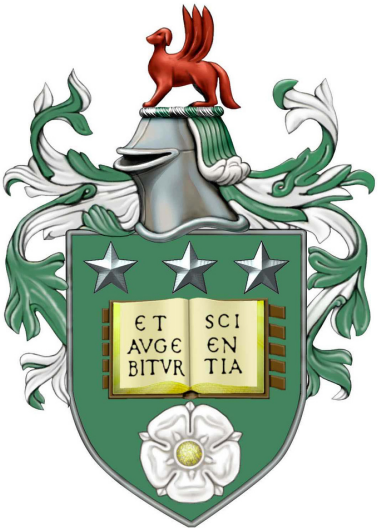
Unsupervised Abstraction for Reducing the Complexity of Healthcare Process Models

Amirah Mohammed Alharbi

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy



The University of Leeds
School of Computing
July 2019

The candidate confirms that the work submitted is her own, except where work which has formed part of a jointly authored publication has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Some parts of the work presented in Chapters 3, 4, 5 and 6 have been published in the following articles:

Publication 1: Amirah Alharbi, Andy Bulpitt and Owen Johnson, 2017, September. Improving Pattern Detection in Healthcare Process Mining Using an Interval-Based Event Selection Method. In International Conference on Business Process Management (pp. 88-105). Springer, Cham.

Publication 2: Amirah Alharbi, Andy Bulpitt and Owen Johnson, 2018, January. Towards Unsupervised Detection of Process Models in Healthcare. In Studies in Health Technology and Informatics (Vol. 247, pp. 381-385). IOS Press.

Publication 3: Amirah Alharbi, Andy Bulpitt, Owen Johnson and Eric Rojas, 2019, July. Multi-objective Optimisation Method for Abstracting Complex Processes. IEEE Journal of Biomedical and Health Informatics, Manuscript in preparation.

The above publications are primarily the work of the candidate.

"We raise in degrees whom We will, but over every possessor of knowledge is one [more] knowing." (The Quran [12:76])

Acknowledgements

First and foremost, I praise my God, Allah, who supported me through every single moment of my life and during my PhD journey. I would like to express my heartfelt gratitude and deepest thanks to my PhD supervisors Dr. Andy Bulpitt and Dr. Owen Johnson for their continuous encouragement, assistance and valuable advice throughout this work, where without them it would not exist.

I would never have the chance to study in the UK without the financial aid of my sponsor, Umm-Alqura University, and the government of the Kingdom of Saudi Arabia. I will do my best to transfer the knowledge and skills that I gained to boost the scientific research in my country and in particular to improve the research of the healthcare sector .

My sincere gratitude goes to my beloved husband Mazen who has been with me through the ups and downs and for his patience, advice and unlimited support. My beautiful children Leen and Qusai who have missed me during my long days and late nights, I promise to make it up to you. Your smiles when I come home cured my stress and kept me motivated.

I have no words to describe my gratitude for the support I received from my marvellous mother. Her prayers and endless care made me who I am today. I faithfully would like to thank every person in my lovely family; my father, sisters and brothers and their children.

Many thanks to our fabulous research team members for sharing their ideas and our stimulating discussions that have enriched my knowledge and experience. My deep thanks also go to Dr. Thamer Ba Dhafari, Dr. Eric Rojas and Dr. Ciarán McInerney for their insightful comments, feedback and encouragement.

I am also thankful to my friends in the school of computing Ahlam, Entisar, Lulu and my best friend Ebtisam for her optimism that she passes on to me every time I have phoned her.
Last but not the least, my other PhD researchers friends, thanks for all the fun we have had in the last four years, my life in Leeds would not have been the same without them.

Abstract

Healthcare processes are complex and may vary considerably among the same cohort of patients. Process mining techniques play a significant role in automating the construction of healthcare models using a system's event log. An event log is a data type that records any event that occurs within the process. It is a basic element of any information system and has three main components: process instance id, event and time when an event has occurred. Using ordinary techniques of process mining in healthcare produces 'spaghetti-like' models which are difficult to understand and thus have little value. Previously published studies have highlighted the importance of event abstraction which is considered as a central tool for reducing complexity and improving efficiency. Although studies have successfully improved the understandability of process models, they have generally relied on involvement from a domain expert. Untangling these 'spaghetti-like' models with the help of domain experts can be expensive and time-consuming. Machine learning techniques such as Hidden Markov Model (HMM) has been used for modelling sequential data for a long time. State transition modelling has also been explored by process mining research and is advocated for sequence clustering purposes where a model is trained over a group of sequences and then used to evaluate if a process instance is more likely to be generated from this model or not. However, state transition models can also be utilised for detecting hidden processes which can be used subsequently for process abstraction. In this thesis, we aim to address healthcare process complexity using unsupervised abstraction. We adopt an unsupervised method for detecting hidden processes using HMM and the Viterbi algorithm. The method in this research includes eight stages; event logs extraction, preprocessing, learning, decoding, optimisation, selection, visualization and lastly model evaluation. One of the main contributions of this research is the design of two different types of process model optimisation which are *strict* and *soft* optimisations. Models that are selected by the proposed optimisation address the limitations of other standard metrics that can be used for model selection in HMM such as Bayesian Information criteria (BIC). Two different real healthcare data sources are used in this research namely the Medical Information Mart for Intensive Care (MIMIC-III) from Boston, USA and the Patients Pathway Manager (PPM) from Leeds, UK. Models are trained using the MIMIC-III medical event log and then tested using the PPM dataset to be evaluated later by a domain expert. Three breast cancer case studies that range in complexity are extracted. The results of our method have significantly improved model complexity and provided a conceptually valid abstraction for several care patterns. Promising results are demonstrated in the improvement of the precision and fitness of the abstracted models. The abstracted models can then be used as a middle step for bringing structure to unstructured processes which helps in finding cohorts of patients based on similar healthcare processes. The healthcare processes of a cohort of patients can then be modelled using any process mining tool where their process similarity could not be captured in the complex models.

Terms and Definition

Term	Definition
Event log, log	a data type that records any event occurs in information systems. It consists of three main components: case id, events and time.
Case	a single process instance consists of different care events and belong to a single patient.
Event type	correspond to the name of distinct event.
Event	an instance of event type that has an associated timestamp and may have other attributes.
Activity	a group of events that aim to achieve a defined task. It has a duration property.
Control-flow	a general term used for describing the flow of business processes inside an organisation.
Pathway, care-flow, clinical pathway	a route that is comprised of a number of serial events which start when a patient needs healthcare and end by the completing of the patient's treatment.
Variant	a pathway that is a different from the most followed pathway.
Sequence, pattern	an ordered set of events that represents how a process is performed.
Production state	a state generates one event type.
Simple state	a state has several event types but 80% of the state is occupied by maximum 2 event types.
Composite state	a state has several event types and 80% of the state is occupied by more than 2 event types.
Complex state	a composite state and contains highly variable processes.

List of Abbreviation

Abbreviation	Term
HIS	Healthcare Information System
PAIS	Process-Aware Information System
GP	General Practice
HMM	Hidden Markov Model
HHMM	Hierarchical Hidden Markov model
MIE	Medical Informatics Europe Conferences
BPM	Business Process Management Conferences
JBHI	Journal of Biomedical and Health Informatics
EM	Expectation-Maximisation
BIC	Bayesian Information Criteria
AIC	Akaike Information Criterion
ICL	Integrated Completed Likelihood
EBIC	Extended BIC
EHR	Electronic Health Record
LTHT	Leeds Teaching Hospitals Trust
MIMIC-III	Medical Information Mart for Intensive Care
PPM	Patient Pathway Manager
CFC	Control-Flow Complexity
BPMN	Business Process Modelling Notation
ED	Emergency Department
ICUs	Intensive Care Units
CPT	Current Procedural Terminology
XES	eXtensible Event Stream
ICD	International Classification of Diseases
CCU	Coronary Care Unit
CSRU	Cardiac Surgery Recovery Unit
MICU	Medical Intensive Care Unit
SICU	Surgical Intensive Care Unit
TSICU	Trauma Surgical Intensive Care Unit
LM	Local Models
ERD	Entity-Relationship Diagram
OSA	Optimal String Alignment
L	Linearity
SC	State Compactness
CSS	Cross State Similarity
SI	State Importance
WHO	World Health Organisation
PAS	Patients Administration System
SOP	Standard Operating Procedure
EC90	Epirubicin and Cyclophosphamide
struct	Structuredness
IM	Inductive Miner
SM	Split Miner

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Leveraging healthcare data

Healthcare Information Systems (HIS) have rich data related to patients' health conditions and delivered healthcare services. Different kinds of data can be found in healthcare information systems such as; text, image and events. There is a body of literature and applications have investigated medical data in different forms, for instance, designing clinical support systems, diseases diagnosis or medical image analysis. However, care events have received relatively limited attention compared with other medical data types that have been explored over the last decades. A care pathway "*describes the sequence of care that is recommended for patients with similar conditions requiring similar treatment*" [4, pg.1]. A care event represents any event that is performed on a patient while he/she requires treatment and is associated with timing information. Modern information systems store events automatically in a particular component of the system called 'event logs' and these systems are known as Process-Aware Information System (PAIS). Most HIS do not record the care events automatically in one single components, however, event data is scattered throughout the system where every department in a healthcare organisation can record its relevant care events [5]. Extracting details of these events can help to construct the care process that a patient has experienced.

Nowadays, healthcare can be provided at different levels relating to the complexity of the care that is required. These are: primary care, such as the General Practice (GP), secondary care, such as hospitals, and tertiary care, represented in highly specialised care mostly for inpatients [5]. Healthcare organisations strive to provide effective services and support best practice while at the same time reducing their costs [6] in order to meet the high demand of providing better healthcare quality. One popular resource for ensuring best practice in the UK is NICE, The National Institute for Health and Care Excellence [7]. It aims to help healthcare providers and commissioners explore standard patient pathways and to enable them to adopt these standards to match their resources. Healthcare practitioners are supposed to follow these guidelines which

are designed mainly to improve standardisation and reduce unnecessary care variations. A clear question raised here is to ask if these guidelines are followed in reality or not.
Process mining can play a significant role in answering this kind of question. It aims to exploit events logs that are recorded by the system to construct a care process model and visualise it. Recently, there has been growing attention on applying process mining in healthcare due to the promising results that have been proven in other various domains such as industry and business. There is great potential for applying process mining in the healthcare domain as discussed in Mans et al.[8]. The main important advantage of this is in providing evidence-based models that are generated from reality which bring deeper insight into patients pathways. In addition process mining supports the exploration, management and improvement of the quality and outcome of the healthcare processes. This includes measuring the compliance with guidelines and discovering the mainstream pattern of care and other deviations. Further, process mining allows stakeholders to be up to date on what is going on in the organisations which in turn ease the adoption of necessary changes. Besides that, process mining supports exploring healthcare from different perspectives for instance, the organisational and performance perspectives. The organisational perspective on process mining can include analysing the relationship between people who interact with the system. It can answer question such as 'How can hospital administration reschedule staff timetables to reduce cost or increase number of patients treated?' whereas the performance perspective focuses on measuring the throughput of healthcare processes and finding the bottleneck in the system by answering questions such as 'when patients have waited and for how long?'.
Modelling the care processes within healthcare organisation is a challenging task due to the inherent complexity of patient care. Processes may vary considerably among the same cohort of patients as organizations and clinicians vary in their response to each individual patients different physiological, psychological and social needs. Unfortunately, the majority of existing process mining techniques are designed to discover models of structured processes unlike in healthcare where processes are complex by nature. Therefore, the process mining benefits mentioned above cannot be achieved over complex models which are described by [9] as spaghetti-like models referring to the tangle and connectivity between its lines. There are different reasons contributing to this complexity and these will be discussed in Chapter 2. Hence, the need to discover a useful model that can represent reality in a simple and understandable way is the first step in order to apply healthcare process mining. This thesis attempts to address that challenge.

