On the Comparison of Markov Chains-based Models in Process Mining for Healthcare: A Case Study

Mauro Vallati
University of Huddersfield
Huddersfield, UK
m.vallati@hud.ac.uk

Stefania Orini
Università degli Studi di Brescia, IT.
Istituto Centro San Giovanni Di Dio
Fatebenefratelli, Brescia, IT
steorini@gmail.com

Mariagrazia Lorusso
Università degli Studi di Brescia
Brescia, IT
mariagrazia.lorusso@unibs.it

Mariachiara Savino
Università Cattolica del Sacro Cuore
Rome, IT
mariachiara.savino@unicatt.it

Roberto Gatta
Università degli Studi di Brescia
Brescia, IT
roberto.gatta@unibs.it

Massimiliano Filosto
Università degli Studi di Brescia
Brescia, IT
massimiliano.filosto@unibs.it

Abstract

In the last decade, Process Mining has become a significant field to help healthcare process experts understand and gain relevant insights about the processes they execute. One of the most challenging questions in Process Mining, and particularly in healthcare, typically is: how good are the discovered models? Previous studies have suggested approaches for comparing the (few) available discovery algorithms and measure their quality. However, a general and clear comparison framework is missing, and none of the analyzed algorithms exploits Markov Chains-based Models.

In this paper, we propose and discuss effective ways for assessing both quality and performance of discovered models. This is done by focusing on a case study, where the pMiner tool is used for generating Markov Chains-based models, on a large set of real Clinical Guidelines and workflows.

Introduction

In the past decade, Process Mining has become a significant field to help healthcare process experts understand and gain relevant insight about the processes they execute (Van Der Aalst 2016; 2011). Given the characteristics and challenges of the healthcare domain, a dedicated Process Mining for Healthcare (PM4HC) branch has recently been presented. The gap between general-purpose aims of Process Mining and the specific needs of Healthcare was early identified (see e.g., (Kaymak et al. 2012)), and the creation of the PM4HC field allowed to clarify aspects and characteristics of the Healthcare domain, for example in terms of usability and interaction with the domain experts (Martin et al. 2020) or a possible different role of Conformance Checking (Gatta et al. 2020). For an overview of the field, the interested reader is referred to the PM4HC manifesto (Munoz-Gama et al. 2022).

Being process mining an emerging field entering the healthcare domain, there are several existing challenges to address (Gatta, Orini, and Vallati 2022), such as data quality, how to build event logs, discover the best process model, what tools should be applied to obtain specific results, and how to include expert knowledge into the process (Rojas et al. 2016). Beside aforementioned challenges, one of the most pressing question arising when a process model is discovered, is: how good is it? Previous studies have analyzed the (few) available discovery algorithms to measure the quality of the models they are able to generate, but none of the included and analyzed algorithms are Markov Chains-based Models (Buijs, van Dongen, and Van Der Aalst 2012; Janssenswillen et al. 2016; Rozinat et al. 2007), and a clear and easy-to-reproduce comparison framework is missing.

In this paper we propose and discuss effective ways for assessing both quality and performance of discovered models. This is done by focusing on a case study, where the *pMiner* tool (Gatta et al. 2017) is used for generating MM-based models, according to real Clinical Guidelines and workflows.

Background

In this section we firstly provide the necessary background on process mining and its healthcare declination, and then shortly describe the *pMineR* tool (Gatta et al. 2017) and the Markov Models.

Process Mining

Process mining is a research discipline that focuses on extracting information from data generated and stored in the databases of information systems; in this case, the Hospital Information Systems. The data are extracted to build events logs, which can be viewed as a set of traces, each containing all the activities executed for a process instance (Van Der Aalst 2016; 2011).

