Unveiling Hidden Patterns in Flexible Medical Treatment Processes – A Process Mining Case Study

Kathrin Kirchner[1(✉)] and Petar Marković[2]

[1] Berlin School of Economics and Law, 10315 Berlin, Germany
kathrin.kirchner@hwr-berlin.de
[2] Technische Universiteit Eindhoven, 5600 Eindhoven, The Netherlands
p.markovic@tue.nl

Abstract. In hospital environments, treatment processes, resp. clinical pathways, are adopted based on the health state of a patient. Modeling of pathways is time consuming and due to the involvement of many participants, the introduction of clinical pathways is cost-intensive. Process mining offers a possibility for automatic or semi-automatic creation of clinical pathways based on the event log data recorded during the process execution in hospital information systems. However, state-of-the-art algorithms struggle to discover meaningful end-to-end patterns from highly flexible clinical log data. This challenge can be addressed by Local Process Models. They allow pathways to be modeled partially, thus enabling the detection of major process steps. In our case study, we apply this recently proposed method on a real world clinical dataset and discuss results and challenges.

Keywords: Process mining · Local process model · Clinical pathway
Flexible process · Case study

1 Introduction

Healthcare faces the challenge to deliver high treatment quality and patient satisfaction while being cost efficient. Clinical pathways as structured, multidisciplinary care plans standardize treatment processes. They define the steps of patient care for a certain disease in a specific hospital [16]. Compared to processes in industry, clinical pathways are more flexible as a treatment process varies for each individual patient. Additional therapies might be necessary and the sequence of treatment steps might change due to interpreting patient-specific data. Thus, deviations from pre-planned treatment processes are frequent, and their effect on the process is much stronger. This is especially evident in certain medical domains, such as organ transplantation, where treatment processes can last long and the unstructuredness is high.

Data about the treatment of a patient is collected in different clinical information systems in a hospital. Treatment steps (events) can be recorded based

F. Dargam et al. (Eds.): ICDSST 2018, LNBIP 313, pp. 169–180, 2018.
https://doi.org/10.1007/978-3-319-90315-6_14

on timestamps and each event refers to a particular patient and a particular activity. More information might be available, for example, the performer of the event, the timestamp or documents or treatment information recorded with the event. This data can be collected in an event log. The idea of process mining is to discover, monitor and improve real processes based on the data in event logs. [22] In this paper the focus is on process discovery where no a-priori process schema exists. Based on an event log, a process mining algorithm constructs a schema of the process.

Traditionally, the majority of the state-of-the-art process mining algorithms search for complete, so called "start-to-end" processes. However, in highly flexible settings, such as the one this paper addresses, they fail to achieve proper results, or appear to have limited effectiveness. This fact is supported by the previous study of [7], where the robust trace clustering approach of [3] was only partial successfully applied on living donor liver transplantation data, the same dataset this paper is referring to. Although one cluster with patients following a typical treatment process could be found, the methodology was not able to find useful pathways for the remaining untypical data. The case study faced challenges because of coarse timestamps, small size of sample, and different lengths of the pathway for different patients.

A recent approach, Local Process Models (LPM) Discovery, is trying to answer these challenges. It is focused on the mining of a set of process models where each model describes the behavior represented in the event log only partially, i.e., local process models are created [20]. This can be especially helpful if no overall process model exists that describes the traces of the process from the start of a process to the end.

In this paper, we use LPMs for discovering short process sequences in a case study of living liver donor transplantation. These sequences represent interpretable derived sub-processes, potentially useful for building models on a higher level of abstraction. The paper is structured as follows: The next chapter discusses shortly the related work. Section 3 explains the used data and methodology. Section 4 shows our results, and a summary and outlook is given in Sect. 5.