1.2 Process mining

According to Aldin and Cesare [10], the origins of process mining go back to the research thesis of Cook [11] where the aim was to discover the process of software development which the author named '*process discovery*'. After that process discovery was applied in Agrawal et al.

[12] on the business domain and was called *'workflow mining'*. The term '*process mining*' was coined later in 2001 when Wil van der Aalst and Weijters proposed a technique to improve workflow management systems [13]. Designing a workflow model is a challenging task because it requires comprehensive specifications and understanding of the processes within an organisation. Besides that, traditional paper-based workflow models do not illustrate the actual flow of processes since they are built based on business rules which may not be followed in reality. Process mining aims to discover information from event logs, which are available in information systems, to monitor the flow of the process and ultimately improve or extend the process model [14].

Process mining has three main goals that generate a different outcome: discovery, conformance checking and enhancement [9]. Discovery techniques use event logs to generate a real process model without the need for any description of business rules. This technique is the most common process mining technique used in organisations. The second goal of process mining is using conformance checking technique, which aims to measure the compatibility between an event log and a standard process model that is already described and known by the business manager and should be followed. Conformance checking takes both the event log and standard model to produce compatibility information which shows if the standard model is actually followed or if there is any deviation. Enhancement is the third goal of process mining and is used to improve the current model by detecting the limitations or bottlenecks of the existing model or by checking the outcomes of variant process flows.

The merit of process mining is not solely related to discovering the control-flow perspective. There are various other perspectives that can be analysed, such as organisational, data and performance perspectives. The organisational perspective aims to display the relations between resources, humans or devices, to gain deep insight into their roles and the interaction flow between them. The data perspective focuses on other information related to a particular case/instance flow. The performance perspective reflects process efficiency and elapsed time for each process; this is important in order to recognise the potential bottlenecks within a system [14].

Healthcare information systems are valuable resources of clinical processes which show the real flow of healthcare procedures. Using process mining in the healthcare domain can improve healthcare standards and outcomes. More precisely, hospitals are very flexible and divers environments in terms of patients care-flows. Therefore, process mining may enable healthcare administration teams to gain deeper insight into daily clinical processes by knowing the mainstream, exceptional pattern and extreme anomalies. Furthermore, the healthcare domain is a competitive sector; thus, process mining is a key solution to help healthcare organisations improve.

Increasing interest in process mining has resulted in the establishment of different process mining tools. The most prominent tool for researchers is ProM [1] which was developed by

[1] http://www.promtools.org/doku.php

the Process Mining Group in Eindhoven Technical University in 2010. It is an open source extensible platform and the current version, ProM 6.8, has more than 120 packages. Other frameworks available for commercial use that may allow a limited research license for example, Disco[2], Celonis [3] and others. Also, there are a number of process mining separated packages that are implemented in the R framework such as; BupaR[4] and pMineR[5], the latter is designed for healthcare process particularly. It is worth noting that, most techniques within the above tools generate non-understandable spaghetti-like models when applied to healthcare processes, with the exception of a few techniques that attempted to abstract care events. These techniques are discussed in Chapter 2.

1.3 Problem statement

The complexity of healthcare processes is the first obstacle that hinders the application of process mining within healthcare (see the example illustrated in Figure1.1). On the one hand, a general approach is suggested by Wil van der Aalst [9] which aims to split the entire event log into smaller homogeneous sub-logs. This approach can help in analysing complex models, however, the general process model for the entire event log cannot be discovered. The hypothesis of this method is that fewer models are complex, but more models are simpler. Research that adopted this approach mostly depended on case clustering and this will be discussed in Chapter 2.

On the other hand, most process mining methods that tried to address complex model issues rely on the concept of events abstraction. The aim of this abstraction is to group fine grain events into high level main events. Therefore, within the process mining literature, abstraction is considered as a central tool to reduce complexity and improve efficiency [15]. Methods that have used event abstraction will be discussed in detail in Chapter 2. Unfortunately, those methods have abstracted events in a supervised way which requires the involvement of domain experts in the stage of events abstraction and throughout process mining steps for validation and evaluation.

Although studies have successfully improved the understandability of process models, they have been overwhelmed by the consequences. Untangling spaghetti-like models with the help of domain experts is expensive and time consuming for many reasons. First, there is a need for finding appropriate domain experts who can understand the process or have worked closely with the intended system. Second, these domain experts mostly need to be paid. Third, it is tedious to arrange the times for long discussions that suit all parties for example, process miner, domain experts and any other possible party. We have been advised of a previous

[2]https://fluxicon.com/disco/
[3]https://www.celonis.com/
[4]https://www.bupar.net/
[5]http://www.pminer.info/progetti/website/main.php

work [16] that developing a model for chemotherapy care using a clinical reference group of domain experts required eight iterations lasting over nearly a year. Clearly, there is a need for developing a better more efficient method for discovering the general process model by applying events abstraction without the need for domain experts or at least to mitigate their involvement. Machine learning techniques can be a key solution for applying unsupervised events abstraction and reducing the need for such domain experts.

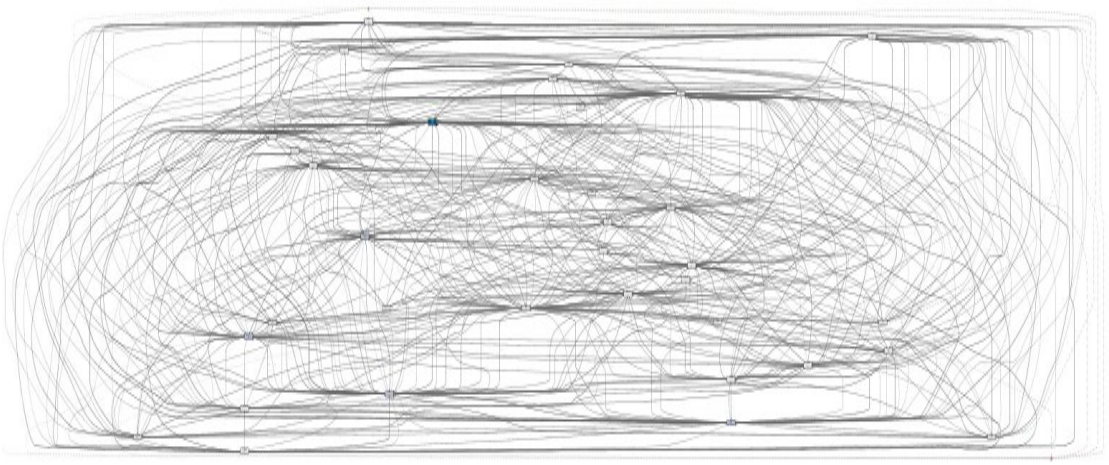


Figure 1.1: A complex healthcare model illustrating spaghetti (provided by Fuzzy miner)

1.4 Process mining challenges in healthcare

Some previous literature [17], [8] [18] and [19] has discussed several challenges of applying process mining on healthcare data however, this research focuses on the following challenges.

1.4.1 Challenge 1: Healthcare model complexity

Healthcare models are complex and this complexity is the expected result of several choices that have to be taken to meet individual patients' needs. The simplest question that should be answered by any process mining technique is 'What is the main pathway that is followed by any cohort of patients?' This cannot be answered using current process mining tools due the highly complex nature of any healthcare process which produced a complex unstructured model. Hence, there is a need to develop a method to cope with this complexity and generate a comprehensible model.

1.4.2 Challenge 2: The need for domain experts in process mining

Although some techniques that have previously been proposed to try to address model complexity, all of these rely on the involvement of domain experts for event abstraction throughout all process mining steps starting with abstraction and modelling and ending with validation and evaluation. Untangling complex models with the help of domain experts can be expensive and time consuming. Hence, the question of 'Can we simplify healthcare process model without the need to involve domain experts in the abstraction phase?' is still open.

1.4.3 Challenge 3: Extracting care events from Electronic Health Records (EHR)

Event log extraction and preprocessing are critical steps for process mining research and this is recognised in the 2011 Process Mining Manifesto [14] as the first challenge in process mining. Some EHRs are non-process aware systems which means, event logs do not already exist in the system and are not available to use but, rather, events are distributed over different system components, for instance different database tables within the EHR. Therefore, extracting care events from EHR is challenging and requires more effort and stages.

1.4.4 Challenge 4: Knowing the right model for unstructured process

There is no clear answer for the 'what is the right model?' question, however, there are some process model quality metrics that have been discussed in literature and will be explored in Chapter 2 of this thesis. It should be noted that, these metrics were invented to assess the quality of business process models in general where these models are basically discovered from structured process. However, no attention has been given to assessing process model quality for complex and unstructured processes. Therefore, using the same metrics of quantifying the quality of a structured process model over an unstructured process model is questionable and requires further investigation. A suggestion would be to consider other alternatives and to think about the question: 'What are the characteristics of best process abstracted model?' since abstraction is a significant principle in resolving the complexity issue.

1.4.5 Challenge 5: Visualising the model

Considering the challenge of abstraction for healthcare model complexity leads to a further challenge in terms of visualising the abstracted model. Process mining tools usually visualise process models by creating a node for every event type and using event name for node labelling. However, applying the same principle for visualising abstracted models will require labelling for model nodes, where the node in the abstracted model is a block containing more than one event type. Labelling the abstracted model will lead us back to the second challenge of the inevitable reliance on domain experts. Therefore, providing a transparent representation (events can be

seen through the pie charts states) for abstracted nodes might improve model understandability while minimizing the cost of involving domain expert in model discovery. Hence, this challenge suggests rethinking the applicability of using the standards process discovery tool for process model visualisation in order to visualise the abstracted model.

1.5 Research overview

This section provides an overview of the research in terms of the research aims, a brief description of the research method and a list of the contributions to current knowledge.

1.5.1 Research aims

The research aims are as follows:

1. To produce a new tool that can be used to reduce the complexity of healthcare models in order to improve process understandability and discover the mainstream pattern of care from apparently unstructured processes. The tool will support an unsupervised event abstraction method which does not require the involvement of domain experts during the abstraction stage.

2. To evaluate the applicability of using the Hidden Markov Model (HMM) for abstracting care events into states. This requires conducting different experiments of learning HMMs with various event logs. In addition, we shall investigate the characteristics of the selected HMMs in order to identify the criteria of the preferable and undesirable abstract models.

3. To test the proposed unsupervised abstraction method and tool on the real-world data of healthcare processes for breast cancer patients in the Leeds Cancer Centre [6].

In order to achieve these research aims, several stages require to be completed. These are extracting care events from EHRs and providing clean event logs that can be used for process mining, identifying the limitations of current process mining algorithms that targeted complexity, selecting the best abstract model based on novel optimisation and providing better visualization for the abstract healthcare model of different case studies. Table 1.1 shows a summary of the research objectives and each related chapter, contribution and publications to date.