Process mining has been applied in the healthcare domain, providing a series of significant results and analysis to the process owners and domain experts (Rojas et al. 2016; Mans, Aalst, and Vanwersch 2015). Figure 1 shows an

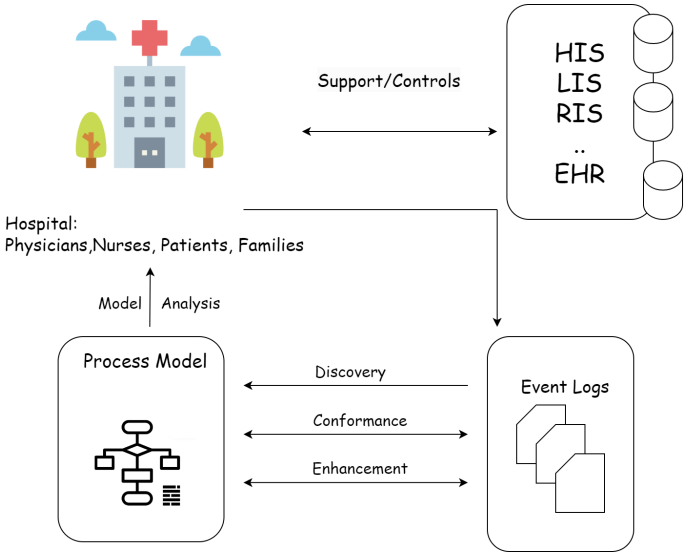


Figure 1: Process Mining in Healthcare, an overview.

overview of how Process mining can be used in the Healthcare domain. PM4HC has helped in identifying the processes executed for a specific procedure like a surgery (Neumuth et al. 2011), to verify the process regarding its compliance to specific guidelines (Grando, Schonenberg, and Van Der Aalst 2011), and also to identify collaboration patterns (Van Der Aalst et al. 2003).

There are three main areas subsumed by process mining: process discovery, conformance checking, and enhancement. Buijs et al. (Buijs, van Dongen, and Van Der Aalst 2012) explain how automatic process discovery allows process models to be extracted from an event log; how conformance checking allows monitoring deviations by comparing a given model with the event log; and how enhancement allows extending or improving an existing process model using information about the actual process recorded in the event log.

The process discovery uses a series of algorithms to represent the discovered models in different types of languages, including Markov chains (Kemeny, Snell, and others 1960), Petri networks (Murata 1989), and BPMN (White 2004). These languages include specifications to model the processes and are included in small amount of specific tools that apply the process mining techniques, such as PROM (Van Dongen et al. 2005), DISCO (Günther and Rozinat 2012), and PALIA (Fernández-Llatas et al. 2013).

In order to measure the quality of a discovered model from any process mining Discovery algorithm in the past, 4 quality metrics have been defined: fitness, simplicity, generalization and precision (Van Der Aalst 2016). Fitness corresponds to how the discovered model can replay an event log, this can be computed by replaying all the traces in the event log in the model (Van Der Aalst 2016). Simplicity corresponds to how simple a model can be while being able to represent the greatest amount of cases, without being too complex to be understood. Generalization can be defined as the capacity of a model not to restrict its behavior only to the cases included in the event, but also can include additional ones. And finally, precision, which is a measure of how a model does not allow for too many additional cases.

The pMineR Tool

pMineR has been designed and developed as an R package. R is one of the most widely used software environment for data analysis.

The modular architecture exploits *collections*, specialized sets of classes built using common methods and exchanging data structures. This is done in order to maximize the reusability of the code and the extensibility of pMineR. A macro-overview of the modular architecture of pMineR, including the interaction between objects, is shown in Figure 2.

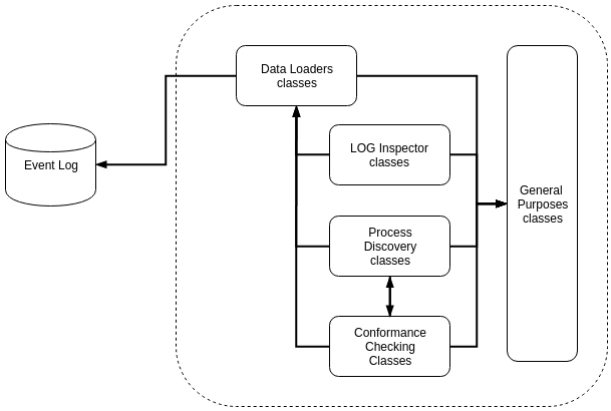
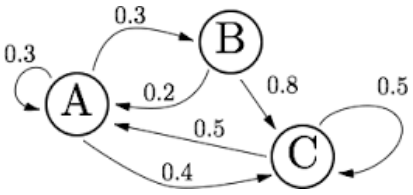


Figure 2: A general overview of pMineR structure, in terms of *collections* and their interaction.

