2 Context-Aware Trace Clustering

Context-aware trace clustering approaches regard the entire trace as a whole sequence which implies all the process context information. Then various string edit distance metrics can be applied on it in conjunction with standard clustering techniques. In [2], Bose and van der Aalst propose a generic edit distance which derives specific edit operation costs so as to take into account the behavior in traces. The context-aware method is further developed in [3], it leverages conserved patterns or subsequences as feature sets to describe the characteristic of a certain trace.

2.2 Model-Based Trace Clustering

2.2.1 Sequence Clustering

Sequence clustering algorithm creates first-order Markov chains for clusters cooperating with the expectation-maximization (EM) algorithm to determine the assignment of a certain sequence. It has been used to automatically group large protein datasets to search for homologous gene sequences in bioinformatics. This technique was migrated into trace clustering by [6].

2.2.2 Active Trace Clustering

Active trace clustering inherits the underlying idea of sequence clustering [15]. Therein a trace is added to the current cluster if the model discovered from the cluster including that trace satisfies the target threshold of fitness. An optimal distribution of execution traces over a given number of clusters is achieved whereby the combined accuracy of the associated process models is maximized. In this way, the quality of process model discovered is under control. More extension on it has been developed to support further objectives in [7].

3 Approach Design

Distance/similarity measurement and clustering strategy with its specific characteristics are both important cluster-theoretical aspects. Therefore, we introduce our approach in the two steps. The framework of the approach in this paper is depicted in Fig. 2.

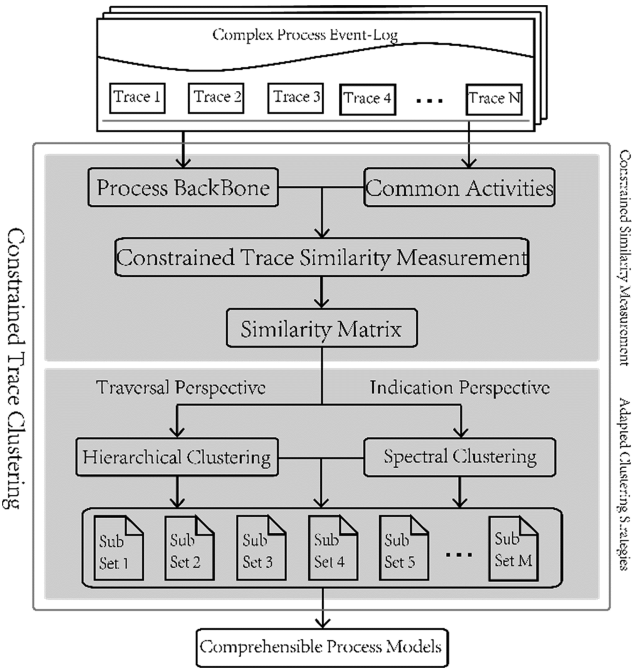

Fig. 2. Framework of Constrained Trace Clustering.

3.1 Similarity Measurement Based on Constrained Traces Alignment

Just as noted by [10], "specifying an appropriate dissimilarity measure is far more important than choice of clustering algorithm. This aspect of the problem is emphasized less in the clustering literature since it depends on domain knowledge specifics." Therefore, it is inevitable to refine the distance/similarity metrics in trace clustering for providing more appropriate information to the clustering algorithm.

We can perceive that identifying some significant behaviors in traces will assist in mining better sub-process models by clustering the traces based on those significant behaviors. However, due to the indiscriminate treatment on behaviors in traces and the lack of domain knowledge, capturing them directly from event logs seems to be a difficult task.

Fortunately, the association rules in data mining shed light on this tough task. Employing the association rules, we are able to reveal the "backbone" of process. Then, some typical causal sequences that reflect the process backbone are identified and

exploited as constraints to guarantee the priority of significant behaviors during similarity comparison.

3.1.1 Dependency Measures to Reveal Process Backbone

Definition 1. (Dependency Measures) Let L be an event log over A. a and b are activities that occur in L, i.e.; $|\sigma|$ denotes the length of the trace.