[6] https://www.leedsth.nhs.uk/a-z-of-services/leeds-cancer-centre/

Table 1.1: Summary of research objectives and related chapter, contribution and publications

Research objectives	Sub-activity	Chapter	Contribution	Publication
Finding and extracting healthcare events		Ch.3	Con.1 and 2	BPM [20]
Identifying the limitations of current techniques in process mining that target complexity		Ch.2 , Ch.3		
Investigating the potential of using machine learning (HMMs) for abstraction		Ch.4	Con.3	MIE [3]
Developing an improved method of state abstraction	Improving model selection	Ch.5	Con.(3.a), Con.(3.b)	JBHI [21]
	Identifying new types of hidden states	Ch.5	Con.6	
	Improving model visualization	Ch.6	Con.5	
Testing the proposed method on real healthcare processes	Extracting three case studies	Ch.6, Ch.7		
Investigating the ability of the abstract model to find similar cohort of patients		Ch.7	Con.4	
Implementing a tool for unsupervised event abstraction		Appendix D	Con.7	R package documentation

1.5.2 Research method and tools

Research method

Our hypothesis is that the aims of this research can be achieved by adopting machine learning techniques based on Hidden Markov Model (HMM) with its relevant algorithms such as the Expectation-Maximisation (EM) and Viterbi algorithms.
First, in this research we have adopted the general method of using the HMM which consists of three main steps: learning, selecting and decoding. Learning is done by estimating the parameters of the HMM using the EM algorithm and then selecting the best model using information criteria model selection metrics that are well-known in literature such as Bayesian Information Criteria (BIC). The sequences of the care provided are decoded using the Viterbi algorithm in order to ascertain which event belongs to which state.
Second, we have improved this method by developing a novel optimisation function for selecting the best model which would replace the traditional selection metrics and cope with the shortcomings of the traditional metrics.
The improved method for healthcare process abstraction developed during this thesis resulted in an eight stage method that can be summarised as:

1. **Extraction:** The first step is event log extraction from an EHR. Extraction is guided by

inclusion and exclusion criteria for several case studies.

2. **Preprocessing:** After extraction, event logs need to be preprocessed. This step includes converting an event log into the form of horizontal sequences where care events that belong to a single case should be in one row, which is the appropriate data format in order to train the HMM.

3. **Learning:** The abstraction stage starts here where the algorithm that is used for learning is the EM, and will be discussed in Chapter 4.

4. **Decoding:** Decoding is conducted by running the Viterbi algorithm, which is the most used decoder with HMM, over all sequences of care events.

5. **Optimisation:** This step is for optimising the models space using our novel multi-objective function which includes both soft and strict optimisation.

6. **Selection:** Models are selected based on the maximum score from the optimisation results and at this stage, the abstraction process ends.

7. **Model visualization:** Process models are visualised using our new visualisation tool that is more appropriate for modelling abstracted processes.

8. **Evaluation:** Models are evaluated using three different aspects for evaluating the selected process models. This step also includes models selection validation using some metrics.

These steps can be repeated if the best selected model still includes a complex state. Events related to the complex state can be isolated and extracted for a repeated analysis that will, in turn, explain and simplify this complex state. This step is required in order to provide a better abstract process model through hierarchical modelling of the complex state. All the steps in our approach are illustrated in Figure 1.2 and will be discussed in detail in Chapter 5.

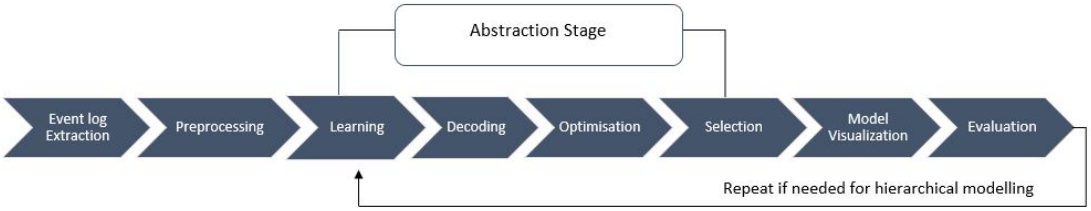


Figure 1.2: Method overview

Research tools

This research is conducted using a mixture of tools. The Postgres-SQL (9.5) platform was used for setting a local version of the EHR for data sources, the R programming language (3.5) was used for training HMMs, process mining tools such as ProM (6.8) was used for exploring

healthcare process and evaluate our method with state-of-art approach and lastly, Disco (2.2) was used for process performance analysis when required.

1.5.3 Healthcare data sources in this research

In this research we used two different sources for healthcare data, namely the Medical Information Mart for Intensive Care (MIMIC-III) [22][23] and Patients Pathway Manager (PPM) [24][16]. According to Pollard and Johnson [23], the MIMIC dataset has been used in 134 publications mostly describing data mining and machine learning approaches. This dataset is provided by the Beth Israel Deaconess Medical Centre in Boston, USA and contains data of 46,520 patients. The PPM, on the other hand, is a mature EHR and holds the clinical and coded data of all patients who have cancer at the Leeds Cancer Centre, which is one of the largest cancer centres in the UK. This healthcare system was developed by Leeds Teaching Hospitals Trust (LTHT) in 2003 and includes the data of 2.39 millions patients. Table 1.2 gives a brief comparison between MIMIC-III and PPM. Further elaboration on these two data sources is presented in their related chapters.

Both sources have a considerable number of recorded care events that can be used for mining patients pathways. The reason for having two different data sources is that, the MIMIC-III dataset is available online and has rich temporal event care that helps us in exploring process models and learning and exploring HMMs, however, we cannot evaluate our method using MIMIC-III since there is no access to domain experts. Therefore, the PPM data is used to test our method and evaluate the results with domain experts.

Table 1.2: Healthcare data sources that are used in this research

Data sources	**MIMIC-III**	**PPM**
Abbreviation for	Medical Information Mart for Intensive Care	Patients Pathway Manager
Origin	Beth Israel Deaconess Medical Centre in Boston, USA	Leeds Cancer Centre in the UK which is a part of Leeds Teaching Hospitals Trust (LTHT)
Span	2001 - 2012	2003 - 2015
Number of patients	46,520	2.39 millions
Diseases	different diseases	cancer diseases
Care level	Intensive care unit in tertiary care	Primary care and secondary care
Patient privacy	anonymised	anonymised
Process tracing	Patient ID , admission ID	Patient ID
Event temporal resolution	Date + Time	Date
ICD code	ICD-9	ICD-10
Accessibility	online available after the accomplishment of an online ethics course	require ethical approval form and access authentication
How it is used in the designed research method	Exploring	Testing
Related chapters	3,4 and 5	6 and 7

It should be noted that, there is another online healthcare event log that is supported by the Business Process Management (BPM) challenge[7], which is recorded in a Dutch Academic Hospital. However, we found some difficulties for using it because it was not in English language.

1.5.4 Research contributions to current knowledge

The contributions of this thesis can be summarised as follows:

1. Providing healthcare event logs in English that is extracted from MIMIC-III database. To the best of our knowledge, the MIMIC-III medical databased in this research has been used for the first time for mining healthcare processes. In order to facilitate the use of MIMIC-III in process mining for future researchers, the steps of creating local database of MIMIC-III and examples of event log extraction are provided in our published paper [20].

2. Developing a novel pre-processing approach to improve pattern detection of periodically occurred events within healthcare processes such as charting events inside intensive care units. This method has successfully reduced the variations of the repeated events and

[7]https://www.win.tue.nl/bpi/doku.php?id=2011:challenge

improved the visualisation of their patterns. The impact of this contribution is published in our paper [20].

3. Developing an improved unsupervised abstraction method that boosts the understandability of complex healthcare process models and reduces the complexity. This method and initial results are published in our paper [3]. In order to develop this method, a number of further contributions are invented as well:
   (a) A novel model selection criteria which includes linearity, state compactness, cross state similarity and state importance and this includes the criteria calculations.
   (b) A novel multi-objective optimisation function that is designed for selecting the best abstracted model. In order to provide more flexibility in the model selection, two types of optimisation are proposed which are *strict* optimisation and *soft* optimisation.
4. Describing a new strategy for selecting similar patients and cohort analysis based on a state abstraction model. This is a significant contribution where the ultimate goal of healthcare process mining research is to find similarity between the care processes which cannot be tackled in complex models. This strategy highlights a further functionality of the state-based model and forms the link between the HMM's abstract model and the process mining framework. The abstract model helps in selecting a similar cohort of patients and then the healthcare processes of this cohort can be modelled in a process mining framework, such as ProM or Disco.
5. Other contributions are considered in the enhancement of the hidden Markov process visualisation in order to provide better modelling for the abstracted process. This new transparent visualisation is inspired from the "*Seqhmm*" hidden Markov model package in R [25]. The improvements include, demonstrating clear start and end nodes to improve model understandability, using frequency instead of probability on edges and supporting better layout of states that helps trace the flow of the process.
6. Providing a new classification of hidden states that is helpful in model description and identifying the necessity for hierarchical modelling of a state. Three types of states are identified which are simple, composite and complex. The definitions of these states are discussed later in Chapter 4.
7. Open-source implementation of the proposed method of multi-objective optimisation and other functions. This package is written in the R environment, see Appendix D.

1.6 Thesis structure

The structure of this thesis is organised as follows:
Chapter 2: Background and Literature Review

This chapter explains the nature of healthcare processes and provides background of process mining research in healthcare with emphasis on healthcare complexity and variations analysis. Also, this chapter discusses general process mining techniques and techniques that are advocated for discovering unstructured process. We conclude this chapter by explaining the theoretical basics of Hidden Markov model (HMM) and how HMM is used in process mining research.

Chapter 3: Event Log Extraction and Pre-processing

This chapter focuses on the early step of our method, which is event log extraction and pre-processing. MIMIC-III is the centric data source of this chapter. Detailed steps of acquiring and preparing event log are presented since this research is the first work that uses MIMIC-III for process mining purposes. This chapter also presents two experiments that are conducted to demonstrate the limitations of the current process mining techniques that can be used for complex process modelling.

Chapter 4: Machine Learning Approach for Healthcare Process Abstraction

This chapter explores the potential of using HMMs to afford an unsupervised abstract model for complex processes. The method of using HMM for process abstraction is explained in three main steps; learning, decoding and model selection. Some well-known information criteria metrics for model selection are described. Then different empirical results are discussed which in turn provide evidence of practical issues which can be found in HMM models that are selected as best models.

Chapter 5: Multi-objective Function for Process Abstraction

In this chapter we propose four criteria that may help in selecting the most desirable abstracted process model. The rationale behind these criteria and their calculation is presented here. In addition to demonstrating the criteria properties and steps for designing a multi-objective function. This chapter focuses on the optimisation role of the designed multi-objective function, which is an important step in our methodology. We conclude this chapter by adopting and improving the method of modelling complex processes using state abstraction that was developed in Chapter 4.

Chapter 6: Case Study 1: Chemotherapy cycles of breast cancer patients

The aim of this chapter is to demonstrate our proposed method for discovering the general pathway of complex healthcare processes. In this chapter we provide an application of our proposed optimisation method using real world event logs from Leeds Teaching Hospitals Trust (the PPM system), which allows us to evaluate the results with domain experts at the last stage of our method. The case study of chemotherapy cycles of breast cancer patients is extracted from PPM and evaluated. Process models for this case study are visualised using the new visualization method.

Chapter 7: Further Experiments: Case study 2 and Case study 3

This chapter focuses on addressing further complexity beyond case study 1. Two case studies are extracted from the PPM. The proposed method is also applied here in order to test the

capability of discovering the mainstream care patterns for more complex processes. Selecting and analysing cohorts of patients is presented in this chapter to demonstrate the applicability of the proposed strategy for selecting patients using our state based abstraction model. The models are validated and evaluated using two aspects, conventional metrics of process mining evaluation and lastly evaluated by the domain expert.