2 Selected Related Work

2.1 Process Mining in Healthcare

Process mining follows the aim to extract novel patterns from event data. Process discovery as a part of process mining focuses on discovering a process model for the set of event log sequences from start to end of a process [22]. In many cases, events recorded in the event log are too fine-grained, causing process discovery algorithms to discover incomprehensible process models or process models that are not representative for the event log [19]. This might lead to an uninterpretable mess of nodes and connections in the process model - a so called spaghetti model.

A literature review of the application of process mining in healthcare is given by [15]. It was found that the most commonly used techniques are Trace Clustering [17], Fuzzy Miner [4] and Heuristics Miner [25], because they can manage

noise and incompleteness, and allow models to be identified for less-structured processes. Furthermore, they allow similar cases to be grouped together, as is the case of trace clustering. Process Mining was applied in 22 different medical fields with oncology and surgery being the most prominent ones. As an example, [14] applied heuristics miner on log data from dentistry and described difficulties to handle flexibilities in the clinical pathways. [6] List challenges for process mining in healthcare, among them complexity and flexibility of treatment processes and that not all treatment steps are fully recorded in a clinical information system.

2.2 Local Process Models

To overcome the problem of obtaining spaghetti processes, a group of techniques exists that is able to reveal stronger patterns of local character (i.e. smaller pieces of traces consisting only of neighboring activities) in the process. Contrary to global approaches (e.g., [10]), these algorithms provide useful insights into the process structure, since these detected partial patterns could be observed as structured steps/parts of the flexible processes on a higher level of abstraction. These techniques are quite alike to frequent sequence mining [1] and use a similar logic.

One of the techniques identified as especially appropriate for addressing the problem is the novel Local Process Models methodology proposed by [20]. Its nature is similar to the very mature Episode mining [9], yet, with many additional advantages presented in Table 1. In this table we compare LPM and Episode Mining with general global process mining techniques.

For our case study, we selected the LPM approach, because it provides the ability to discover processes that are formal and include all possible kinds of behavior (sequences, concurrency, choices, loops). Furthermore, the existence of a transparent, highly customizable set of quality measures was the key reason for evaluating LPM on our data. Due to space restrictions, only a brief formal introduction of LPMs is given, whilst for the in-depth explanation of the method, readers should refer to [18] or [20].

Formally, Local Process Models (LPMs in the further text) are Process Trees [2] of size (i.e. number of leaf nodes) between 2 and 5, built of frequently re-occurring activities originating from event log subtraces. Process trees consist of leaf nodes representing activities, and non-leaf nodes representing relation operators between activities:

- $\rightarrow$ sequential execution: second child executed after first,
- $\wedge$ parallel execution: first and second child are both executed in any order,
- $\times$ exclusive choice: either first or second child is executed,
- $\circlearrowleft$ loop: after the execution of the first, *"do"* child, the second, *"re-do"* child, followed again by "do" child, could be executed afterward, minimally once.

The composition of two process trees is also a process tree. For example, given a *language* (i.e. the set of allowed activity execution paths) of the example model, $\mathcal{L}(M) = \{\langle b, c, a\rangle, \langle b, a, c\rangle, \langle d, a, c\rangle, \langle d, c, a\rangle\}$, a resulting process tree M has a

Table 1. Characteristics of LPM, Episode Mining and global techniques

Local process models [20]	Episode miner [9]	Global techniques, e.g. [10]
Advantages		
Able to mine frequent process steps in flexible logs (local scope)	Able to mine frequent process steps in flexible logs (local scope)	Large number of mature, scalable and robust techniques to choose from
Able to capture all of process behavior (sequential, parallel, loops and exclusive choice flow)	Computationally efficient for larger event logs	
Formal process models based on process trees, thus sound by design	Informal process models, thus unstable results	
Represented as Petri nets, thus easily convertible to other notations		
Large number of adjustable quality metrics		
Disadvantages		
Computationally inefficient (long running times for large sized logs w.r.t. number of distinct activities), but has heuristics for speed-up	Limited type of process behavior can be modeled (only sequential and parallel flow, no loops or exclusive choice)	Observes process as a whole, from "Start-to-end" without possibility to focus on relatively stable process parts (Global scope)
		Limited quality metrics
		Proven extremely low performance on flexible processes
		Many of the techniques are computationally over-complex

form of $\rightarrow (\times(b,d), \wedge(a,c))$, as first b or d is executed, followed by a and c, in any order.