$|a >_L b|$ is the number of times a is directly followed by b in, i.e.,

$$|a >_L b| = \sum_{\sigma \in L} L(\sigma) \times |\{1 \le i \le |\sigma| \;\; |\sigma(i) = a \wedge \sigma(i+1) = b\}| \tag{1}$$

$|a \Rightarrow_L b|$ is the value of the dependency relation between a and b:

$$|a \Rightarrow_L b| = \begin{cases} \dfrac{|a >_L b| - |b >_L a|}{|a >_L b| + |b >_L a| + 1} & if \;\; a \neq b \\ \dfrac{|a >_L a|}{|a >_L a| + 1} & if \;\; a = b \end{cases} \tag{2}$$

$|a \Rightarrow_L b|$ produces a value between –1 and 1. If $|a \Rightarrow_L b|$ is close to 1, then there is a strong positive dependency between a and b.

By setting μ, the threshold of $|a >_L b|$, we can filter out the infrequent items. And when $|a \Rightarrow_L b|$ meets certain thresholds υ, we specified that there is a connection between a and b.

To illustrate the basic concepts, we use the following event log L:

$$L = [< a,d,e,f,g,i >^{16}, < a,b,e,f,i >^{1}, < a,c,e,f,h,i >^{1}, < a,c,b,e,f,h,i >^{12},$$
$$< a,e,f,g,i >^{5}, < d,d,a,d,e,f,g,i >^{1}, < a,b,c,e,f,h,i >^{13}, < a,d,d,d,e,f,g,i >^{1}]$$

Figure 3(A) depicts the dependency graph corresponding to the threshold of $\mu = 2$, $\upsilon = 0.7$. And Fig. 3(B) is another dependency graph with the threshold of $\mu = 5$, $\upsilon = 0.85$. Obviously, the dependency graph does not show the routing logic but it reveals the "backbone" of the process model. The derived dependency graph (denote as G) is then used as a reference to reveal the general order of some typical activities.

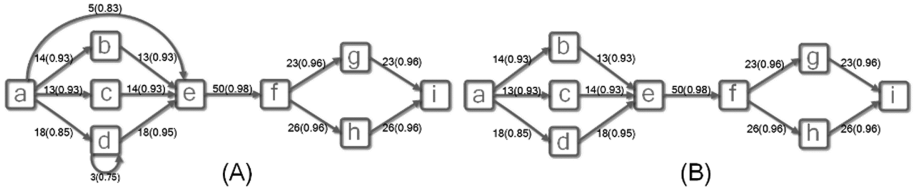

Fig. 3. Dependency graphs according to the dependency measures.

3.1.2 Constrained Similarity Measurement

Let A_i represents the set of activities that involved in trace σ_i (known as Alphabet) while B represents the set of activities that involved in the "process backbone". $A_i \cap A_j \cap B$ are the common activities between σ_i and σ_j with respect to the "process backbone". Referring to the dependency graph mentioned above, we can get the causal sequences of them. For example, the involved common activities between traces $< d, d, a, d, e, f, g, i >$, $< a, d, d, d, e, f, g, i >$ in L and the dependency graph are $< a, d, e, f, g, i >$, their constraints are shown as the black highlights in Fig. 4. We call these common activities that attached with causal sequences as typical behaviors.

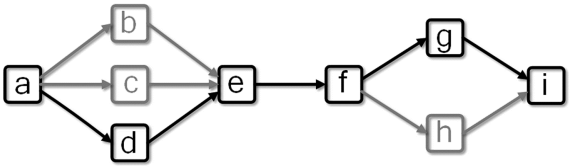

Fig. 4. Illustration of constraint instance.

These behaviors provide an approximation of the essence of the both comparing traces in a global perspective. Next, we utilize them as constraint conditions to guarantee the priority of typical behaviors between traces. In the following, the typical behaviors between traces σ_i and σ_j concerning the process backbone are denoted as $C_{i,j}$.

Definition 2. (Constrained Similarity Measurement) σ_i and σ_j are two traces, the constraints of them are $C_{i,j}$. Then, the similarity of σ_i and σ_j, $Sim(\sigma_i, \sigma_j)$ is defined as:

$$Sim(\sigma_i, \sigma_j) = \frac{length(CLCS(\sigma_i, \sigma_j, C_{i,j}))}{\max(length(\sigma_i),\ length\ (\sigma_j))} \tag{3}$$

The constrained measurement is relied on the constrained longest common sequences (*CLCS*). The *CLCS* has already been applied in bioinformatics for the computation of the homology of two biological sequences. For more details about *CLCS*, please refer to [14].