Chapter 8: Conclusion

We conclude the thesis in this chapter by outlining the main contributions of this work and how they have been achieved. A discussion of possible improvements and future work is presented as well.

Chapter 2

Background and Literature Review

2.1 Overview

This chapter explores the nature of healthcare processes and provides background of process mining research in healthcare with emphasis on healthcare complexity and variations analysis. Also, this chapter discusses general process mining techniques and techniques that are advocated for discovering unstructured process. We conclude this chapter by explaining the theoretical basics of Hidden Markov model (HMM) and how HMM is used in process mining research.

2.2 Healthcare process and its implications

Healthcare organisations are complex systems because they involve different components represented in people such as administrators, doctors, patients and nurses and different departments and clinics in the healthcare process [17]. Therefore, several challenges are posed on applying process mining in the healthcare. Analysing the challenges of process mining that are discussed in [17] can help with categorizing these challenges into two main main areas of concern: data quality and process characteristics. Data quality problems include missing data, incorrect data (such as mismatch between entries and its corresponding fields in the system) and imprecise data for instance, time data needs to be accurately logged. On the other hand, a major process characteristics issues concerns case heterogeneity. A case, which is a single care pathway, can be highly divergent because of various patients conditions, unforeseen complications requiring medical intervention, which in turn may alter the pathway, and different levels of patient adherence to treatment. This problem increases the difficulty of finding the most frequent pathway and other allowed variations. According to [26], although there are a number of techniques that tried to tackle high variations problem, more work is needed to solve this issue. Other problems

of process characteristics are handling the large number of distinct events, which may lead to the construction of overly complex process models, and how to deal with high volumes of data that require scalable and efficient process mining algorithms.

2.3 Process Mining in Healthcare

According to Wil van der Aalst [27], there are three types of process mining projects in general, which can be applied to healthcare as well. The first type is data-driven project which aims to gain insight to the process in hand and there is no a particular question intended to be answered. The second type is a question-driven project where this type has a question or list of questions that requires answers whereas the third type is a goal-driven project which is motivated to achieve a particular goal for example, reducing the throughput time of a given task. The general proposed methodology, which is known as L* life cycle, for conducting any type of process mining projects cannot be used easily in complex healthcare processes as stated by the author in [27]. This is due to the challenges that are posed in some stages of the projects such as control-flow discovery and model enhancement. An example of early work that has done by Mans et al. in [28] where they used a real case study of oncology patients in a Dutch hospital for process mining. They adopted the L* general methodology in that research. However, few techniques of process mining were explored since most of the available techniques were designed for mining structured processes and several challenges were encountered. The results were encouraging and showed the applicability of using process mining in providing new insights of the healthcare process.

There are a number of papers that provide a literature review of process mining in healthcare. Four of them have discussed a general review of healthcare process mining. The centric of these papers are to explore different aspects of techniques, process based questions, case studies that are applied on healthcare process mining [18] [19] [29] [30]. On the other hand, another four review papers are focused on a particular domain that has used process mining for healthcare such as oncology [31], cardiology [32], elderly care [33] and primary care [34].

In this section we are interested in highlighting the findings of the general literature reviews since they provide bigger picture of healthcare process mining techniques and future directions. In [18], 37 process mining papers that are published between 2004 and 2013 were examined. The results showed that three major areas have been covered which are process discovery, process variations analysis and process model improvement and evaluation.

For process discovery, different techniques are used such as Heuristic miner, fuzzy miner with clustering and association rules mining. Although these techniques generated understandable process models, they failed to produce an accurate or generalised model that reflects the reality of healthcare processes.

The area of analysing process variations was examined through conformance checks under a standard model. Heuristic miner was used but was unable to detect variations because of the

complex and detailed model. While the association rule miner successfully detected outliers, the generated model suffered from confusion around the AND/XOR joint, as well as split or missing events. As a result, the model lacked simplicity.
The third and least explored area of discussion from existing literature concerns the application of process mining and how process models can be improved in the context of healthcare. The idea behind process model enhancement involves adding new event log data to improve the model or keep it updated.
The findings of [18] have broadly shown the drawbacks and limitations of current process mining techniques in clinical pathway analysis. This is because of the unsuitability with highly dynamic, unstructured nature of the health care environment and the lack of comprehensive clinical process mining framework that is able to combine more than one process perspective analysis. The authors of [18] have asserted the need for further improvements on clinical process mining techniques, especially for challenging aspects of healthcare domain such as variations analysis and their influences, patients identification, unceasing model adaptation and active clinical process model recommender.

Another review paper [19], has a wider scope of healthcare process mining studies where 74 papers were investigated. For each study different aspects have been explored included the kinds of data used in process mining, key questions that should be answered, process mining tool that is used, the perspective of the analysis , medical case study and process mining algorithms that are applied. The findings for each aspect are summarized as follows. Two main types of data were explored which are medical treatment process that includes vital signs and other medication events or organisational process that includes administrative events. There were also two kinds of questions asked: the first about what happens generally in healthcare organisations and the second about guideline compliance among processes. Most process tools that were used are ProM and Disco. The review paper found that all process perspectives, which are mentioned early in the introduction, were explored in healthcare process mining research. The top medical case studies that adopted and utilised process mining were oncology and surgery. Lastly, the most frequent process mining algorithms that were applied were fuzzy miner, Heuristic miner and sequence clustering which is sometimes known as trace clustering approach, these algorithms will be discussed later in this chapter.
In addition, the author of [19] suggested possible future directions for process mining in healthcare such as the design of a portable clinical process mining framework easily integrated within any hospital information system and the shift to process-aware information systems able to automatically support clinical pathway decisions. More efforts should be advocated to improve clinical pathways visualisation and to create a simple and uncomplicated process model to eliminate the necessity of experts.
The two other literature review papers [29] [30] have emphasized on the same challenges and trends however, [30] is focused mainly on the importance and challenges of conformance check-

ing in healthcare. The aim of this paper was to investigate what kind of features can help in building conformance analyser tool for healthcare process. Interestingly, the paper has found that only few studies (3 out of 11 examined papers) have proposed process enhancement using some insights that were suggested by conformance checking results. The author of [30] asserted the need for building more appropriate conformance tool for measuring the compliance of complex healthcare process.

Variations analysis is mentioned often as a significant companied issue of complexity in all literature review papers. It is an essential goal in healthcare process mining because of highly divergent patients pathways. Variations analysis comes at the following step after finding the mainstream process care in a complex environment. Recently, variations analysis has received considerable attention by healthcare process miners and some studies have showed promising results. The detection and analysis of pathway deviations can be addressed using different methods.

One of the methods, which was suggested by Rebuge and Ferreira in [35], was to recognise mainstream and deviation pathways in an emergency radiology process using Markov sequence clustering as a preliminary step before using ProM tool for model discovery. Sequence clustering method in this paper is an extended work of [36] and it aimed to split an event log into smaller groups and build a process model related for each event log subset. It was designed to group sequences into only three clusters, set by threshold. One cluster is for the most frequent sequence and its similar sequences, the second cluster is for variant sequences and the third cluster is for infrequent sequences. However, this method tested on one single department where processes are presumably not very complex, also this method lacked a clearly defined evaluation metric, which is a common issue in clustering evaluation, and depended on a user defined threshold for setting the number of clusters.

Leemans et al. [37] have proposed a new process miner algorithm called the inductive miner, which was designed to improve variations extracting and visualisation. The inductive miner was able to explore the type of deviation whether it is resulted from adding new event that is absent in event log or some events that are not permitted by the model. Another approach for process anomaly detection was reported by Bouarfa & Dankelman [38]. They used multiple sequence alignment algorithm to extract outliers and find consensus sequence from surgery event logs.

Hompes et al. [39] have discussed variations analysis from different perspective. They have adopted Markov cluster algorithm (MCL), which is a graph clustering algorithm, in order to cluster sequences based on different attributes for example diagnosis-based clustering or age-based clustering. This technique has focused on detecting variations within a synthetic patient log according to data perspective and not a control flow perspective. Using MCL allowed clustering the sequences into unpredefined number of clusters with different sizes. The idea behind the MCL relied on using a transition matrix probability that represents the likelihood of transitions between events in all sequences.

Depaire et al. [40] also proposed a framework for process variation analysis. This framework aimed to answer administrative questions regarding swapped activities, repeated subsequence and deviations places. This paper highlighted different categories of deviations and distinguished between exception - a positive deviation based on deviation outcome- and anomaly- a negative deviation refers to human error or fraud.
It should be noted that, all these studies have tackled variations detection without further analysis of other critical questions, such as the possible reasons for pathway deviations or how deviation correlates to positive/negative outcomes.
Very few papers that have attempted to find the correlation between pathway deviations and outcomes. An example of this is the work of Li et al. [41] where they transformed clinical behaviours into first-order logic sequences and used a particular metric which helps in pattern recognition. This method returned promising results regarding the correlation between deviations and outcomes. For instance, they found a positive correlation between different congestive heart failure care-flows and the frequency of patient readmission.

2.4 Complexity in healthcare processes

Modelling the healthcare processes is a challenging task due to the inherent complexity of patient care. Processes may vary considerably within the same cohort of patients as organizations and clinicians vary in response to each individual patients different physiological, psychological or social needs. Process mining techniques can play a significant role in understanding these real patterns of care through the application of machine learning algorithms to the event logs extracted from Electronic Health Record (EHR) systems [19].
EHRs log numerous events during a patients visit to a hospital including medical, administrative, laboratory, intensive care and billing events. An event log records each event as a tuple with identifiable attributes including event name, event time and patient ID. Many healthcare events overlap or occur in conjunction with other events which aptly reflecting the “interrelatedness” of healthcare processes [1].

2.4.1 Complexity Definition

From healthcare point of view, the term complexity has been defined in [1] based on the interaction between systems components which include both people or department. These interactions are refereed to as “interrelatedness”. This definition is commonly agreed on and adopted from non-healthcare fields as [42] [43]. The complexity increases by increasing the number of system components, the interaction between them and the uniqueness relations of interaction, how often an interaction happens once or repeatedly is a key consideration in determining a processs complexity.
We believe that the definition of healthcare complexity should not be defined as per these three

factors alone as there might be hidden contributing factors, such as emergent events or the day when the healthcare service was needed. Yet we can think of those factors as implications or fingerprints of complexity, which then allows us to measure how complex an interaction really is.

From process mining point of view, however, complexity is defined as confounding factor that can prevent generating useful models [44]. In this thesis, we think the first definition, healthcare definition for complexity, is more meaningful and should be embraced to where healthcare process model complexity is defined by how a component corresponds to an event; how an interaction represents a link or edge between events; and how unique an interaction or relationship is based on the variation of sequence of events. Furthermore, the type of interrelatedness is significant in process modelling, and we consider it the fourth factor that increases complexity. Examples of different types of interaction in process modelling known as process structures such as; sequence, parallel and choice. These fundamental process structures are explained later in Table 2.1 in this chapter.

Therefore, we suggest to modify the range of complexity that is mentioned in [1] to include the type of interrelatedness. The complexity increases as the interaction may represent several types as illustrated in Figure 2.1. This adapted figure outlines different levels of complexity. Increasing the number of components and interaction, where this interaction represents different types of interrelatedness between components, resulting in a more complex healthcare process. Unlike a simple healthcare process, with only a small number of components and near homogeneous interactions between components.