The main *collections* of pMineR are described in the following.

- **Data Loader**: this *collection* includes classes handling the loading of event logs, and data pre-processing.
- **LOG Inspection**: is a set of classes aiming to provide some descriptive statistics on event log data, such as events and processes.
- **Process Discovery**: classes in this *collection* implement one or more Process Discovery algorithms, given a *Data Loader* object. In the current version of pMineR there are two classes implementing, respectively, first and second order Markov Models-based algorithms. Those algorithms are used in this paper to compare the resulting models.
- **Conformance Checking**: is a set of classes specialized in Conformance Checking.
- **General Purposes**: this *collection* include classes addressing a wide range of common issues that are faced during programming, such as exception handling, and strings manipulation.

We can now turn our attention to the Markov Models, that are exploited by the Process Discovery collection of pMineR.



	A	B	C
A	0.3	0.3	0.4
B	0.2	0.0	0.8
C	0.5	0.0	0.5

Figure 3: A graphical (top) and transition matrix (bottom) representation of a three state Markov chain.

A Markov model (or a Markov chain) is a mathematical model for stochastic systems whose states are governed by a transition probability. The current state in a Markov chain is a function of its previous states, e.g. for a first order Markov chain the following condition holds:

$$P(x_t|x_{t-1}, ..., x_0) = P(x_t|x_{t-1}) \quad (1)$$

where $X = \{x_0, ..., x_t\}$ is the set of the states of the system at each step in time.

In a first order Markov model each state depends only on the immediately preceding one. In second (or higher) order Markov models the next state depends on the two (or more) preceding ones. Thus, for a second order Markov model, condition (1) becomes:

$$P(x_t|x_{t-1}, ..., x_0) = P(x_t|x_{t-1}, x_{t-2}) \quad (2)$$

Given a list of the possible states of a system, the possible transition paths between those states, and the rate parameters of those transitions (stored in the transition matrix), one can represent a first order Markov model graphically, with each state usually depicted as a "bubble", arrow arcs denoting the transition paths between states, the annotated numbers on the arcs being the transition probabilities. Figure 3 provides an example of a graphical representation and its corresponding transition matrix.

Methodology

This section is devoted to describe the models, metrics, and data sets we considered in our analysis, and the experimental settings.

Models

We considered First Order Markov Models (FOMM) and Second Order Markov Models (SOMM), with different probability thresholds to accept arcs among nodes: a FOMM with threshold 0.2 is a FOMM where, in the transition matrix, probabilities below 0.2 are set to zero, and the remaining probabilities are re-normalized accordingly, in order to have sum 1 on each row of the transition matrix. In terms of notation, we will write $FOMM(i)$ and $SOMM(i)$ to indicate, respectively, a FOMM and a SOMM built with a threshold i.

Metrics

In order to catch the idea of fitness, generalization and precision, we tested how Markov models are able to detect good and bad processes, and how they –which are limited in expressiveness due to increasing values of thresholds– are able to produce good process models. As measures we considered accuracy, sensitivity, specificity, and F1, which are usual metrics for assessing the capabilities of predictive models . Accuracy measures the number of correct predictions over the total predictions made; sensitivity (specificity) provide the true positive (negative) rate; finally, F1 is a metric that allows to gain an overview of both precision and recall abilities of the model.

To estimate the simplicity of a model, we considered the total number of arcs, which is a reasonable measure of complexity from a human point of view. We consider this as a sort of "heuristic" that a human user will exploit in order to quantitatively assess the complexity of a model.