3.2 Adapted Trace Clustering Strategies

Traditional trace clustering only adapts data-centric clustering algorithms. However, as described in [3], the most important evaluation dimension for trace clustering is from a process discovery perspective. This entails the compatibility with process features on the adopted clustering strategies. Here, we suggest two apposite clustering strategies for different applications.

3.2.1 Agglomerative Hierarchical Clustering

Thanks to the hierarchical characteristic of AHC algorithm, it pertinently agrees with the *continuum* ranging from unstructured processes to structured processes. The bottom of AHC means clusters corresponding to each trace whose processes mined are surely structured while the top represents the only cluster that contains everything whose process mined is usually unstructured when confronted with flexible environment. Thus, we are able to ascertain the applicable level as desired or traverse all the hierarchy straightforward to gain an overall insight into the complex process.

The method we adopt is proposed by [4] termed as GuideTreeMiner. The GuideTreeMiner uses AHC algorithm to build a guide tree (also known as dendrogram). Any horizontal line spanning over the dendrogram corresponds to a practical clustering at a specific abstraction level.

3.2.2 Spectral Clustering

Except for hierarchical clustering algorithms, most of the trace clustering require predefined parameters for clustering such as the amount of clusters, the maximum cluster size etc. The truth is the definition of these specialized parameters are far from easy for general users due to the lack of domain knowledge. Actually, even for experts, it's also not a trivial work as well owing to the complexity and flexibility of real-life process. Against this background, the spectral clustering was selected as it provides a good recommendation about the number of clusters [11] which can guide us to a generic abstraction level.

It's worth point out that the *affinity* matrix always calculated as *Gaussian kernel*, however, it doesn't reflect the nature of processes. So, in this work, it is calculated as the constrained similarity described in the previous section. Likewise, the *laplacian* is often normalized, but duo to the robustness of constrained similarity measurement against infrequency and the pursuit of stable clustering indication, we select the non-normalized *laplacian*. As for the indication about the cluster number k, it can be recognized whereby the sudden drop in the eigenvalues. Actually, in many cases, the two solutions can be used in union.

4 Experiments

4.1 Experiment Configuration and Evaluation Criterion

We used the ProM[1] framework which has been developed to support process mining algorithms to perform the experiments. The data is from Dutch Financial Institute[2]. And we adopted the HeuristicsMiner to derive the process model as it has the best capability to deal with real-life event logs. The approaches to compare with are presented as follows: DWS Mining [5], Trace Clustering [13], GuideTree Miner [2, 3], Sequence Clustering [6] and ActiTraC [15].

[1] http://www.processmining.org.
[2] http://www.win.tue.nl/bpi/doku.php?id=2012:challenge.

We evaluate the results with respect to their model complexity, as they are measured by a comprehensive list of metrics reported in [12]:

1. $|A|$ signifies the number of arcs in the process model.
2. $|N|$ signifies the number of nodes in the process model.
3. $|CN| = |A| - |N| + 1$ signifies the cyclomatic number of the process model.
4. $CNC = \frac{|A|}{|N|}$ signifies the coefficient of connectivity of the process model.
5. $\Delta = \frac{|A|}{|N| \cdot |N-1|}$ signifies the density of the process model.

4.2 Clustering Results

We made comparisons with different number of clusters for three different abstraction levels. Here, level 1 represents the original trace set, i.e. there is only one cluster. Level 2 stands for 2/3/4 clusters while level 3 contains 5/6/7 clusters. We calculated $|A|$ by taking their corresponding nodes weighted average and the same as $|N|$.

The aggregated results are presented in Table 1. All the data has been depicted to the radarplots in Fig. 5. We can see that all cluster techniques lead to models with lower complexity than the original log file. However, the DWS, the ActiTraC and the Trace Clustering approaches lead to clusters whose models have higher density values than the unclustered one though they perform well in the other metrics. The smaller area and more balanced capabilities shown in the radarplots from two abstraction levels proved the effectiveness of our constraints.