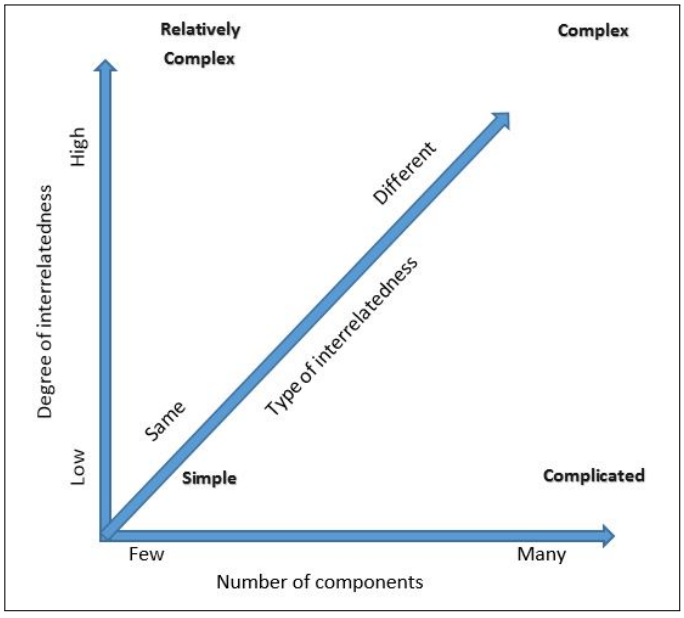


Figure 2.1: Range of complexity in healthcare processes, a modified Figure from [1].

From a theoretical point of view, when the term complexity is mentioned, we should consider Occam Razor's and its principle of favouring simple solutions over complex ones.

2.4.2 Causes of Complexity

Complexity in healthcare is a hot topic and has been discussed widely in healthcare papers. It is an expected result of several reasons that have to be taken to meet individual patients needs as discussed in [45].
A primary reason for complexity is the variation within care processes. This variation is inevitable in the healthcare domain due to a range of causes. The causes can be categorised into; medical causes, organisational causes and implications of both medical and organisational causes.
• Medical causes include such variables as patient condition and treatment required. We as process miners have no control over these causes.
• Organisational causes involve multiple care dimensions, such as healthcare system providers, who might follow rules that are different than other systems; doctors, who diagnose patients and decide on the best treatment or intervention; and nurses, who help patients throughout their process of care, record patient notes and all three of these dimensions interact with the system.
• Implications of both medical and organisational causes that left fingerprints on event logs it could be improved using process mining techniques such as repeated events, care events that are recorded with different levels of granularity, looping over number of events. The resulting complexity from all these factors combined will dramatically affect on the understanding, description, prediction and management of healthcare processes.

2.4.3 Complexity Measurement

Based on the complexity measurements that are discussed in [46], [47] and [48] we can classify these measurements into two types which are; measures that are generated from event log and measures that are generated from process model. It should be noted that, the measurements attained through process models are subject to the algorithm type that is used for discovering that model. Event log measurements are based on the number of cases and the number of events while process model measurements require more complex formulas for generation, including:
• **Size**: it measures how big a model is which is sometimes depending on the number of nodes in a model or the number of nodes in addition to the control flow elements [49].
• **Control-Flow Complexity (CFC)**: which is the number of branching edges from all nodes.
• **Structuredness**: this metric is concerned with the structure of the control flow where every split node must have a corresponding joint node. For instance, a node with outgoing edges should have a node of incoming edges to ensure a (single-entry single-exit) block structure between the outgoing node and the incoming one. It can be calculated as one minus the

number of nodes inside a structured block divided by the number of total nodes in a model.
• **Understandability**: an objective metric for which there is no accurate measure. This metric is, however, affected by previous complexity measurements.
According to [50], structuredness does not necessarily improve model comprehensibility; two models may have the same structuredness score, but a model of bigger size will result in low understandability, just as the more edges or branches a model has, the less understandable it will be. Therefore, the challenges of designing a process model in a structured and meaningful way are difficult to quantify.

2.5 General process mining techniques

There are several techniques and approaches that have been implemented to do process mining such as Alpha miner, Heuristic miner and Inductive miner. According to [9], the difference between process mining techniques mainly relies on the adopted process discovery method and how this method addresses two major aspects; representational bias and noise and incompleteness.
The first aspect is representational bias, which refers to the capability of a modelling representation language to represent various process structures. Basically , there are a number of modelling languages that can express and model processes such as Petri net, which is the most prominent one in ProM tool, BPMN (Business Process Modelling Notation), heuristic net that is known as causal nets or (C-nets) and process tree. For more detail about these languages see [51]. Every one of these languages has its own strengths and limitations however they all try to demonstrate high expressiveness of various process structures. Process structures, sometimes called constructs, represent the type of relation between events. A sample of fundamental process structures are outlined in Table 2.1.

Table 2.1: Examples of fundamental process structures

Process structure	**Definition**	**Synonym**
Sequence	event x is followed by event y	-
Parallel	event x is followed by some events for instance event y and event z regardless of their order	AND , fork
Choice	event x is followed by at least one or more of events; event y, event z or both y and z	OR
Exclusive Choice	event x is followed by either event y or event z	XOR
Loop	Event x is followed by event x for at least one time	iteration, cycle

It should be noted that, there is a workflow pattern initiative[1], which is supported by Eind-

[1] http://www.workflowpatterns.com/

hoven University of Technology and Queensland University of Technology, it aims to identify all possible process topologies or structures.
These topologies are captured within process mining and workflow management research and the initiative has categorised them based on their relevant mining perspective for instance, process structures that are captured when mining control-flow, data or resource perspectives. This initiative helps in understanding the representational bias of various modelling languages.

The second aspect relates to noise and assumptions of incompleteness. Methods that are designed for process discovery are presumed to have mechanism to cope with noise and incompleteness of event log. Noise in this context, as defined by Wil van der Aalst in his book [9], means infrequent process instance in event log that is very dissimilar of the mainstream process. While incompleteness is related to the ability of a process discovery method to discover a generalizable model that can reflect all process instances presented in event log and other possible processes that are very similar but might be missing in the log.

2.5.1 Alpha miner

Alpha miner [13] was the first process mining algorithm that attempted to discover process model and to bridge the knowledge gap between event log and business modelling. The main idea of Alpha miner is to scan all process instances of event log to find possible relations between events and use them to build what is called "footprint" matrix. The basic algorithm of Alpha miner was able to detect only four basics process structures which are direct follow relation (event 'a' is followed directly by event 'b') no direct follow relation (event 'a' is never followed directly by event 'b'), dependency relation which is a special type of direct follow (event 'a' is followed directly by event 'b' and event 'b' is never followed directly by event 'a') hence, there is a dependency relation between 'a' and 'b', and the last relation that is used in Alpha miner is the parallel relation that is discussed above.
Alpha miner uses Petri net modelling language that supports simple and understandable notations. Alpha miner has been improved upon several times and extended to include advanced process structures for instance, loop and XOR.
Although Alpha miner can produce a simple and understandable model in the form of Petri net, model quality was low in terms of fitness and precision metrics, which will be discussed later in this chapter. Also,the Alpha miner cannot handle infrequent process instances since all process instances are used to build the "footprint" matrix, that is used for building a process model.

2.5.2 Heuristic miner

Heuristic miner [52] was designed to deal with noise and incompleteness in event logs. It focuses on extracting the relation between events for instance, finding the dependency of two events.

Constructing a process model using a Heuristic miner can be achieved through three main steps. The first step involves extracting dependency and frequency information between events (for example find the frequency of event 'a' when it is followed by event 'b'). The second step requires the construction a graph based on dependency and frequency information. In other words, some rules are derived from the dependency and frequency information based on a predefined threshold of relations occurrence. This step shows how Heuristic miner can deal with noise, infrequent process instance, of event logs. The third step calls for the design of a process model based on the second step. Heuristic miner uses causal nets, C-nets, as a representation modelling language.
Further improvement of Heuristic miner is implemented using time perspective to construct a causal dependency matrix. For instance, some sequences of a log have event 'a' that is followed by event 'b' but in other sequence event 'b' occurred before event 'a' is finished. This means there is no causal dependency between event 'a' and 'b'. Although Heuristic miner tried to eliminate low frequent sequences, noise, it is impractical miner because it may generate unreadable model for logs with high number of events. In addition of generating unsound model, which violates model quality where the model has a fired transition that cannot reach the end of the process. The overall quality of process models generated by the Heuristic miner depends highly on the configuration of removing infrequent sequence that affect on the extracted relations of dependency [53].

2.5.3 Inductive miner

Inductive miner [54] was created to explore process through the support of different configurations of process model. It was also developed to cope with model unsoundness, which is a major limitation in Heuristic miner [55]. It applies divide and conquer technique by splitting the log into sub-log recursively. This can be done by finding a proper cut-off relations such as sequence, exclusive choice and parallel. Moreover, the Inductive miner supports a visualization of deviated sequences with its frequency besides a number of process instances filtering techniques. Models that are discovered by the Inductive miner are built in the form of a process tree, which in turn can be converted to a Petri net.
The most advantageous feature of the Inductive miner is the ability to replay all process instances which guarantees high model fitness. On the other hand, it has a limited number of cut-off relations and a problematic representation of long dependency between events. From our experience and based on [55], in the case of large event logs which might have high variable process instances, the Inductive miner may generate a useless model where imprecise cut-off points can be found. Hence, the discovered model will be in the form of 'flower model', as described by [55], where all transitions between events are allowed.

2.6 Process mining techniques targeted complexity

The process mining algorithms previously mentioned are used to discover models of structured processes, and thus they cannot handle complex processes and generate 'spaghetti-like' process models. However, there are some process mining algorithms that are designed to generate understandable models from complex environment.
A guideline of the properties that should characterise any algorithm intended to address complexity is outlined in [56]. The authors aimed to bring structure to unstructured or complex process using map metaphor. The four properties suggested by [56] are aggregation, abstraction, emphasise and customization. Aggregation is concerned with minimising the amount of information on the map, while abstraction is to do with hiding unnecessary information but with the ability to recover it at any time. Emphasis is used to highlight important marks on the map using colour or contrast, and customisation refers to how the user can customise the map based on his or her preference.
Several techniques have been proposed to address complex processes; these techniques are discussed below. The following algorithms have tried to include one or more of the above discussed properties.

2.6.1 Fuzzy miner

The main purpose of the fuzzy miner, which is discussed in [57],[58] and [59], is to simplify process model that is generated from a very flexible environment, such as hospitals.
The fuzzy miner uses a simple mechanism aims to find highly frequent events and preserves them. The less frequent but highly connected events are aggregated into clusters, while the less frequent and less connected events are removed. The word fuzzy in its name refers to the degree of abstraction and vagueness of the output model. The resultant model is constructed using fuzzy model representation language, which is a graph that consists of nodes (events) and edges (relations). The graph model does not, however, distinguish between fundamental process structures such as choice, parallelism and others. In addition, a fuzzy model cannot be converted into a Petri net, in cases where further performance analysis is needed. It requires considerable time for selecting the best setting configuration such as setting the frequency threshold and other parameters.
Despite Fuzzy miner does not show all sequences that are presented in logs, by neglecting exceptional process instances, it is still the most used technique in process mining applications [57]. This is due to the powerful features that are provided for instance, an interactive and zoom-able model, using different colours to boost model visibility; for example, fuzzy miner uses darker colours corresponding to the high frequency events in the log, and wider lines corresponding to highly connected events. In addition to using configurable parameters to show different levels of model details.