Dataset

It is well-known that real-world data –particularly in medicine– can be extremely noisy and, in many cases, they have been generated from unknown workflows (usually significantly different from the expected ones). For these reasons, real-world data can not be reliably used for effectively testing the ability of a model generated via Process Mining, in representing medical guidelines. Therefore, instead of real-world data, in this analysis we consider synthetic data generated according to real-world workflows. This allow us to generate noise-free data, or data with a specified element of disturbance, and to have a ground truth for comparison (Jouck et al. 2018). As it is pivotal to generate data as close as possible to real-world data, we only adopted real clinical workflows or real clinical guidelines.

To represent CG and workflows, we used the Pseudo Workflow Language, which is the internal language of *pMineR* specifically designed to build and represent Workflows, for implementing four different sets:

- **Test1**. A workflow representing the different steps of patient care in a Radiotherapy Department.
- **Test2**. An internal protocol, we adopted to treat patients affected by rectal cancers.
- **Test3**. A Clinical Guideline about the surveillance procedure for patients treated with I131 for thyroid cancer (Pacini et al. 2006).
- **Test4**. A Clinical Guideline about the treatment of the lung cancer, from the Italian Association of Medical Oncology (aio 2016).

Data Generation

We generated synthetic data according to the following approach:

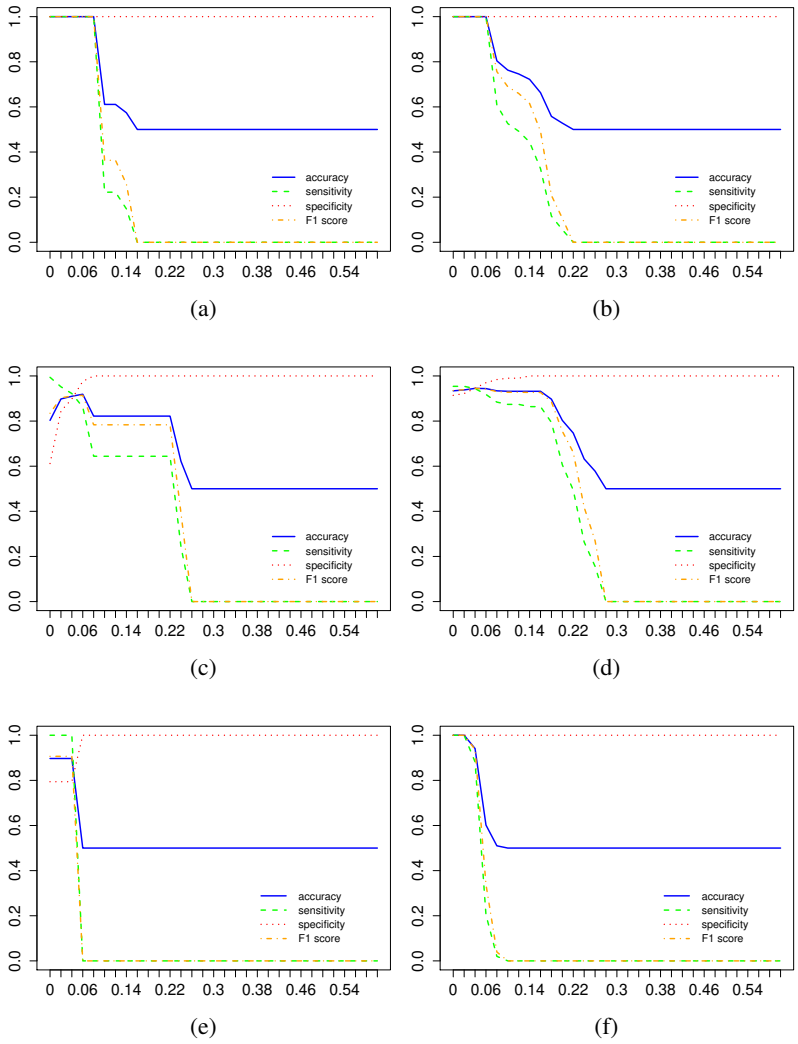


Figure 4: FOMM (on the left) and SOMM (on the right) performance on Test1 (a, b), Test2 (c, d), and Test4 (e, f) in terms of accuracy, sensitivity, specificity, and F1 score. X-axis indicates the different threshold levels exploited, while Y-axis indicates the performance of the considered metrics. Test3 results are omitted for the sake of conciseness, as both approaches show very similar figures.