In LPMs, underlying process tree models are represented as Accepting Petri-nets $APN = (LPN, M_o, MF)$, that is, labeled Petri-nets with an initial marking $M_o \in \mathbb{N}^p$ and a set of possible final markings $MF \subseteq \mathbb{N}^p$, such that $\forall_{M_1,M_2 \in MF}\ M_1 \nsubseteq M_2$, where a marking represents a state of a Petri net (i.e. distribution of tokens over its places). A labeled Petri net is defined as a bipartite graph, i.e. a tuple $LPN = \langle P, T, F, \Sigma_M, \ell \rangle$ consisting of a set of places P, a set of transitions (activities) T such that $P \cap T = \emptyset$, and arcs (flow-relation $F = (P \times T)(T \times P)$). The set of labels representing the names of activities is Σ_M (with invisible events $\tau \notin \Sigma_M$), and the labeling function $\ell : T \rightarrow \Sigma_M \cup \{\tau\}$).

The intention of the approach is to partition the event log L, consisting of the traces $\sigma = \langle e_1, e_2, \ldots, e_n \rangle \in L$ (sequences of events) in the way that the number of events that would fit the language-fitting LPM subtrace sequences is maximized; i.e. a segmentation of a LPM trace σ: $\lambda(LPM) = \alpha_0\beta_1\alpha_1\beta_1 \ldots \beta_n\alpha_n$ where $\beta_j \in \mathcal{L}(LPM)$ represents a language-fitting subsequence, whilst $\alpha_j \notin \mathcal{L}(LPM)$ is a non-fitting subsequence, such that the number of events in $\{\beta_1, \beta_2, \ldots, \beta_n\}$ is maximized [20]. E.g., for a given trace $\sigma = \langle b, a, c, d, a, d, b, d, c, a, b, c, b \rangle$ projected on activity set $\{b, c, d\}$, an optimal partition, so that fitting subtraces are $\beta_1 = \langle b, c, d \rangle$ and $\beta_2 = \langle d, b, c \rangle$ (with corresponding non-fitting subtraces $\alpha_0 = \langle\rangle, \alpha_1 = \langle d \rangle$ and $\alpha_3 = \langle b, c, b \rangle$) results in a projected trace $\sigma' = \langle b, c, d, d, b, c \rangle$.

LPM discovery has four repetitive steps. In the first step, a pool of models fitting to the partitioning is *generated*, which are then *evaluated*, and the best ones *selected* and kept, whilst the others are disregarded. In the final fourth step, the best candidate models are *expanded*, i.e. their activity nodes are replaced by operator nodes whose children are the replaced activity node and another node from the language. The cycle is repeated until there is no tree in the candidate set that meets the desired threshold. During the detection, in the evaluation step, LPMs are being evaluated using the weighted average of the following quality measures [21] (formulas omitted due to space restrictions):

1. *Support* - the frequency of the detected LPM in the log (number of detected fragments in the log). The higher the support, the stronger the model.
2. *Confidence* - the percent of the events of the activities in the log that abide to LPM. The higher the confidence, the stronger the model.
3. *Language fit* - ratio of the behavior allowed by the LPM that is observed in the event log. LPMs that allow more behavior than it is observed have tendency to overgeneralize.
4. *Determinism* - the average number of enabled transitions in each marking of the model that was reached while replaying the event log. LPM that has higher non-determinism is giving less information on ordering of the events (i.e., models with high level of non-determinism are so called "flower-models"). The higher the value, the better the model.
5. *Coverage* - The frequency of activities described in the LPM in the event log. The higher the value, the better the model.