2.6.2 Abstraction based Methods

The word abstraction in the field of Computer Science can be used in different contexts, such as object oriented programming, software engineering and data representation, the latter of which is the scope of this thesis. However, the meaning of abstraction in all contexts is similar.
There are some definitions of abstraction in literature. According to Oxford dictionary of Computer Science[2], the word abstraction is defined as *"The principle of ignoring those aspects of a subject that are not relevant to the current purpose in order to concentrate solely on those that are".*
A similar definition mentioned by Guttag in [60, pg.43] *"The essence of abstractions is preserving information that is relevant in a given context, and forgetting information that is irrelevant in that context."*
Colburn and Gary in [61, pg.174] highlighted the aim of abstraction as *"abstraction in computer science facilitates the modelling of interaction".* While Aho and Ullman in [62, pg.1] defined abstraction as a method for problem solving *"science of abstraction - creating the right model for thinking about a problem and devising the appropriate mechanizable techniques to solve it."*

Process mining methods that address complex model issues mostly rely on the concept of event abstraction. Some papers called it as a Two-step strategy because firstly, events should be abstracted then, a process model is built.
A general review of process mining literature identifies three major approaches adopted for event abstraction: the supervised-based approach; the pattern-based approach; and the local process model approach.

a) Supervised based approach

In complex organizational environments such as healthcare the details of specific events are often recorded with a different degree of granularity. Variable granularity events produce complex process models which can be reduced by mapping low level event, highly specific events, to high level main event.
This mapping is approachable through two main ways; domain expert knowledge or the availability of ground truth source.

• Using domain expert knowledge:
In [63], the authors suggested a formal method for mapping events to activities using the domain knowledge provided by stakeholders. This method has successfully captured m:n mapping relations between low level events and high level activities.
Previous work in [5] utilized a domain expert in identifying and mapping events. Although they have produced a valuable process model, this strategy was extremely time consuming and

[2] http://www.oxfordreference.com/

expensive in terms of domain expert participation. Another paper [64] proposed a supervised event mapping method by involving a domain expert to identify activities patterns.

• Mapping with ground truth data:
Ground truth data can be found in the intended system which consider as internal source or external source. An example of internal source is the data available in event log. The ground truth data method is dependent on information availability from the event logs or the system used to extract the event log. For instance, modelling the process based on organisational roles in a hospital as mentioned in [65] where model nodes represent who has performed an event, if it is performed by a nurse, physician or technician. Also, similar work is discussed in [66], where care sub processes were modelled using the name of hospital premises, where that events occurred, for tracking patients locations.
On the other hand, an example of external data source is data can be available on-line. Some simple mapping can be achieved where low level event names are grouped into categories or medical ontologies such as SNOMED-CT[3], as highlighted in [67].

b) Pattern based approach

Several events abstraction methods are suggested by [68] as preprocessing steps for mining process models. The goal is to explore various definitions of patterns and how these patterns might be linked to process constructors on the models. The focus of the patterns was on loop structure, which was treated as an array of event with different time and was considered as a conserved pattern. Techniques for extracting primitive patterns such as maximal and super-maximal repeats were used. The method was tested on real-world healthcare processes and the results showed a relative simplification of the model.
Special patterns can be extracted based on their frequency as discussed in [69] where the author aimed to find what he called an 'episode', which is an unordered subset of events that occurred frequently within process instances. An example of this approach is the episode miner in the ProM tool, which can extract events with direct follow and parallel relations only.
Such an approach does not support exclusive or and loop relations and can be extremely slow with large event logs. Also, users must select initial parameters to limit the length of the extracted episode. Then such groups of episodes can be used for abstracting events and consequently, discovering process model over abstracted episodes.

c) Local process model approach

A local process model approach aims to discover related events that are not necessarily have explicit sequential order. These events may come with different relations such as parallel, choice

[3]http://www.ihtsdo.org/snomed-ct/snomed-ct0/

and loops. The proposed algorithm works incrementally on subsets of events to build local model based on a process tree.
Niek et al. in [70] developed a method to find the best-fit local process models through the generation of multiple possible local models from a limited number of events and a recursive process tree exploration approach. An evaluation of the generated models using four different suggested quality criteria is performed. These criteria are; confidence, determinism, language and coverage where confidence is the average number of observing events in a local model, determinism concerns about the ability of local model to predict next event, language here measure the ratio of all allowed behaviour and the observed pattern and lastly coverage metric aims to calculate the ratio of total number of events in the log to the number of events included in local model. The resulted models are ranked based on weighted average score of the proposed metrics.
Although this method has successfully discovered frequent non-sequential patterns , it has two major limitations. First, this approach cannot capture start-to-end process. Basically, it is designed to discover local process model that is fit between episode miner, which has mentioned above, and process discovery miner. Second issue is that, the limited number of potential events that could be discovered which ranges between three and five.
Using local process model for events abstraction is suggested in [71]. The proposed method aimed to pick the top ranked local models for abstracting sub-logs. Although this method has effectively abstracted some sub-logs, it required a large number of parameters configurations in addition to the ambiguity of selecting multiple local models with the same score.

2.7 Machine learning techniques for unstructured process discovery

Several machine learning techniques have been used to address complexity in unstructured processes. Examples of techniques such as clustering and sequential model learning are discussed below.

2.7.1 Model-based sequence clustering

Clustering in process mining is used also to simplify complex models. Generally, model-based clustering methods depend on a metric for similarity and a clustering algorithm. As explored in [72], computing similarity between sequences is different than other data and attention should be paid for the order of events. Therefore, the authors have suggested a transformation technique to convert every sequence to a vector and call that vector sequence profile. After that, some well-known distance measures such as Euclidean and Hamming distance can be used for

computing the similarity score over processes.
Several clustering algorithms are investigated in [72] for instance, k-means, Agglomerative Hierarchical Clustering (AHC), and sequence clustering [35]and [36], the latter are mentioned earlier in variation analysis section. The results were promising and algorithms have found multiple processes clusters. The concept of sequence profile has helped in clustering similar cases for complex logs and could be extended for adding further attributes in sequence vector.
In order to develop a more robust clustering technique, the researcher in [73] has proposed a semantic clustering method. The aim was to use the metric of edit distance and improve it to be context aware metric. This method involved differentiating edit operations weighting. In other words, the weights of edit operations for example, addition, deletion or substitution will vary based on the frequency of occurrence. The proposed edit distance has outperformed other sequence clustering techniques in terms of fitness and precision score for the resulted clusters. It should be noted that, the idea of context aware edit distance is used in this research for computing state compactness, as will be discussed later in Chapter 5.

HMM Background

This section provides a general background of HMM algorithm that is used widely for modelling sequences. Some relevant algorithms such as the EM and Viterbi are discussed likewise. A discussion of the canonical problems that can be solved using Hidden Markov Model is out of the scope of this chapter for further detail the reader refer to [74].

2.7.2 Hidden Markov Model (HMM) - late 1960s

Hidden Markov model (HMM) is a probabilistic model developed for modelling sequences. It has been used widely in different fields such as language and speak recognition and bioinformatics. The origin of HMM comes from Markov chains which provide a representation of transitions between observations. The transitions between observed states, in the case of Markov chain, or hidden states, in the case of HMM, are controlled by the Markov assumption where a transition to the next state depends on the current state or in other words, the future depends on the present.
The theory of HMMs can be described mathematically by these parameters λ =(A, B, π):
• An initial state probability which describes where a system can start. It is represented by a vector π where: $\sum_{i=1}^{n} \pi_i = 1$
• A set of states $S = s_1, s_2, s_3, ..., s_n$ where n is the number of states.
• A set of transitions $A = a_{11}, a_{12}, ..., a_{nn}$ where A is the transition probability matrix.
• A set of events or (observation) symbols $E = e_1, e_2, e_3, ..., e_m$ where m is the number of event types.
• A set of observations $B = b_{e1s1}, b_{e2s1}, ..., b_{e_m s_n}$ where B is the observation probability matrix.

a) The Expectation-Maximisation (EM) algorithm

In most cases on HMM is used for unsupervised learning which means the hidden states that have generated sequences are unknown. The Baum Welch or forward-backward algorithm, which is a form of Expectation-Maximisation (EM) algorithm, is used for HMM parameter estimation. Learning the parameters from sequences includes estimating the probability of transition matrix and the probability of observation matrix given a set of observations and a number of hidden states. There are three basic questions that should be answered in order to estimate HMM parameters:

- The first question is how likely is a state to be the first state in the process?
- The second question is how likely is an event e be observed in state s?
- The third question is how likely a state s_i will transit to state s_j?

All these questions can be combined into one general question which is 'what is the probability of a specific event occurring in our data?'. The basic answer would to find the expected count of an event which is equal to the sum of the probability of that event happens in the data [75].

The EM algorithm provides an efficient technique to answer this question through two steps; Expectation (E step) and Maximisation (M step). In the E step, the aim is to count the expected probability of transitions between states and observing events in states using the initialization values of λ =(A, B, π).

The forward-backward algorithm is used here to find the expected probability by summing over all possible paths. The M step, takes the expected count generated by the E-step and considers it as a real count to produce the maximum likelihood. For instance, the maximum likelihood of a transition between state i to state j is the expected count of that transition normalised by the total number of transitions from state i as presented in the following equations where a_{ij} is used for updating the transition matrix and $b_j(e)$ is used for updating the observation matrix.

$$a_{ij} = \frac{expected\ number\ of\ transitions\ from\ state\ i\ to\ state\ j}{expected\ number\ of\ transitions\ from\ state\ i}$$

$$b_j(e) = \frac{expected\ number\ of\ times\ in\ state\ j\ and\ observing\ event\ e}{expected\ number\ of\ times\ in\ state\ j}$$

The E-step and M-step are iteratively computed until convergence.

The convergence is reached when the difference between two subsequent models is very small based on a predefined tolerance rate [25]. The EM algorithm computes the likelihood of generating sequences from estimated parameters using the Forward algorithm at each iteration and this step is known as an evaluation procedure. For more detail of the calculations in the EM algorithm refer to [74].

b) Viterbi algorithm

The Viterbi algorithm is a standard decoder used for finding the best sequence of hidden states [76]. This algorithm [77] is used after HMM model training step to understand the underlying states that most likely generate the observations. The Viterbi algorithm efficiently finds the most probable sequence of hidden states by picking the path with the maximum state probability for the entire sequence. Also, it supports an internal pointer to backtracking the states that maximize the probability. The forward-backward algorithm can also be used for decoding the sequence of underlying states. It aims to find the most likely state at each position of the sequence then assemble these states to construct the underlying path. However, this may result in finding unpermitted transition between states for instance, state at time t has zero probability to transition to state at time t+1 [78].

2.7.3 HMMs in process mining literature

Although hidden Markov models (HMMs) have been widely used in various domains such as bioinformatics, speech recognition and handwriting recognition for handling sequences, there is a limited number of research that has applied HMMs in process mining. According to Rojas et al. [19], only 4% of healthcare process mining techniques have been done by HMMs. In this section we aim to present how HMMs have been employed in process mining field.
Peters et al.[79] and Poelmans et al. [80] used a hybrid method consists of data discovery technique such as formal concept analysis (FCA) and Hidden Markov model. The aim of using FCA is to extract semantic information about different patients clusters and then feed these clusters to the available Matlab toolbox HMM model for process discovery.
This approach has achieved good results in [80] for finding exceptions care flow in each cluster and simplifying correlation analysis between length of stay in hospital and some missing treatment practise. Also, in [79] this hybrid method found some process improvements suggestions. However, the adopted method had a priori separation of process into different clusters before process discovery step and consequently, the results are highly affected by the validity of clustering method. Also, discovering the mainstream process model is not applicable using this method.