1. we implemented the workflow or guideline and we generated a set of 1,000 valid processes (training set); we also generated a set (testing set) of 500 valid processes and 500 invalid processes. The invalid processes were generated by corrupting valid processes in a subtle way, by adding one non-legal consequent event to the process;

2. given the built processes, we trained 30 SOMMs and 30 FOMMs with threshold values ranging from 0 to 0.6, with a step of 0.2.

We then used pMiner as a case study for testing the effectiveness of FOMMs and SOMMS in recognizing correct and incorrect sequences of events, and assessing the complexity of generated models.

Experimental Results

The experimental analysis has been designed for investigating three main aspects: (i) the ability of models in discriminating between valid and invalid processes; (ii) the capability of catching heterogeneous valid processes; and (iii) the complexity, from a human perspective, of the models.

Recognition of Valid and Invalid Processes

In this section we assess the performance of generated FOMM and SOMM models in recognizing valid and invalid processes. Results are presented in Figure 4 in terms of accuracy, sensitivity, specificity, and F1-score. In the figure we omit the results of Test3, as the compared approaches performed in a very similar way.

Overall, in terms of accuracy, FOMM and SOMM mod-

els tend to perform similarly, but SOMMs usually allow to achieve better performance when low threshold values are exploited. Similarly, SOMMs tend to provide better sensitivity and F1 score when very low threshold levels are used. According to the presented results, SOMMs usually exhibit better performance on the considered data sets, but differences tend to become less pronounced when threshold levels increase.

Ability of Generating Heterogeneous Valid Processes

Here we focus on the ability of FOMMs and SOMMs to generate models that can recognize heterogeneous valid processes. Remarkably, increasing the threshold in a FOMM/SOMM model can reduce the set of valid processes, due to transition with a corresponding possibility of zero. In a nutshell, if P is the set of possible valid processes generated by a PWF and $F(t_k)$ and $F(t_j)$ are the sets of the possible processes generated by a $FOMM(k)$ and $FOMM(j)$, trained by the same subset of P:

$$t_k < t_j => F(t_j) \subseteq F(t_k) \quad (3)$$

To test how thresholds can reduce the set of possible valid processes we generated 100 runs on FOMM and SOMM models, with different thresholds, and we checked how many different valid processes SOMM and FOMM were able to create. The number of different valid processes gives and empirical statistical idea of the dimension of the sets $F(t_i)$. We did this test in an empiric way because $F(t_i)$ can have an infinite number of elements (due to the possible presence of auto-loops in the transition matrices). The result of the test is a heterogeneity index (hereinafter *et_index*) which represents the percentage of different valid processes with regards to the the total amount of valid processes generated. Since repeated transitions on the same state (loops) are normally meaningless in our domain, we considered sequences such as: AAABCD,ABCD, ABBBCD, ABCDDD as belonging in the same equivalence class. In fact, all of them become the same sequence: ABCD after the suppression of the auto-loops.

Figure 5 shows the et_index of SOMMs and FOMMs generated from the considered data sets. The exploitation of SOMM allows to consistently achieve a better level of heterogeneity (i.e., the et_index level is higher). On the one hand, this result is not surprising as SOMM can leverage on a greater degree of expressivity. On the other hand, it is interesting to check, in the next section, if this comes at the cost of significantly increased complexity of the models.

Complexity of Generated Models

In order to assess the complexity of a generated model, following the metrics introduced in previous sections, we counted the arcs included in the model. However, it should be noted that SOMMs can not be represented graphically, because of their higher level of dependency between states.