These metrics represent local equivalents of global quality metrics defined in [23]. For different types of problems, different measures (and their corresponding weights) can be preferred in order to detect models of the best quality in terms of "local" fitness, precision and generalization (while simplicity is controlled with input size of the model), in a particular case. Since this is a domain-related aspect, it will be investigated in more detail in the results section later. Usually, a trade-off between quality dimensions is needed, since a perfect model does not exist. However, in our study, besides objectively good models, we also prefer the ones with the highest level of interpretability.

In order to overcome high computational complexity of the full-sized generation of subsets of activities in the a-priori-like manner, and enable the analysis of the larger logs, [21] introduced a set of three heuristics (Markov clustering, Log entropy, and Maximal relative information gain) which, in addition to the execution speedup, simultaneously allow mining of higher quality models through log projections. The authors have reported that the selection of an appropriate heuristic is highly dependent on the specific log and domain.

Finally, the most promising characteristic in the LPM methodology is its ability to use the detected patterns for generating the abstractions of process steps as shown in [19]. Thus, we applied this approach on our data set. We argue that this way, we are able to detect relatively stable subprocesses in the flexible process. These low level patterns can be abstracted to a high-level log [12] which can be mined using global process mining techniques [13].

3 Data and Method

During the research project PIGE (Process Intelligence in Healthcare), a clinical pathway for living liver donors was modeled in BPMN together with medical personnel [5]. The process can be roughly described as follows (BPMN diagram in Fig. 1): A healthy person can donate a part of her/his liver to a near relative. First, medical doctors have a first talk and investigation with all possible donors for a certain patient. The preselected person, before he/she can become a living donor, has to undergo an evaluation procedure to ensure that the individual is physically fit. Computer tomography (CT) scans or magnetic resonance tomography (MRT) are done to image the liver. The pre-examinations are predetermined, but can change in the sequence depending on the availability of necessary resources. During and after operation, complications can occur that lead to additional interventions or even an additional operation.

Fig. 1. BPMN process model of living liver donor (high level)

Besides of the high-level process shown in Fig. 1, 10 subprocesses were modeled on three more detailed levels. Flexibility in process execution was expressed using, e.g., annotations with a list of possible steps or documents named *checklist*. [8] Especially for the *evaluation of potential donor* (Fig. 1, a checklist is used that lists all mandatory and possible additional investigations for a potential donor. The sequence of these investigations are planned by the medical personnel due to availability of resources like medical experts and tools. The interesting question occurs whether hidden patterns in the treatment execution data can be unveiled that can give interesting insights in the treatment process: Which treatment steps are more often conducted in a direct sequence than others? This might be helpful in a hospital to optimize the treatment procedures, which is in our case especially the pre-evaluation subprocess before the operation.

The data set was extracted from a clinical information system during the PIGE project. All patient data which were marked as living liver donors in a time period of 3 years were selected and anonymized. The resulting data set contained 50 living liver donors with all together 331 events. Not all patients went through all process steps. If the pre-examination found the person not suitable for donating the liver, an operation is not done. Therefore, the number of process steps for patients was different. Patients that were already in a later process step in the considered time period were also in the data set. Thus, not all pathways had the same start- and endpoint. Furthermore, the timestamps for all events were only dates (no concrete time was available), and thus, several events were executed on the same day (Table 2). The event log file consists of

treatment steps on a higher level of detail (e.g., CT as a part of the evaluation procedure).

Table 2. Event log file for living liver donors

Patient-ID	OPS-code	Treatment	Treatment day	Admission day	Discharge day
12345678	3-225	CT:Abdomen	10.10.2014	10.10.2014	12.10.2014
12345678	3-226	CT:Pelvis	10.10.2014	10.10.2014	12.10.2014
...					
23456789	3-225	CT:Abdomen	08.02.2015	08.02.2015	10.02.2015

The average trace has length of approximately 7 events. There are, on average, 2.07 ± 1.52 activities per day. 32 distinct activities can be found in the log. Figure 2 depicts the relative distributions of recorded activities per day. The figure straightforwardly show a negligible negative effect of timestamp granularity to the future conclusions, since the majority (cca 72%) of the population has no more than two activities executed on the same day.

