Discovering deviating cases and process variants using trace clustering

Discovering Deviating Cases and Process Variants Using Trace Clustering

B.F.A. Hompes [a] J.C.A.M. Buijs [a] W.M.P. van der Aalst [a]

P.M. Dixit [b] J. Buurman [b]

[a] *Department of Mathematics and Computer Science*
Eindhoven University of Technology, Eindhoven, The Netherlands
[b] *Philips Research, Eindhoven, The Netherlands*

Abstract

Information systems supporting business processes generate event data which provide the starting point for a range of *process mining* techniques. Lion's share of real-life processes are complex and ad-hoc, which creates problems for traditional process mining techniques, that cannot deal with such unstructured processes. Finding mainstream and deviating cases in such data is problematic, since most cases are unique and therefore determining what is normal or exceptional may depend on many factors. *Trace clustering* aims to group similar cases in order to find variations of the process and to gain novel insights into the process at hand. However, few trace clustering techniques take the context of the process into account and focus on the control-flow perspective only. *Outlier detection* techniques provide only a binary distinction between normal and exceptional behavior, or depend on a normative process model to be present. As a result, existing techniques are less suited for processes with a high degree of variability. In this paper, we introduce a novel trace clustering technique that is able to find process variants as well as deviating behavior based on a set of selected perspectives. Evaluation on both artificial and real-life event data reveals that additional insights can consequently be achieved.

1 Introduction

In flexible environments such as the healthcare domain, business process flows typically depend on many factors. As a result, many cases follow a unique path through the process. This variability causes problems for existing process mining techniques. Many assume structured and noise-free input and return spaghetti-like processes and potentially misleading results when there is a lot of variability [4, 15, 25]. Finding common and deviating behavior in such data is problematic, since most cases are unique and what is normal or exceptional may depend on many factors and in practice is often blurred.

Discovering exceptional behavior can lead to possible points for improvement and identify risks. Negatively deviating cases can e.g. lead to an increase in resource utilization and cost, decreased efficiency, or even signify fraudulent behavior. Deviations can also be positive of nature, e.g. when a patient spends less time in the hospital than expected, or when repeated surgery is not necessary where it usually is.

Trace clustering aims to group similar cases in order to find variations of the process. Most trace clustering techniques however are limited to finding more structured process models by finding structurally similar cases. The process context in the sense of case and event data is not taken into account. Additionally, the amount of clusters to be discovered is generally manually determined beforehand. As a result, exceptional cases are often included in mainstream behavior. *Outlier detection* aims to identify cases or events that differ from the mainstream behavior. However, existing outlier detection techniques either assume a clear binary distinction between normal and exceptional behavior or depend on a normative process model to check conformance against. *Thus, potential improvements lie in finding a technique that takes into account the process context in order to find and explain both common and exceptional behavior.* Figure 1a shows the relevance of the problem with a spaghetti-like process model

(a) **Entire event log**: 1143 patients, spaghetti model

(b) **Process variant**: 126 similarly diagnosed patients

(c) **Deviating cases**: 6 patients diagnosed differently

Figure 1: Process models of hospital log, mined with Fuzzy miner. Clustering based on case data.

mined from healthcare data [10]. Figures 1b and 1c show one selected process variant and six deviating cases that where discovered by applying our approach.

We propose a trace clustering technique that groups cases that show similar behavior on a set of selected perspectives. Our technique is based on the Markov cluster (MCL) algorithm, which is a fast and scalable cluster algorithm for graphs [24]. It is able to autonomously discover a (non-specified) number of clusters of different sizes and densities. Hence, the resulting clustering shows both mainstream and deviating behavior. The technique can be used to exploratively analyze the event data at hand in order to gain novel insights into the underlying process(es).

The remainder of this paper is organized as follows. Section 2 discusses related work on trace clustering and outlier detection. Section 3 briefly introduces necessary preliminary definitions. Section 4 describes how the MCL algorithm is used to cluster cases into process variants and deviations. An experimental evaluation on both artificial and real data is performed in Section 5. Section 6 concludes the paper and discusses future work based on the obtained results.