Figure 5 shows the complexity index of FOMMs and SOMMs generated from the considered data sets. For providing a more meaningful indication, complexity has been

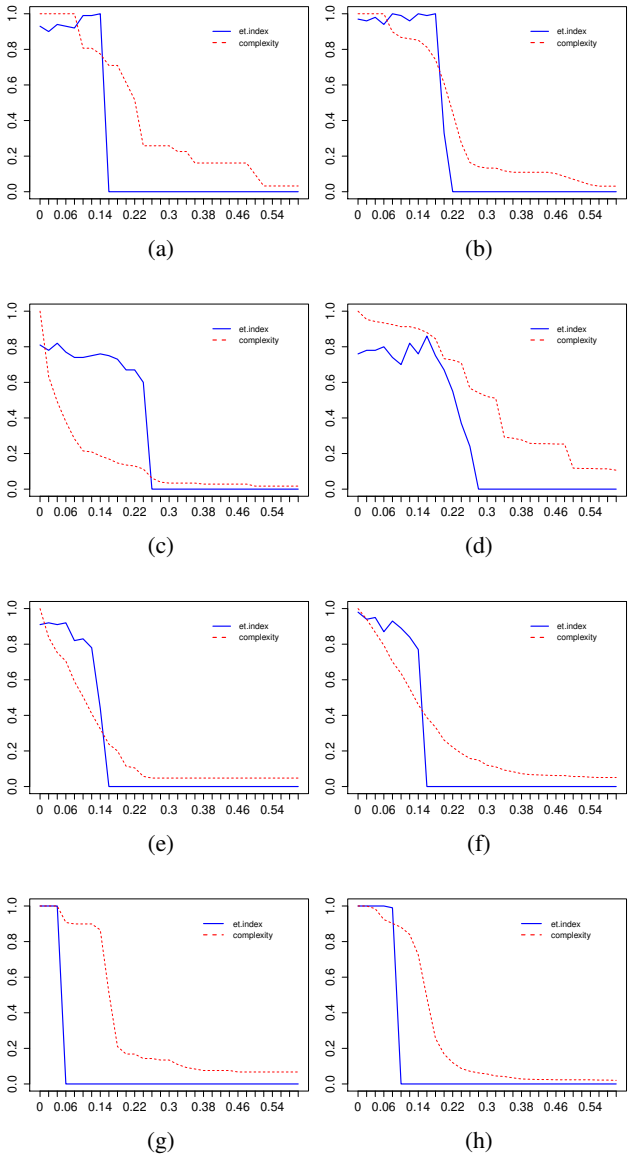


Figure 5: et_index (blue) and complexity (red) of the FOMM (on the left) and SOMM (on the right) models generated on Test1 (a, b), Test2 (c, d), Test3 (e, f), and Test4 (g, h). X-axis indicates the different threshold levels exploited, while Y-axis indicates the performance of the considered metrics.

normalized to the highest value of the model of the same class. Unsurprisingly, FOMMs have a significantly lower level of complexity: this is also confirmed by the fact that the highest number of arcs of models of the same class is lower in FOMM than in SOMM.

The complexity analysis confirms to a good extent the fact that FOMM are generally less performant when it comes to recognize heterogeneous valid processes, but at the same time they can provide an easy to represent and to interpret model, that could be beneficial for physicians and practition-

ers in the healthcare field. SOMMs can be more accurate, but their complexity and the inability to effectively represent them can hinder their usability in practice.

Discussion

Process Mining is an emerging discipline, and subsumes two important areas: Process Discovery and Conformance Checking. Process Discovery, in particular, is the step where an algorithm tries to build a model able to fit given real world data in the form of a process. In this step, two elements are of pivotal importance: the language used to represent the model of the process, and the algorithm exploited to build such model. Given the fact that Process Mining is a relatively recent field; few tools dealing with such task are currently available. Moreover, there is a lack of frameworks and methods for comparing and evaluating different approaches.

In this paper we proposed an experimental Case Report, applying FOMM and SOMM to data generated by considerign four real-world workflows, for mining the underlying processes, in order to understand their behavior according to different probability threshold. Threshold, in Markov-based models is of primary importance; good values allow to reduce the noise of raw data and simplify the generated models. We assessed the performance of the two approaches by considering different perspectives and proposing several metrics.