Carrera & Jung [81] have utilized a HMM to model resources workflow and improve resources allocation. The novelty of this work comes from combining organizational, control-flow and probabilistic perspectives in one process model.
HMM parameters were initialized manually not randomly by constructing footprint matrix with frequencies for observations, where observations here are the resources names, and states, which represent events. Initial transition probability was created based on a dummy start/end observation. The algorithm of Expectation-Maximisation was used to learn the hidden structure of resource workflow. Results showed that HMM can be used to model resources workflow and

consequently improve managing resources allocation and avoiding overload.

Rozinat et al. [82] have employed HMM to measure the quality of process model. This work was motivated by the need for finding new evaluation metrics for process model to measure what is beyond the ability for replying process instances such as noise resistance and incompleteness.
A Petri net model was constructed and each labelled tasks on that model is mapped to a hidden state on HMM. The experiment aimed to gradually inject noise to several event logs then standard process model metrics for instance fitness and precision are measured.
They found that HMM can provide a reliable method to evaluate model accuracy in the existence of noise besides other common metrics such as precision, fitness and simplicity, which will be discussed in the following section.

Applying HMMs for sequences clustering has been proposed by Elghazel et al. [83] and Silva [84]. In [83], they proposed a hybrid approach of graph-based clustering and HMMs. In the first step, patients pathways are clustered based on a graph clustering method suggested in [85]. The second step is learning HMM for each cluster.
Although this method could suggested a pathway for new different patient, however, this approach applied on healthcare events during hospital stay only which considers relatively short process. The scalability of this approach has not been tested on complex processes. Besides of the same shortcomings of previous methods which is inability to model the mainstream care process.
In similar work of [84], HMMs performed as a framework to do a general sequences clustering method. The clustering relies on the probability of a sequence that may generate from a constructed HMM. The probability of generating a sequence is calculated and then the sequence is added to the most similar cluster.
This paper has discussed a theoretical framework for applying HMM in process mining but no experimental results were presented.
On the other hand, Khodabandelou et al.[86] have offered a new application of using HMMs to extract intentional process model from business event log (not healthcare processes) where hidden states correspond to user behaviour. Several HMMs were trained with different number of hidden states suggested by the stakeholder and the best model was selected using a well-known metric the Bayesian information criterion (BIC) .
Moreover, an extended work of that is a comparison between supervised and unsupervised learning is experimented by Khodabandelou et al. [87] using different event logs. The aim was to test if HMM capable to drive new insights on customer strategies comparing with strategies that already known to the stakeholder. They have developed a framework called 'map miner' which uses a HMM to learn transitions between customer behaviour and events. Then the transition matrix and observation matrix are imported to a map miner algorithm in order to visualize customers behaviour.

Interestingly, HMM has revealed many strategies more than the expected ones and the results were promising and have been verified by stakeholder. It should be noted that, this method used business event logs which mostly represent structured processes. Also, selecting the best model relied on the validity of BIC metric.

For prediction purpose, Meier et al. [88] suggested a clinical decision support system using HMMs that help physician explore the best treatment flow for a specific cohort of patients and predict the current phase of oncology treatment. Their method has intended to learn two different HMMs with three and seven hidden states which was recommended by physicians with experience using that data.
Li et al.[89] has implemented HMMs for similar goal which is detecting variations in multi-stage treatment disorder. Two steps are included, first, the model identifies treatment stage based on patients data then displays number of variations of the current stage. HMM was learned with annotated processes where stage label is known. Therefore, the model represented high performance in detecting accurate treatment stage.

The following table provides a summary of how HMMs are used in process mining research that have been proposed, the case studies, whether these techniques are available and ready to use and the general approach of adopting HMM. Four out of nine of the proposed methods are applied on healthcare data. Interestingly, the majority of these papers have trained HMMs closely with domain experts or by using a priori clustering technique to divide the process before process model discovery, which consequently prevents an ideal representation of mainstream process model.

Table 2.2: HMMs approaches and ProM.

HMM in Process mining papers	**Case study**	**Available to reuse**	**Approach**
Poelmans et al. [80]	Breast cancer patients	No	multiple HMMs for each a priori set number of clusters
Peters et al.[79]	Synthetic data for business process	No	multiple HMMs for each a priori set number of clusters
Carrera & Jung [81]	Synthetic data for business process	No	single HMM with a priori set number of states
Rozinat et al. [82]	Synthetic data	Yes	mapping Petri net to HMM
Elghazel et al. [83]	Real healthcare process	No	multiple HMMs for each a priori set number of clusters
Silva[84]	Synthetic data	No	theoretical discussion of using HMM for sequence clustering
Khodabandelou et al. [87] [86]	Real Eclipse developers log	Intended to install on ProM	single HMM selected by BIC
Meier et al. [88]	oncology	No	two different HMMs where number of states are recommanded by physicians
Li et al. [89]	congestive heart failure	No	single HMM with a priori set number of states

2.8 Models quality metrics in process mining

Model evaluation is a critical issue in process modelling due to the fact that there might be more than one model that can represent the data. Also, a question of "what is the best model?" is difficult to answer and depends highly on the domain itself.
Furthermore, event logs are built from the reality thus, negative examples of the process would not be found in the log. Interestingly and according to [90], there are two different characteristics of event logs, that seem to be conflicted, which are :
Event logs are **(trustworthy)** because everything that is recorded in the log must have happened, however, not everything that might have happened is recorded in the log **(incompleteness)**. To cope with the incompleteness issue, a model should be able to represent other possible behaviours that might be allowed, but attention should be paid to prevent modelling ones that are not allowed.
In order to resolve evaluation issues, some metrics must be developed. Plenty of research such as [90] and [91] have discussed several process model evaluation metrics. Those metrics are used to assess process models from different perspectives such as fitness, precision, generalisation and simplicity.

(a) Fitness

Fitness, sometimes called as reply fitness, is the ability of a model to reflect all processes that are recorded in the log. The idea of model fitness can be derived by counting how many false negative examples that cannot be presented in a model but are exist in the log.
Fitness is calculated in our experiments based on [92], where simple alignment score costs on move on log (if event is observed on log only) and move on model (if event is observed on model only) and no cost is made for a synchronized move that is occurred between log and model. The alignment cost is calculated for each individual process instance with respect to the generated model after finding the optimal alignment. Then alignment cost is normalized by the cost of the worst scenario that may happen where no synchronisation move occurs between the log and model (the denominator). The best fitness score is 1 and the worst model fitness is 0. Fitness score is calculated for each individual trace then normalised by the number of total traces.

$$fitness = 1 - \frac{optimal\ cost(log, model)}{move(log) + move(model)} \tag{2.1}$$

For more explanation an example for fitness calculation is discussed. Suppose we have a model M that is presented in Figure 2.2(left) and trace (a,b,c,d,e,g). First we need to calculate the alignment between the trace and model M which counts the moves between model and trace. Different alignments can be found such as the alignments that are presented in (A) and (B), however, ProM tool uses an algorithm that guarantees the optimal alignments [92] such as the alignment in (B) where the cost = 2 however, the cost of alignment in (A)= 4.

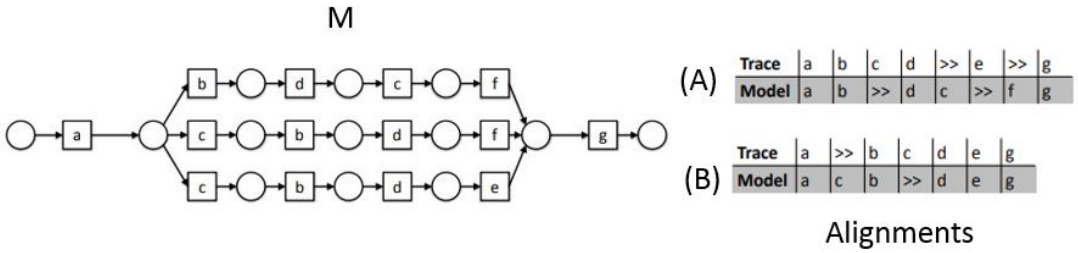


Figure 2.2: Fitness calculation example, the symbol '>>' represents no synchronisation.

Hence, using the fitness formula that is presented in (2.1), the fitness(M,trace) of alignment (A) $= 1 - \frac{4}{6+6} = 0.67$ and the fitness(M,trace) of alignment (B) $= 1 - \frac{2}{6+6} = 0.83$ which is the used fitness in ProM tool. It should be noted that, evaluating the process model by measuring its ability for reflecting reality (fitness reply) is not enough because of the incompleteness issue and the lack of negative examples.

(b) Precision

Precision metric aims to measure the fraction of the behaviours that are presented in the log compared with the allowed behaviours on the model. In other words, a non-precise model is

the model that represents a negative trace (if the definition of negative examples are known) or an extremely anomaly process that is different than the observed ones on the log.
In our evaluation, precision is computed using [48] and [93] which counts the score of alignment between traces and model with considering illegal behaviour that is never seen in the log.

$$precision = 1 - \frac{number\ of\ observed\ events\ in\ log\ at\ a\ particular\ position}{number\ of\ allowed\ events\ on\ model\ at\ that\ position} \tag{2.2}$$

To calculate the numerator and the denominator they used the best alignment sequence to construct a tree of prefix automata that is weighted by the occurrence of a prefix of events in each position. Then the prefix automata is enriched by the edges between prefix that are allowed by the model but not observed in the log which they called it as escaping edge. The method can help in identifying the set of observed behaviour besides the set of invalid ones that have generated from the model. An example of the precision calculation is adopted from [2] and explained here.

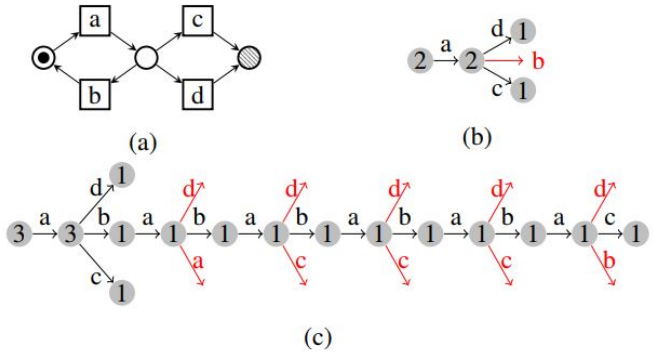


Figure 2.3: Precision calculation example adopted from [2, pg.5]

Suppose we have the model M in (a), log L_1 includes two traces [(a,c), (a,d)] that are shown in the prefix tree in (b) and log L_2 includes three traces [(a,c),(a,d),(a,b,a,b,a,b,a,b,a,b,a,c)] that are presented in the prefix tree in (c). The red edges represent the moves that are allowed by the model only and not observed in the log. The gray circles of the prefix automata are weighted by the number of tokens that enabled the move. Based on the model M and the precision formula in (2.2), the precision (L_1, M)= $\frac{2*1+2*2+1*0+1*0}{2*1+2*3+1*0+1*0}$= $\frac{6}{8}$= 0.75 and precision (L_2, M)= 0.714

Models with low precision allow a high number of unobserved events in a particular move, however, models with high precision only permit observed events. As in data mining evaluation metrics, the F-score measurement is used to balance between both accuracy metrics (fitness and precision) using the formula;

$$F - score = 2 * \frac{fitness\ *\ precision}{fitness\ +\ precision} \tag{2.3}$$

An attention should be paid for preferring a high precision model due to the risk of getting poor generalizable model conversely.