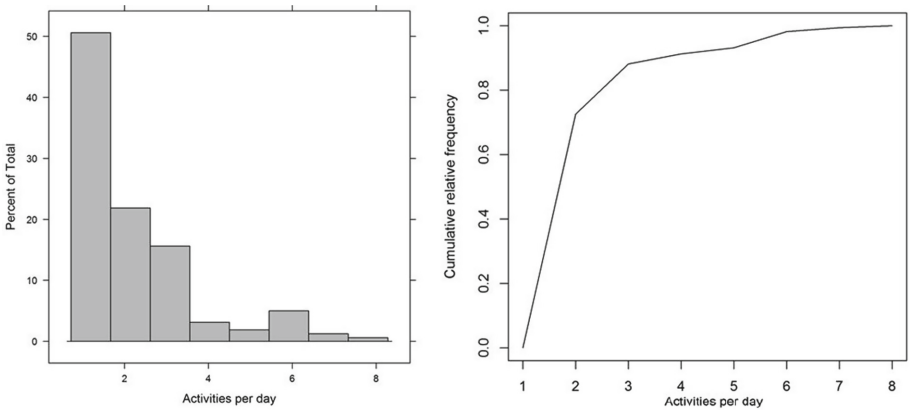

Fig. 2. Relative and cumulative distribution of activities per day - donors

Taking all previously mentioned into account, the methodology we used in this paper, consisted of the following four steps:

1. Mine for k-sized Local Process Model (where k can take values from 2 to 5) with non-restrictive settings of a-priori thresholds
2. Examine the obtained models and select the best ones, according to the domain knowledge and quality measures weights most appropriate for the purpose.

3. Repeat steps 1–2 with application of heuristics
4. When the best LPMs are known, use them for inducing the higher level of abstraction on the non-flexible process steps, so that the process can be analyzed from the higher level.

4 Results

In order to find local process models, we used ProM process mining software [24] with the LPM algorithm plugin. We intended to mine for LPMs of sizes fitting the full possible range (i.e. models consisting of 2, 3, 4, and 5 activities). However, the algorithm was able to detect only one model consisting of more than 4 tasks. Altogether, we calculated 60 different models using the following weights for support: 0.1, language fit: 0.1, confidence: 0.4, coverage: 0, determinism: 0.3 and average number of firings: 0.1. For finding LPMs, the traditional approach (no heuristics used) was applied. We found 18 models with 2 tasks, 27 with 3 tasks and 14 comprising 4 tasks. Figure 3 gives examples of five mined LPMs from the living liver donors.