Our analysis highlighted that: (i) SOMMs have generally better performances than FOMMs, but FOMMs are simpler and easier to represent. It is therefore important to select the best approach according to the expected use; (ii) high threshold values have a detrimental impact on overall performance. We empirically observed that "soft" threshold values (i.e., around 0.02) usually achieve better performance, and (iii) soft threshold values are also useful for generating heterogeneous models, that are therefore more robust. Interestingly, the exploitation of 0-valued threshold does not lead to the best heterogeneity results. This is due to the presence of a large number of low-probability auto-loops, which are ignored by the final models generated for the the medical domain.

Future work includes the evaluation of different approaches for generating Process Mining models, and the formalization of an extensive and robust framework for comparing tools. Specifically, we are interested in investigating metrics that allows to compare all the steps of the Process Mining task.

Acknowledgments

Mauro Vallati was supported by a UKRI Future Leaders Fellowship [grant number MR/T041196/1]

References

2016. Aiom, italian association of medical oncology; 2016 lung cancer treatment guidelines.

Buijs, J. C. A. M.; van Dongen, B. F.; and Van Der Aalst, W. M. P. 2012. On the role of fitness, precision, generalization and simplicity in process discovery. In *Proc. of the Conference On the Move to Meaningful Internet Systems: OTM*, 305–322.

Fernández-Llatas, C.; Benedi, J.-M.; García-Gómez, J. M.; and Traver, V. 2013. Process mining for individualized behavior modeling using wireless tracking in nursing homes. *Sensors* 13(11):15434–15451.

Gatta, R.; Lenkowicz, J.; Vallati, M.; Rojas, E.; Damiani, A.; Sacchi, L.; De Bari, B.; Dagliati, A.; Fernandez-Llatas, C.; Montesi, M.; Marchetti, A.; Castellano, M.; and Valentini, V. 2017. pminer: An innovative r library for performing process mining in medicine. In ten Teije, A.; Popow, C.; Holmes, J. H.; and Sacchi, L., eds., *Artificial Intelligence in Medicine*, 351–355.

Gatta, R.; Vallati, M.; Fernandez-Llatas, C.; Martinez-Millana, A.; Orini, S.; Sacchi, L.; Lenkowicz, J.; Marcos, M.; Munoz-Gama, J.; Cuendet, M. A.; de Bari, B.; Marco-Ruiz, L.; Stefanini, A.; Valero-Ramon, Z.; Michielin, O.; Lapinskas, T.; Montvila, A.; Martin, N.; Tavazzi, E.; and Castellano, M. 2020. What role can process mining play in recurrent clinical guidelines issues? a position paper. *International Journal of Environmental Research and Public Health* 17(18).

Gatta, R.; Orini, S.; and Vallati, M. 2022. Process mining in healthcare: Challenges and promising directions. In *Artificial Intelligence in Healthcare: Recent Applications and Developments*. Springer. 47–61.

Grando, M.; Schonenberg, M.; and Van Der Aalst, W. 2011. Semantic-based conformance checking of computer interpretable medical guidelines. In *International Joint Conference on Biomedical Engineering Systems and Technologies*, 285–300.

Günther, C. W., and Rozinat, A. 2012. Disco: Discover your processes. *BPM (Demos)* 940:40–44.

Janssenswillen, G.; Jouck, T.; Creemers, M.; and Depaire, B. 2016. Measuring the quality of models with respect to the underlying system: An empirical study. In *Proceedings of BPM*, 73–89.

Jouck, T.; Bolt, A.; Depaire, B.; de Leoni, M.; and van der Aalst, W. M. 2018. An integrated framework for process discovery algorithm evaluation. *ArXiv* abs/1806.07222.

Kaymak, U.; Mans, R. S.; van de Steeg, T.; and Dierks, M. M. 2012. On process mining in health care. *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* 1859–1864.

Kemeny, J. G.; Snell, J. L.; et al. 1960. *Finite markov chains*, volume 356. van Nostrand Princeton, NJ.

Mans, R.; Aalst, W. M. P. V. D.; and Vanwersch, R. J. B. 2015. *Process Mining in Healthcare - Evaluating and Exploiting Operational Healthcare Processes*. Springer Briefs in Business Process Management.