2 Related Work

A comparison of different related techniques is shown in Table 1. Many different *outlier detection* techniques have been proposed and applied in fields as bioinformatics, fraud detection, and networking [3, 17, 28]. In 2006, Yang et al. [27] introduced a process-mining framework for the detection of anomalous behavior such as healthcare fraud and abuse. In [14], statistical properties of the log as well as constraints associated with the process model are taken into account to identify outliers by means of clustering traces. However, in both cases, only the control-flow (CF) perspective is considered and the distinction between normal and deviating behavior is binary. Another means to finding deviating cases is using conformance checking, where an event log is aligned with a process model [1, 2, 20, 21]. Conformance checking, however, is computationally demanding and requires a process model to be present. Furthermore, this model is assumed to be 100% correct. Anomalous behavior might be possible in the model and thus will not be discovered as such using these techniques.

In order to find mainstream behavior, *trace clustering* techniques can be used. Several trace clustering techniques have been proposed in the field of process mining, and an extensive comparative analysis of trace clustering techniques has recently been performed in [23]. Techniques such as [8, 9, 18] aim at somehow optimizing underlying process models according to some measure such as replay fitness or precision. Hence, process context in the sense of case or event data is not considered. For example, De Weerdt et al. [9] employ an active learning inspired trace clustering approach in order to maximize the combined accuracy of underlying process models. Other techniques try to align similar cases [5, 12] or use a probabilistic approach [25] in order to find better process models. Multiple CF patterns are selected by Bose et al. to discover so called trace signatures [7]. Vector space-based approaches such as [4, 6, 16, 19] aim to group cases into homogeneous sets based on trace profiles. MCL has previously been used in [13] where it was used to separate outliers from trace clusters. However, these techniques are also limited to finding more structured process models by grouping structurally similar cases and discovering multiple models for a given event log. Again only the control-flow context is considered. Song et al. [22] include case and event attributes as well in the trace profiles to incorporate process context in trace clustering, but only aim at improving process model accuracy.

Table 1: Comparison of related work. ✓ = supported, ⌣ = not supported, but could be added.

Functionality	[1, 2, 20, 21]	[4, 6, 16, 19]	[5]	[8]	[9]	[13]	[14]	[18]	[22]	[25]	[27]	This paper
Finds CF variants		✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
Includes context				⌣	⌣				✓			✓
Finds CF outliers	✓			✓		✓	✓			✓	✓	✓
Finds context outliers				⌣								✓
Non-binary outlier detection	✓					⌣	✓					✓
No input of no. of clusters	✓			✓		✓	✓	✓			✓	✓
Doesn't require process model	✓	✓	✓			✓			✓	✓	✓	✓

We propose to extend on the ideas and techniques proposed in vector space-based approaches. We incorporate trace clustering with deviation detection by using the MCL algorithm to group cases on multiple perspectives, thereby including the process context. This way we are able to discover both mainstream and deviating behavior, in larger and smaller clusters respectively.

3 Preliminaries

Definition 1 (Event log) *Let $\mathcal{E}$ be the event universe, i.e. the set of all possible event identifiers. Let N be a set of attribute names. $n(e)$ is the value of attribute $n \in N$ for event $e \in \mathcal{E}$. If e does not have an attribute named n, then $n(e) = \perp$ (null value). Let $\mathcal{C}$ be the case universe, i.e. the set of all possible case identifiers. $n(c)$ is the value of attribute $n \in N$ for case $c \in \mathcal{C}$ ($n(c) = \perp$ if c has no attribute named n). Typically, the following attributes are present in all events: $activity(e)$, $time(e)$, $resource(e)$, which are the* activity*,* timestamp*, and* resource *associated to event e. Additional event attributes can be the cost or the outcome of an activity. Cases, like events, have attributes. Each case has a mandatory attribute 'trace': $trace(c) \in \mathcal{E}^*$. $\hat{c} = trace(c)$ is a shorthand notation. A trace is a finite sequence of events $\sigma \in \mathcal{E}^*$, such that each event appears only once, i.e. for $1 \leq i \leq j \leq |\sigma|$: $\sigma(i) \neq \sigma(j)$. For any sequence $\sigma = \langle \sigma_1, \sigma_2, \ldots, \sigma_n \rangle$, $set(\sigma) = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ converts a sequence into a set, e.g. $set(\langle a, b, c, b, c, d \rangle) = \{a, b, c, d\}$. An* event log *is a set of cases $L \subseteq \mathcal{C}$ such that each event appears at most once in the entire log, i.e. for any $c, c' \in L$ such that $c \neq c'$: $set(\hat{c}) \cap set(\hat{c}') = \emptyset$.*