Table 1. The aggregated clustering results

Abstraction level	Method	Cluster number	A	N	CNC	CN	Δ
Level 1	Unclustered	1	141.0	36.0	3.917	106.0	0.112
Level 2	SC	3	101.3	35.7	2.838	65.6	0.082
	DWS	4(Std.)	62.3	22.5	2.769	39.8	0.129
	TC	3(1–3)	50.0	16.7	2.994	33.3	0.191
	ATC	4(3-Std.)	21.5	12.0	1.792	9.5	0.163
	GED	3	86.6	34.6	2.503	52.0	0.074
	MRA	3	94.4	34.6	2.728	59.8	0.081
	Co-TC	3	86.5	35.3	2.450	51.2	0.071
Level 3	SC	6	74.5	34.7	2.147	39.8	0.064
	DWS	6(5-5-5-10)	41.7	19.0	2.195	22.7	0.122
	ATC	7(6-Std.)	15.6	10.1	1.544	5.5	0.170
	GED	6	76.5	33.6	2.277	42.9	0.070
	MRA	6	82.3	33.9	2.428	48.4	0.074
	Co-TC	6	73.5	33.1	2.221	39.4	0.069

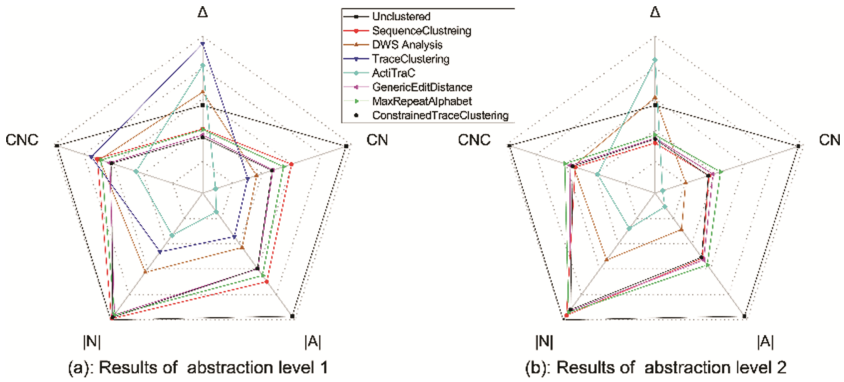

Fig. 5. Radarplots of different trace clustering techniques.

Moreover, Fig. 6 depicts Constrained Trace Clustering at different levels. With the increasing number of clusters, there is only a little improvement in all aspects. Considering the extra elaboration on more clusters, it is inefficient and meaningless to set the number of clusters to a higher value. This is consistent with the spectral clustering. Just as the eigenvalues scatterplot shown in Fig. 7, only the first three clusters are well separated. Therefore, the spectral clustering can guide us to correctly capture the right level of abstraction by providing a good recommendation about the number of clusters instead of the iterations on different hierarchies.

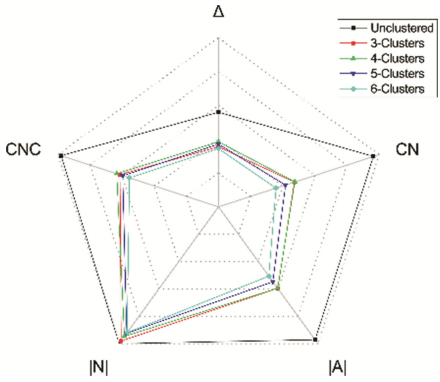

Fig. 6. Constrained Trace Clustering from different abstraction levels.

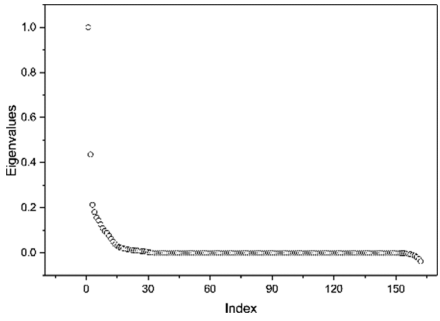

Fig. 7. Similarity matrix eigenvalues.

In a nutshell, all these experiments confirm the effectiveness and efficiency of Constrained Trace Clustering to deliver comprehensible process models from the flexible and complex logs.