Martin, N.; De Weerdt, J.; Fernández-Llatas, C.; Gal, A.; Gatta, R.; Ibáñez, G.; Johnson, O.; Mannhardt, F.; Marco-Ruiz, L.; Mertens, S.; Munoz-Gama, J.; Seoane, F.; Vanthienen, J.; Wynn, M. T.; Boilève, D. B.; Bergs, J.; Joosten-Melis, M.; Schretlen, S.; and Van Acker, B. 2020. Recom-

mendations for enhancing the usability and understandability of process mining in healthcare. *Artificial Intelligence in Medicine* 109:101962.

Munoz-Gama, J.; Martin, N.; Fernandez-Llatas, C.; Johnson, O. A.; Sepúlveda, M.; Helm, E.; Galvez-Yanjari, V.; Rojas, E.; Martinez-Millana, A.; Aloini, D.; Amantea, I. A.; Andrews, R.; Arias, M.; Beerepoot, I.; Benevento, E.; Burattin, A.; Capurro, D.; Carmona, J.; Comuzzi, M.; Dalmas, B.; de la Fuente, R.; Di Francescomarino, C.; Di Ciccio, C.; Gatta, R.; Ghidini, C.; Gonzalez-Lopez, F.; Ibanez-Sanchez, G.; Klasky, H. B.; Prima Kurniati, A.; Lu, X.; Mannhardt, F.; Mans, R.; Marcos, M.; Medeiros de Carvalho, R.; Pegoraro, M.; Poon, S. K.; Pufahl, L.; Reijers, H. A.; Remy, S.; Rinderle-Ma, S.; Sacchi, L.; Seoane, F.; Song, M.; Stefanini, A.; Sulis, E.; ter Hofstede, A. H.; Toussaint, P. J.; Traver, V.; Valero-Ramon, Z.; van de Weerd, I.; van der Aalst, W. M.; Vanwersch, R.; Weske, M.; Wynn, M. T.; and Zerbato, F. 2022. Process mining for healthcare: Characteristics and challenges. *Journal of Biomedical Informatics* 127:103994.

Murata, T. 1989. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE* 77(4):541–580.

Neumuth, T.; Jannin, P.; Schlomberg, J.; Meixensberger, J.; Wiedemann, P.; and Burgert, O. 2011. Analysis of surgical intervention populations using generic surgical process models. *International Journal of Computer Assisted Radiology and Surgery* 6(1):59–71.

Pacini, F.; Schlumberger, M.; Dralle, H.; Elisei, R.; Smit, J. W.; and Wiersinga, W. 2006. European consensus for the management of patients with differentiated thyroid carcinoma of the follicular epithelium. european thyroid cancer taskforce. *Eur J Endocrinol* 154(6):787–803.

Rojas, E.; Munoz-Gama, J.; Sepúlveda, M.; and Capurro, D. 2016. Process mining in healthcare: A literature review. *Journal of Biomedical Informatics* 61:224 – 236.

Rozinat, A.; De Medeiros, A. A.; Günther, C. W.; Weijters, A.; and Van Der Aalst, W. M. 2007. Towards an evaluation framework for process mining algorithms. *BPM Center Report BPM-07-06, BPMcenter. org* 10.

Van Der Aalst, W. M.; van Dongen, B. F.; Herbst, J.; Maruster, L.; Schimm, G.; and Weijters, A. J. 2003. Workflow mining: a survey of issues and approaches. *Data & knowledge engineering* 47(2):237–267.

Van Der Aalst, W. 2011. *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media.

Van Der Aalst, W. 2016. *Process Mining: Data Science in Action*. Springer.

Van Dongen, B. F.; de Medeiros, A. K. A.; Verbeek, H.; Weijters, A.; and Van Der Aalst, W. M. 2005. The prom framework: A new era in process mining tool support. In *International Conference on Application and Theory of Petri Nets*, 444–454.

White, S. A. 2004. Introduction to bpmn. *IBM Cooperation* 2(0):0.