Definition 2 (Perspective) *Let P be a perspective. $\restriction_P : \mathcal{C} \to \mathbb{R}^m$ denotes the function that maps a case to a real vector of length m according to perspective P. m is the number of attributes in P, e.g. the number of different resources in the log. $c\restriction_P$ denotes the projection of case $c \in L$ to a perspective P. Furthermore, we let $c\restriction_{\{P_1, P_2, \ldots, P_K\}} = c\restriction_{P_1} \| c\restriction_{P_2} \| \ldots \| c\restriction_{P_K}$, i.e. the resulting profile vectors from projection to multiple perspectives are concatenated.*

Perspectives can exist of case and event attributes such as the age of a patient, or can be derived values such as the time spent in hospital. By using both control-flow and data perspectives we include the process context in clustering.

Definition 3 (Trace Similarity Matrix) *Let $L \subseteq \mathcal{C}$ be an event log. $\mathcal{M}(L) = (L \times L) \to [0.0, 1.0]$ denotes the set of all possible trace similarity matrices over L. For cases $c, c' \in L$ and a trace similarity matrix $M \in \mathcal{M}(L)$, $M(c, c')$ denotes the similarity between c and c'.*

Definition 4 (Trace Clustering) *Let $L \subseteq \mathcal{C}$ be an event log. A trace cluster over L is a subset of L. A trace clustering $TC \subseteq \mathcal{P}(L)$[1] is a set of trace clusters over L. We assume every case to be part of at least one trace cluster, i.e. $\bigcup TC = L$. Cases can be in multiple clusters, i.e. cluster overlap is allowed.*

4 Discovering Process Variants and Deviating Cases

Our technique combines trace clustering and outlier detection in order to find mainstream and deviating behavior. It relies on the Markov cluster (MCL) algorithm, which has already been successfully applied in different fields, amongst which computational genomics [11]. Here, large sequences of proteins are clustered into protein families. This application of MCL is of particular interest as traces in event logs are sequences of events. A high-level overview of our technique is shown in Figure 2.

[1] $\mathcal{P}(L)$ denotes the powerset over event log L, i.e. all possible sublogs of L.

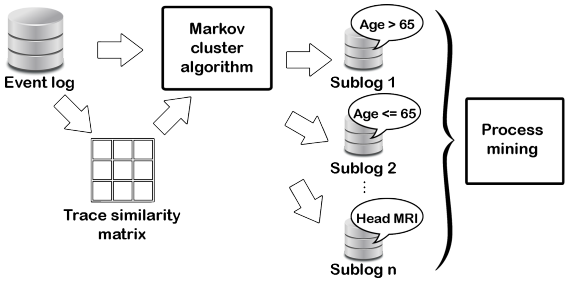


Figure 2: High-level overview of the trace clustering approach.

Contrary to existing techniques we do not label cases as normal or exceptional as in flexible environments the distinction is often not binary. Because MCL is neither biased towards globular or local clusters nor to finding clusters of similar density we can distinguish common and exceptional behavior based on cluster sizes. Unlike in many other techniques, the number of clusters is discovered rather than set beforehand.

MCL uses stochastic matrices representing transition probabilities between nodes in a graph. It alternates an expansion step that raises the matrix to a given power with the inflation step. Inflation raises each element to a given power and normalizes the matrix such that it is stochastic again. As such, there are two parameters to MCL: the expansion and inflation parameter. This alternation eventually results in the separation of the graph into different components which is interpreted as a clustering. The idea is that random walks starting from a node in the graph will frequently visit nodes similar to the starting node with a higher probability than they will dissimilar nodes.