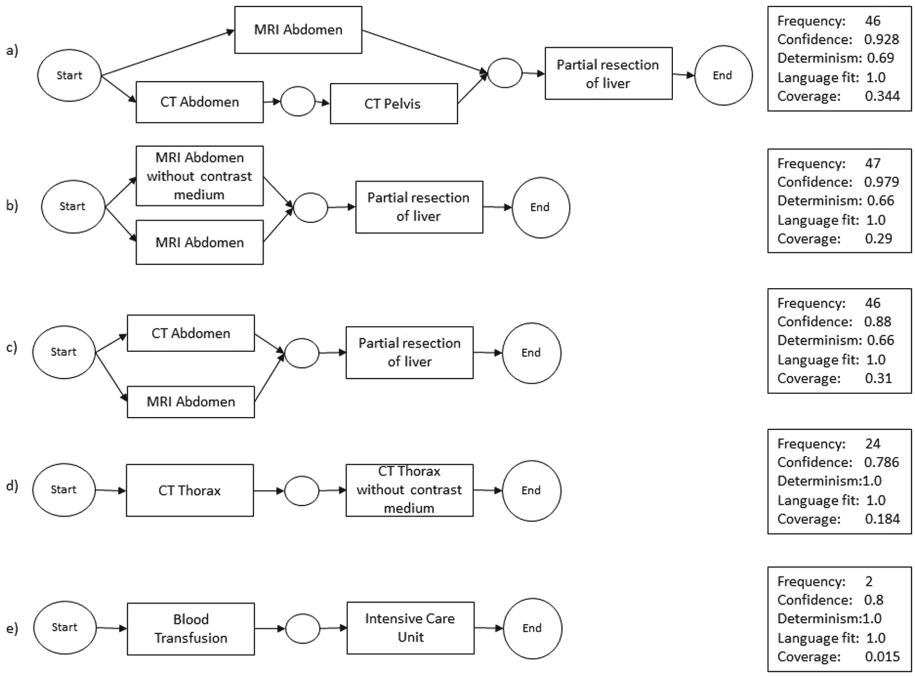

Fig. 3. Examples of mined local process models

Figure 3(a) shows the only one relevant model with four process tasks, that was valid for 46 cases. All other 13 mined process models with four tasks were

only valid for 2 patients. In the visualized model, the patient undergoes an evaluation before the liver operation. In the evaluation, most patients get a MRI, while some get alternatively a computer tomography of abdomen and pelvis. Figure 3(b) and (c) show also evaluation steps before the operation can be done. In (d) two CTs are executed. The first CT does not provide useful results, so a second CT using contrast medium is done. In (e), that is only valid for two patients, the post-operation phase is visible: First, the patient gets a blood transfusion, before he is brought to intensive care unit.

The mined local process models define high-level tasks such as: ≪Evaluation≫, ≪Operation≫ or ≪Post-operative Phase≫, which are useful for raising the level of abstraction of the process, and hence, its interpretability (see also Fig. 1):

- Figure 3a–c: Evaluation of a potential donor → Operation of a living liver donor
- Figure 3d: Evaluation of a potential donor
- Figure 3e: Operation of a living liver donor → Post-Operative treatment in the hospital

The 60 mined local models have different frequencies. In the case of models containing 2 steps (like in Fig. 3e), we found frequent patterns with frequency 44 or 37 (total number of patients: 50), but also two infrequent patterns that only occurred for 2 patients. In case of patterns that were 3 steps long, we found 10 patterns that are valid for 44 out of 50 patients.

Our case study faced some challenges, because of the coarse timestamps (date instead of date and time), the small size of sample (only 50 patients), and the large deviation of the pathway length. Nevertheless, we could identify interesting local process models. Even if the frequency of a model is low, it might provide interesting insights for medical personnel (e.g., handling complications during operations). An additional issue was that not all process steps were recorded in the clinical information system, e.g., the step *admission to hospital for operation*.

In addition to the data set used in this case study, we investigated another event log of liver transplantation patients, consisting of 2294 events referring to 256 cases in the period of 586 recorded days. The average trace consists of approximately 9 events. On average, there are 3.91 ± 3.08 activities per day (Fig. 4). In terms of the distribution of activities per day, this log is more complex and sensitive to the timestamp granularity issue than the living liver donor data, since less than 50% of the patients has up to two activities executed on the same day. Nevertheless, due to the highly skewed distribution (indicated with high standard deviation) and the nature of the LPM method, it could be presumed that the negative effects would be minimal. Unfortunately, in this case, there are 234 distinct activities, which is currently too large and complex to be analyzed due to computational limitations of the LPM methodology. Thus, we applied all the heuristics proposed in [21] in addition to the traditional LPM approach. However, the results showed that the log size limitations for application of these heuristics (even the most advanced Markov clustering one), still were unable to overcome this issue.