5 Conclusions and Future Perspectives

In this paper, we contribute to trace clustering techniques by imposing constraints on trace similarity/distance measurements to guarantee the priority of important activities in traces. By this means, significant process information is preserved as much as possible such that more accurate trace similarity measurement will be obtained. Moreover, we integrate two clustering strategies that agree with the process mining perspective to cluster these traces.

There are still a number of challenging issues remain open for future research work. Firstly, more refined similarity/distance measurements that felicitously agree with the process domain knowledge are encouraged. On the issues of processing sequence data, we should learn from bioinformatics which has more mature applications on sequence data mining. Another direction of research might be that of integrating advanced clustering strategies, as currently available techniques only adapt traditional data clustering techniques which are data-centric instead of process-centric into process mining. More generally, designing ad hoc clustering strategies usually leads to more suitable clustering results. Finally, trace clustering techniques only take into account of sorting in the horizontal direction, it would be promising to combine with techniques that focus on abstraction in the vertical direction, such as FuzzyMiner [9] which enforces cartography to the process mining by means of clustering activities.

Acknowledgment. This work is supported in part by the National Natural Science Foundation of China under Grant No. 61672022, Key Disciplines of Computer Science and Technology of Shanghai Polytechnic University under Grant No. XXKZD1604, the Fundamental Research Funds for the Central Universities and Foundation of Graduate Innovation of Shanghai Polytechnic University, and Foundation of Graduate Innovation Center in NUAA under Grant No. kfjj20161601.

References

1. Aalst, W.V.D.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011)
2. Bose, R.P.J.C., van der Aalst, W.M.P.:: Context aware trace clustering: towards improving process mining results. In: SIAM International Conference on Data Mining, pp. 401–412 (2009)
3. Bose, R.P.J.C., van der Aalst, W.M.P.: Trace clustering based on conserved patterns: towards achieving better process models. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 170–181. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_16
4. Bose, R.P.J.C., van der Aalst, W.M.P.: Trace alignment in process mining: opportunities for process diagnostics. In: Hull, R., Mendling, J., Tai, S. (eds.) BPM 2010. LNCS, vol. 6336, pp. 227–242. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15618-2_17
5. de Medeiros, A.K.A., Guzzo, A., Greco, G., van der Aalst, W.M.P., Weijters, A.J.M.M., van Dongen, B.F., Saccà, D.: Process mining based on clustering: a quest for precision. In: ter Hofstede, A., Benatallah, B., Paik, H.-Y. (eds.) BPM 2007. LNCS, vol. 4928, pp. 17–29. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78238-4_4

6. Ferreira, D.R.: Applied sequence clustering techniques for process mining. In: Handbook of Research on Business Process Modeling, pp. 492–513 (2009)
7. García BañUelos, L., Dumas, M., La Rosa, M., De Weerdt, J., Ekanayake, C.C.: Controlled automated discovery of collections of business process models. Inf. Syst. **46**, 85–101 (2014)
8. Greco, G., Guzzo, A., Pontieri, L., Sacca, D.: Discovering expressive process models by clustering log traces. IEEE Trans. Knowl. Data Eng. **18**, 1010–1027 (2006)
9. Günther, C.W., van der Aalst, W.M.P.: Fuzzy mining – adaptive process simplification based on multi-perspective metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 328–343. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_24
10. Hastie, T., Friedman, J., Tibshirani, R.: The Elements of Statistical Learning, vol. 167. Springer, New York (2009). https://doi.org/10.1007/978-0-387-21606-5
11. von Luxbur, U.: A tutorial on spectral clustering. Stat. Comput. **17**, 395–416 (2007)
12. Reijers, H.A., Mendling, J.: A study into the factors that influence the understandability of business process models. Trans. Sys. Man Cyber. Part A **41**, 449–462 (2011)
13. Song, M., Günther, C.W., van der Aalst, W.M.P.: Trace clustering in process mining. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008. LNBIP, vol. 17, pp. 109–120. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00328-8_11
14. Tsai, Y.T.: The constrained common subsequence problem. Inf. Process. Lett. **88**, 173–176 (2003)
15. Weerdt, J.D., Broucke, S.V., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. IEEE Trans. Knowl. Data Eng. **25**, 2708–2720 (2013)