In our case, the MCL algorithm takes as input a left stochastic (column normalized) version of the trace similarity matrix that contains similarities between all cases in an event log. The output is a trace clustering over that log. In order to create a trace similarity matrix, cases in the event log are mapped to a profile vector by projecting them to a selected set of perspectives for which we want to find similar and deviating cases. A pair-wise similarity score is then calculated between these profile vectors. The computation of this initial similarity matrix can be separated from the actual clustering, hence different algorithms can be used for both parts. In this paper we use the cosine similarity, i.e. $M(c, c') = \frac{c\uparrow_P \cdot c'\uparrow_P}{||c\uparrow_P|| \, ||c'\uparrow_P||}, \forall c, c' \in L$, where L is some event log, M a trace similarity matrix for L, and P a set of perspectives. However any measure can be used. Cosine similarity is one of many similarity measures for vectors, and was chosen because of its proven effectiveness, efficient calculation and boundedness to $[0, 1]$. Also, it is able to represent non-binary term weights and allows for partial matching. A typical downside of vector similarity measures is that the order of terms is lost. This problem can be solved by incorporating order in the perspectives, such as the occurrence of frequent patterns.

By selecting the perspectives we can find similarities between cases and pinpoint those cases that on those perspectives either show or deviate from mainstream behavior. This perspective-based clustering therefore takes into account the context of the process. For detailed information on the MCL algorithm we refer to [11, 24].

5 Evaluation

We have evaluated our technique on both a synthetic and a real-life event log. It has been implemented in the process mining tool ProM[2] (version 6.5), and is publicly available in the *TraceClustering* package.

5.1 Synthetic event log

The synthetic event log is used to verify the results with respect to a known ground truth. We generated the log from a fictive, manually created radiology process with 17 unique activities. The control-flow of this process heavily depends on the data-attributes and some tasks can be performed in different orderings, be repeated, or are optional. We distinguish four different scenarios, based on patient age and the part of the body the radiology exam is to be made of. Patients younger than 5 or older than 65 years old are

[2]See `http://promtools.org`

Table 2: Results for trace clustering techniques. We are able to find the underlying reasons for variations in control-flow without a large sacrifice in model quality.

Technique	# clusters	Avg. entropy	Avg. split rate	Avg. FHM Fitness	Med. FHM Fitness	Avg. Replay Fitness MM	Med. Replay Fitness MM	Avg. Replay Fitness ML	Med. Replay Fitness ML
This paper	4	0	1	0,78	0,91	1	1	0,44	0,41
Trace Alignment [5]	4	0,72	2	0,57	0,75	1	1	0,50	0,50
ActiTrac [9]	4	1,09	3,25	0,79	0,93	1	1	0,50	0,50

considered special cases that demand more attention. Head exams require different types of images to be made and need to be checked by specialists. For this process, 1,000 runs were simulated which led to a total of 6,812 events and 164 different control-flow variants in the resulting event log. Each case is labeled with its scenario; 1: special age head exam (52 cases), 2: special age regular exam (422 cases), 3: regular patient head exam (83 cases), and 4: regular patient other exam (443 cases).

Even for this small process, most process discovery algorithms will return an unstructured spaghetti-like model or a flower model when fed the full log. These models allow for either too little behavior or behavior that is not possible (e.g. skipping the exam entirely). Either way, process structure is lost and we are unable to retrieve mainstream and deviating behavior.

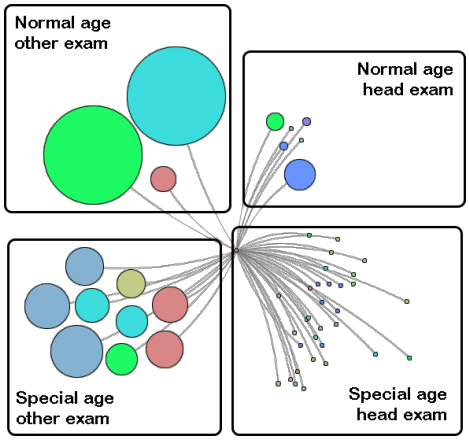
We compare our technique against two of the state of the art approaches discussed in Section 2, namely *Trace Alignment* [5] and *ActiTrac* [9]. For each technique, the event log is projected to each resulting cluster, and a model is mined using the flexible heuristics miner (FHM) [26]. For these models the move on model (MM) and move on log (ML) replay fitness is calculated using cost-based log alignment [2]. We consider two additional metrics for evaluation: *cluster entropy* and *split rate*. Cluster entropy shows the average Shannon entropy for each cluster. The more scenarios are clustered together, the higher the entropy will be. Split rate shows over how many clusters a given scenario is spread. Ideally no clusters contain multiple scenarios and no scenarios are split over clusters, i.e. entropy is 0 and split rate is 1. Note that we cannot use precision and recall measures since it is likely that we discover different clusters, and each case will be included in at least one cluster.