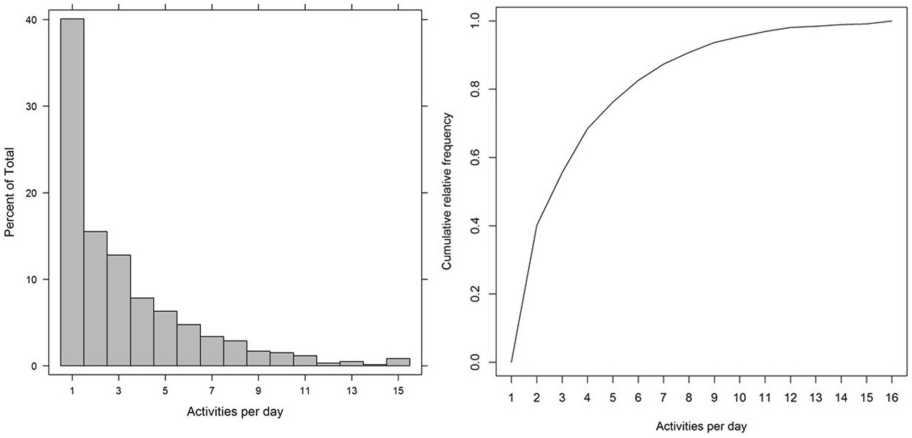

Fig. 4. Relative and cumulative distribution of activities per day - liver transplantation patients

Regardless of the faced issues, it should be stressed that the results bring valuable insights, as to the best of the authors' knowledge, there exists only one paper where LPMs were applied on medical data [13] until now. In this paper, a large pool of LPMs used for abstracting event logs is generated in automatic fashion. However, interpretability and usefulness of generated models is not discussed, nor the models and their corresponding settings are presented, as the main aim of the paper is overall benchmark assessment of the robustness of the approach, which is solely based on aggregated quality metrics discussion (i.e. F-score values change compared to the non-abstracted model), and not the analysis of the underlying medical process. In addition, LPMs were mined on a significantly less flexible and larger event log describing patients with SEPSIS [11], originating from hospital ERP system, consisting of approx. 1000 cases, 15000 events, and only 16 distinct activities. The log was already successfully mined using end-to-end approaches beforehand, thus although the algorithm possibly found interesting process steps, it is debatable whether the discovered knowledge yielded any new previously unknown valuable insights about the process for decision makers.

5 Summary and Outlook

In this paper, we provided a case study on how flexible event logs from medical domain can be successfully mined and analyzed within a local scope using LPMs. Contrary to the complicated end-to-end spaghetti models, the local process models allow for better understanding of major process steps, thus improving communication of results with medical personnel. Due to their simple structure and high interpretability, they allow the validation of the results using the expert knowledge, which is of utter importance for the specific domain. The naming of the local process models has to be done by a medical expert. Discovered local

process models are used for raising the level of abstraction into a single subprocess step. Thus, the flexible parts of the treatment process are explained using local process models.

Although the LPM methodology provided good results on the living liver donor data, it was not possible to use it on the larger data set of liver transplantation patients. Thus, in the future, a smart preprocessing of the liver transplantation patients' dataset is intended in order to enable execution of LPM miner on it. Since there exist a large number of activities which are recorded on the level of granularity lower than needed (e.g. different types of CTs are recorded separately), the log size could be reduced without large loss of information (i.e. it is important to know that CT activity is done, whilst it is not important to know which specific type was applied). Successful implementation of this improvement could lead to more convincing and stable findings.