We used a variety of parameter settings and listed the best outcomes for each technique in Table 2. To calculate a similarity matrix in our technique only the occurrence of data attributes ‘age’ and ‘bodypart’ were used, i.e. no control-flow perspectives were considered.

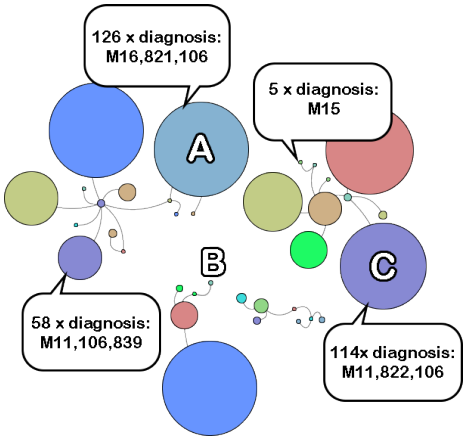
We can see that our technique successfully separates the four scenarios. No scenarios are split over multiple clusters and no clusters contain cases of more than one scenario (entropy equals 0). In both other techniques cases of different scenarios are grouped together and scenarios are split across clusters in order to optimize model quality. However, the fitness of the models mined from clusters discovered by our approach is very close to that of other approaches. This indicates that in the existing, purely control-flow oriented techniques, deviating cases are often grouped together with mainstream behavior, complicating the resulting process models. Also, the underlying reasons for the differences in control-flow are not discovered and therefore the gained insights are limited.

For the four identified scenarios, different control-flow paths are possible. We cannot compare sensitivity to deviating behavior against the other techniques using e.g. different generated noise levels in the log since different aspects are used to cluster upon. As such, in order to evaluate sensitivity we cluster using the occurrence of activities as an additional perspective and describe the effects of deviating behavior on the results of our technique alone.

As can be seen in Figure 3a, many clusters of different sizes are discovered. There are 51 clusters in total. Bigger clusters show more frequent behavior, while smaller clusters show deviating behavior, on the three selected perspectives. Distinctions are made between e.g. cases of the same scenario where tasks are skipped or one optional path is chosen over another. Edges between clusters show a positive similarity between at least one pair of cases. The results show that the process for head exams for patients of special age is much more flexible (and less frequent) than e.g. the process for other exams for regular patients, which seems more standardized. It is clear that although the technique is capable of autonomously discovering the amount of clusters, the choice of perspectives has a large effect on what is discovered, and thus is vital to obtaining new insights. Note that the tool allows for exploring several options easily and quickly.



(a) Synthetic log clustered on age, bodypart and activity.



(b) Hospital log clustered on diagnosis code.

Figure 3: Clustering of event logs.

Table 3: Unique activities per cluster.

(a) Cluster A (126 cases)

Unique activities
maag - ontledig.-scint vast-vloeib.voeds
ovarium - redebulking ovarium carcin
afereseplasma gesplitst fq1 en fq2
buik - stageringslaparotom.-oment
longfunctie - co2 - capnografie
cytologisch onderzoek - ascites
buikoverzicht

(b) Cluster C (114 cases)

Unique activities
lymfeklier - scint sentinel node met pro
lymfeklier - scint sentinel node vervolg
lymfeklier - scint sentinel node dynamis
vulva - ruime lokale excisie van a
vulva - vulvectomie zonder lieskli
vulva - vulvectomie - liesblok
mri bekken

5.2 Real-life event logs

The real-life log is of a Dutch academic hospital and contains cases pertaining to cancer treatment procedures. It was originally used in the first Business Process Intelligence Contest (BPIC 2011) [10]. This log contains 1,143 cases, 150,291 events and 624 distinct activities. There are 981 different control-flow variants. The log contains cases of different stages of malignancy and of different parts of the body. Also, information is present about the diagnosis, treatment, specialism required, patient age, organizational group (hospital department), etc. Each attribute can obtain several different values, leading to a large heterogeneity in the log. For example, there are 38 combinations of values for the 16 attributes 'Diagnosis code' to 'Diagnosis code 15'. Using our technique (parameters expansion=2, inflation=15), clustering on the occurrence of values for these attributes leads to 35 clusters. Each cluster holds cases with different combinations of diagnoses. Cluster sizes vary from 153 to 1, signifying that the event log contains common as well as exceptional diagnoses, as can be seen in Figure 3b.

Figure 1 shows process models for the complete log and for clusters A and B, showing both a process variant and deviating cases respectively. It is clear that the process models for these individual groups of cases are much more readable than the model mined on the complete event log. This indicates that the performed activities differ across diagnoses but are shared between similarly diagnosed patients. This is again demonstrated in Table 3 where some of the (more frequent) activities that are unique to clusters A and C are listed.

Other clusterings are also possible. For instance, taking the attributes 'Treatment code' to 'Treatment code 15' as selected perspectives leads to a total of 82 clusters, of which only 7 clusters contain more than 20 cases, and most contain less than 5. Clustering on the combined perspective of diagnosis and treatment returns 62 clusters of which 13 contain more than 20 cases.

This indicates that for some diagnoses standard treatment procedures are used, while for other diagnoses the treatment is decided upon case-by-case. It may be of interest to the process owners to

further look into both. Insights such as these can be gained easily and can be used e.g. to specify protocols, find cases related to a complex running case or for auditing purposes. It also shows that our technique can be used to quickly explore the data.

6 Conclusions and Future Work

In this paper we introduced a trace clustering technique based on the Markov cluster (MCL) algorithm that is able to find variations and deviations of a process based on a set of selected perspectives. By clustering on multiple perspectives we can find cases that on those perspectives show either mainstream or deviating behavior, and by using both control-flow and case data we take into account the process context. This information can then be used to create more structured process models and to improve the underlying process(es) based on gained insights. Positively deviating cases can be used in an attempt to improve the process for a larger group of cases. Insights gained from negatively deviating cases can lead to the ability of earlier detection of those cases or to process modifications preventing such unwanted behavior in the future.

While the MCL algorithm is non-parametric in the number of clusters, it requires the expansion and inflation parameter to be set manually. In the future, ways to predict meaningful values could be looked in to. Also, potential improvements lie in automatically finding perspectives that have discriminative power, i.e. those perspectives that when used, will lead to a non-trivial clustering. Other ways of computing case-by-case similarities can be looked into. Another direction of research is evolutionary (partial) trace clustering. Due to concept drift processes change over time. For example, at a certain time, treatments for a certain diagnosis might change or patients might have similar treatment up to a certain point after which their process starts to differ and vice-versa. Ultimately, we would like to classify deviating behavior as either positive or negative, and be able to suggest process improvements based on gained insights.

References

[1] W.M.P. van der Aalst, K.M. van Hee, J.M. van Werf, and M. Verdonk. Auditing 2.0: Using process mining to support tomorrow's auditor. *Computer*, 43(3):90–93, 2010.

[2] A. Adriansyah, B.F. van Dongen, and W.M.P. van der Aalst. Conformance checking using cost-based fitness analysis. In *Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International*, pages 55–64. IEEE, 2011.

[3] C.C. Aggarwal and P.S. Yu. Outlier detection for high dimensional data. In *ACM Sigmod Record*, volume 30, pages 37–46. ACM, 2001.

[4] R.P.J.C. Bose and W.M.P. van der Aalst. Context aware trace clustering: Towards improving process mining results. In *Proceedings of the SIAM International Conference on Data Mining*, pages 401–412. Society for Industrial and Applied Mathematics, 2009.

[5] R.P.J.C. Bose and W.M.P. van der Aalst. Trace alignment in process mining: opportunities for process diagnostics. In *Business Process Management*, pages 227–242. Springer, 2010.

[6] R.P.J.C. Bose and W.M.P. van der Aalst. Trace clustering based on conserved patterns: Towards achieving better process models. In *Business Process Management Workshops*, pages 170–181. Springer, 2010.