Acknowledgment. The authors kindly thank Niek Tax, Ph.D. candidate at TU Eindhoven, the author of the LPM methodology, for his valuable help and advice on installing and running the novel LPM mining ProM nightly build plug-in, which was crucial for conducting the experimental part of this paper.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the 11th International Conference on Data Engineering, pp. 3–14. IEEE (1995)
2. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: A genetic algorithm for discovering process trees. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2012)
3. Delias, P., Doumpos, M., Grigoroudis, E., Manolitzas, P., Matsatsinis, N.: Supporting healthcare management decisions via robust clustering of event logs. Knowl. -Based Syst. **84**, 203–213 (2015)
4. 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
5. Herzberg, N., Kirchner, K., Weske, M.: Modeling and monitoring variability in hospital treatments: a scenario using CMMN. In: Fournier, F., Mendling, J. (eds.) BPM 2014. LNBIP, vol. 202, pp. 3–15. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15895-2_1
6. Homayounfar, P.: Process mining challenges in hospital information systems. In: Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1135–1140. IEEE (2012)
7. Kirchner, K., Marković, P., Delias, P.: Challenges in automatic creation of clinical pathways: a case study. In: Proceeding of XV International Symposium SYMORG 2016, pp. 188–194 (2016)
8. Laue, R., Kirchner, K.: Using patterns for communicating about flexible processes. In: Conference on Business Process Modeling, Development, and Support (BPMDS), pp. 12–19. CEUR (2017)
9. Leemans, M., van der Aalst, W.M.P.: Discovery of frequent episodes in event logs. In: Ceravolo, P., Russo, B., Accorsi, R. (eds.) SIMPDA 2014. LNBIP, vol. 237, pp. 1–31. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27243-6_1

10. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs - a constructive approach. In: Colom, J.-M., Desel, J. (eds.) PETRI NETS 2013. LNCS, vol. 7927, pp. 311–329. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38697-8_17
11. Mannhardt, F.: Sepsis cases - event log. Eindhoven University of Technology (2016). https://doi.org/10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460
12. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: From low-level events to activities - a pattern-based approach. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 125–141. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_8
13. Mannhardt, F., Tax, N.: Unsupervised event abstraction using pattern abstraction and local process models. In: Proceedings of the Business Process Modeling, Development, and Support Working Conference 2017 (BPMDS) (2017)
14. Mans, R., Reijers, H., van Genuchten, M., Wismeijer, D.: Mining processes in dentistry. In: Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, pp. 379–388. ACM (2012)
15. Rojas, E., Munoz-Gama, J., Sepúlveda, M., Capurro, D.: Process mining in healthcare: a literature review. J. Biomed. Inform. **61**, 224–236 (2016)
16. Rotter, T., Kinsman, L., James, E., Machotta, A., Willis, J., Snow, P., Kugler, J.: The effects of clinical pathways on professional practice, patient outcomes, length of stay, and hospital costs: Cochrane systematic review and meta-analysis. Eval. Health Prof. **35**(1), 3–27 (2012)
17. 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
18. Tax, N., Genga, L., Zannone, N.: On the use of hierarchical subtrace mining for efficient local process model mining. In: Proceedings of the 7th International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA 2017), pp. 8–22 (2017)
19. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Event abstraction for process mining using supervised learning techniques. In: SAI Intelligent Systems Conference 2016, pp. 1–10. IEEE (2016)
20. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Mining local process models. J. Innov. Digit. Ecosyst. **3**(2), 183–196 (2016)
21. Tax, N., Sidorova, N., van der Aalst, W.M.P., Haakma, R.: Heuristic approaches for generating local process models through log projections. In: IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8. IEEE (2016)
22. van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
23. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdisc. Rev.: Data Min. Knowl. Discov. **2**(2), 182–192 (2012)
24. Verbeek, H.M.W., Buijs, J.C.A.M., Van Dongen, B.F., van der Aalst, W.M.P.: ProM 6: the process mining toolkit. In: Proceedings of the BPM Demonstration Track, vol. 615, pp. 34–39 (2010)
25. Weijters, A.J.M.M., van der Aalst, W.M.P., De Medeiros, A.K.A.: Process mining with the heuristics miner-algorithm. Technical Report WP, Technische Universiteit Eindhoven, vol. 166, pp. 1–34 (2006)