[7] R.P.J.C. Bose and W.M.P. van der Aalst. Discovering signature patterns from event logs. In *Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on*, pages 111–118. IEEE, 2013.

[8] A.K.A. De Medeiros, A. Guzzo, G. Greco, W.M.P. van der Aalst, A.J.M.M. Weijters, B.F. van Dongen, and D. Saccà. Process mining based on clustering: A quest for precision. In *Business Process Management Workshops*, pages 17–29. Springer, 2008.

[9] J. De Weerdt, J. Vanthienen, B. Baesens, et al. Active trace clustering for improved process discovery. *Knowledge and Data Engineering, IEEE Transactions on*, 25(12):2708–2720, 2013.

[10] B.F. van Dongen. Real-life event logs - hospital log, 2011. URL: http://dx.doi.org/10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54.

[11] A.J. Enright, S. Van Dongen, and C.A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic acids research*, 30(7):1575–1584, 2002.

[12] J. Evermann, T. Thaler, and P. Fettke. Clustering traces using sequence alignment. In *Proceedings of the 11th International Workshop on Business Process Intelligence,. International Workshop on Business Process Intelligence (BPI-15), located at International Conference on Business Process Management, July 31 - August 3, Innsbruck, Austria*, 2015.

[13] F. Folino, G. Greco, A. Guzzo, and L. Pontieri. Mining usage scenarios in business processes: Outlier-aware discovery and run-time prediction. *Data & Knowledge Engineering*, 70(12):1005–1029, 2011.

[14] L. Ghionna, G. Greco, A. Guzzo, and L. Pontieri. Outlier detection techniques for process mining applications. In *Foundations of Intelligent Systems*, pages 150–159. Springer, 2008.

[15] S. Goedertier, J. De Weerdt, D. Martens, J. Vanthienen, and B. Baesens. Process discovery in event logs: An application in the telecom industry. *Applied Soft Computing*, 11(2):1697–1710, 2011.

[16] G. Greco, A. Guzzo, L. Ponieri, and D. Sacca. Discovering expressive process models by clustering log traces. *Knowledge and Data Engineering, IEEE Transactions on*, 18(8):1010–1027, 2006.

[17] V.J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[18] C. Li, M. Reichert, and A. Wombacher. Discovering process reference models from process variants using clustering techniques. 2008.

[19] D. Luengo and M. Sepúlveda. Applying clustering in process mining to find different versions of a business process that changes over time. In *Business Process Management Workshops*, pages 153–158. Springer, 2012.

[20] J. Muñoz-Gama, J. Carmona, and W.M.P. van der Aalst. Hierarchical Conformance Checking of Process Models Based on Event Logs. In *Petri Nets*, pages 291–310, 2013.

[21] A. Rozinat and W.M.P. van der Aalst. Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1):64–95, 2008.

[22] M. Song, C.W. Günther, and W.M.P. van der Aalst. Trace clustering in process mining. In *Business Process Management Workshops*, pages 109–120. Springer, 2009.

[23] T. Thaler, S.F. Ternis, P. Fettke, and P. Loos. A comparative analysis of process instance cluster techniques. In *Proceedings of the 12th International Conference on Wirtschaftsinformatik. Internationale Tagung Wirtschaftsinformatik (WI-15), March 3-5, Osnabrck, Germany*. Universitt Osnabrck, 3 2015.

[24] S. Van Dongen. A cluster algorithm for graphs. *Report-Information systems*, (10):1–40, 2000.

[25] G.M. Veiga and D.R. Ferreira. Understanding Spaghetti Models with Sequence Clustering for ProM. In *Business Process Management Workshops*, pages 92–103. Springer, 2010.

[26] A.J.M.M. Weijters and J.T.S. Ribeiro. Flexible heuristics miner (FHM). In *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 310–317. IEEE, 2011.

[27] W. Yang and S. Hwang. A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications*, 31(1):56–68, 2006.

[28] Y. Zhang, N. Meratnia, and P. Havinga. Outlier detection techniques for wireless sensor networks: A survey. *Communications Surveys & Tutorials, IEEE*, 12(2):159–170, 2010.