(d) Generalisation

Generalisation is the ability of a model to represent other possible behaviours that are not recorded in the log to cope with incompleteness of an event log. However, attention should be paid to the tolerance of behaviour generalisation to prevent modelling negative behaviour.
In our evaluation, generalisation score is calculated based on [47] where k-fold cross validation method is used. Fitness score is calculated for each time between the hold-out part and model generated by other parts. Then , the average of fitness scores is the generalisation score.

(e) Simplicity

Despite the fact that understandability is a subjective metric, a study showed that the main reason for perceived complexity is the size of the process model [94].
Although the available metrics try to help in assessing the quality of process models, this is still an open issue subjective to domain criteria.

2.9 Summary

This chapter has discussed the nature of healthcare processes and their implications on modelling the healthcare processes. A general background of process mining research in healthcare is provided. Two main areas are identified by process mining literature review as important areas which are healthcare complexity and variations analysis, which will be the focus of the following chapters. Also, this chapter explained several process mining techniques that are implemented for simple structure process and unstructured complex process to provide sufficient width of the available process mining techniques. The theoretical basics of Hidden Markov model (HMM) and how HMMs were used in some process mining research are discussed in order to formulate a potential use of this technique. In addition to, the discussion of process mining metrics for evaluating process models, which will be used later for evaluating the discovered process models.

Chapter 3

Event Log Extraction and Pre-processing

3.1 Overview

This chapter focuses on the early steps of our method, which are event log extraction and preprocessing. MIMIC-III is the centric of this chapter due to the need for using it in exploring our method and validate it. Detailed steps of acquiring event log are presented. Event log extraction and preprocessing are critical steps for process mining research and this is recognized in the 2011 Process Mining Manifesto [14] as the first challenge for process mining.

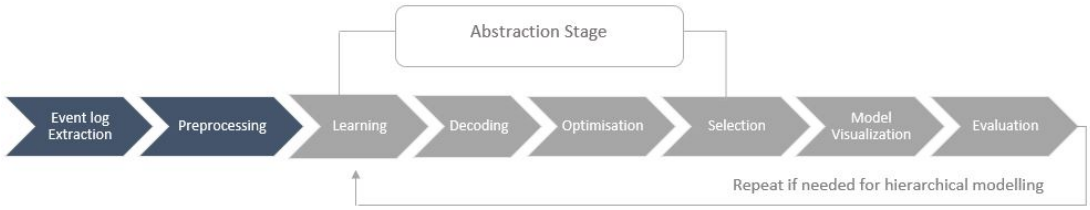


Figure 3.1: Research method and the scope of Chapter 3

3.2 Medical Information Mart for Intensive Care (MIMIC-III)

MIMIC-III (Medical Information Mart for Intensive Care)[22] is a publicly available medical research database of de-identified records of patients who were admitted to the Beth Israel Deaconess Medical Centre in Boston, USA between 2001 and 2012.
The MIMIC-III database is integrated from multiple sources which include the hospital electronic health records, social security administration death master file and two distinct critical

information systems that are called Philips CareVue and iMDSoft Metavision. The different data structures between the two critical information systems used by the hospital have largely been resolved at the database integration stage. It is an important medical database that provides free access to researchers under agreement licenses which prohibit any attempt to re-identify patients. Different types of medical data are available, such as readings of vital signs, medications, laboratory tests, nurses and physicians observations and notes, fluid balance, diagnosis and treatments codes, care giver information, length of stay and time of death.
The data comprise 58,976 hospital admissions, and 46,520 distinct patients. 55.9% of the patients are male and 44.1% are female. There are around 380 types of laboratory measurements and 4,579 types of Intensive Care Unit (ICU) charted observations, such as heart rate and blood pressure. The admissions cover five critical care units which are the Coronary Care Unit (CCU), Cardiac Surgery Recovery Unit (CSRU), Medical Intensive Care Unit (MICU), Surgical Intensive Care Unit (SICU) and Trauma Surgical Intensive Care Unit (TSICU).
According to [23], the MIMIC dataset has been used in 134 publications mostly describing data mining and machine learning approaches. None of these have described a process mining approach. In this chapter and as published in our related paper [20], we describe how we have used the MIMIC-III database to extract and process mined an event log in order to explore patients pathways for diabetes patients as a precursor to further work in diabetes. Further work done by our research group published in [95] to offer a structured assessment of the data quality issues related to process mining of MIMIC-III. In addition to the work in [96] which explores the potential of using MIMIC-III for getting insights on the cardiovascular disease trajectories using a process mining approach.

3.3 MIMIC-III and process mining

MIMIC-III can be used as a rich data source for process mining applications because it has many records with timestamps that can be extracted as medical events. There are 16 out of 26 tables in MIMIC-III database that contain medical events. These tables are used as a healthcare data reference model, which is discussed in the following section, for our healthcare process mining research.
In our earlier paper [20] we have mentioned that, in order to respect patient confidentiality the MIMIC-III dataset de-identification process included obfuscation of dates. The dates of all events have been shifted into the future using time offsets randomly generated for each patient. This approach preserves the time intervals and ensures the sequence of medical events are internally consistent, but it means that certain process mining analytics approaches such as looking for arrival time bottlenecks cannot be used.
There are two main data types for time attributes in MIMIC-III which are chart time and chart date. They provide different time resolution of the event, for instance, the chart date field has date only without time, this is because the accurate time for that event is not known, whereas

chart time field has date and time with hour, minute and second of that event.
Most of chart time fields are recorded in the database with two columns, store time and chart time. In healthcare processes, observations are usually charted and then validated by a care provider such as a nurse. The validation process usually happens within an hour [22]. Therefore, chart time is the time when an observation is charted while store time is the time when the observation is validated. In the scope of this research, we use chart time as the event time because it is the closest to reality according to MIMIC-III documentation [23].

3.4 The healthcare reference model

A healthcare data model is a model that shows the relation between tables in a medical database that may contain healthcare events. The data model is significant in process mining research because it helps to extract event logs and to understand process oriented questions [5]. We developed a healthcare data reference model by analysing the MIMIC-III database and using table descriptions based on [22] and [23]. Figure 3.2 shows the Entity-Relationship(E-R) diagram we constructed for the MIMIC-III database using PostgreSQL editor.

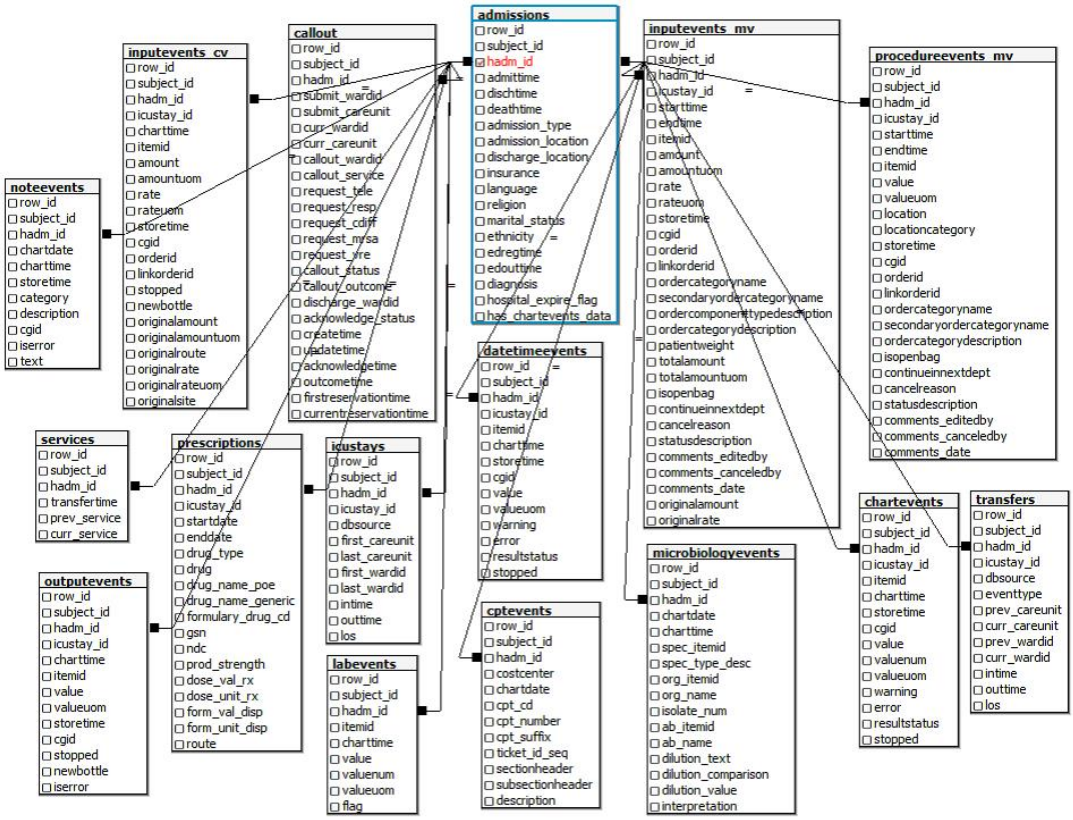


Figure 3.2: MIMIC-III data reference model generated in this research

The relevant healthcare events in our data model can be categorized into six groups of events which are administrative events, charted events, test events, medication events, billing events and report events.
In the following section, a description of the six event categories is provided along with a brief description of the sourced tables, for more detail about the tables the reader may refer to [22].

1. Administrative events identify patients admission pathways which show if a patient has been admitted from an emergency department or the patient has a pre-arranged admission. Also, administrative events include all patient transportation activities during their stay in different care units of the hospital through to a discharge event. This group of events is located in Admissions, Callout, Transfer and ICU stay tables.
   Admissions table : holds demographic information about the patient, admission time, emergency department (ED) registration time 'edreg', emergency department out time 'edout', discharge and death time, discharge and death time.
   Callout table : contains information about the time of discharge request and the time of the request outcome if it is fulfilled or cancelled.
   $Transfer\ table$: holds information about patient transportation such as the time when a patient is moved in or moved out of different wards which include different critical care units.
   ICU stay table : this is a sub-table from Transfer table especially for patients' transportations in Intensive Care Units (ICUs).
   Services table : shows which medical service that a patient was admitted for. This table is a sub-table from Transfer table.

2. Charted events contain all bedside observations that are related to vital signs measurements such as heart rate and blood pressure or other care intervention. This group of events is stored in the Chart-events and DateTime-events tables.
   $Chart-events\ table$: has all patients charted observations. There are more than 4500 types of charted observation. The table includes information about the time when an observation is taken and the time of observation validation performed by clinical staff.
   $DateTime-events\ table$: this table contains the observation date of particular interventions such as dialysis or insertion of lines.

3. Test events correspond to all tests that have been measured on the patient such as laboratory tests and test results. This category of events is captured in Out-put-events, Microbiology-events and Lab-events tables.
   $Output-event\ table$:has all output measurements for example, urine or blood. This table stores the time and value of the output measurement when is taken from the patient.
   $Microbiology-events\ table$: this table contains information about tests and antibiotic sensitivities.
   $Lab-events\ table$: this table has around 380 items for measurements some of them

related to hematology and chemistry. It records output and microbiology results.

4. Medication events include prescribed medication and intravenous medication. These events can be extracted from Prescription and Input-events-CV and Input-events-MV tables.
   Prescription table : this table contains information about when a drug starts and ends besides prescription order if it is needed.
   $InputCV - events\ and\ InputMV - events\ tables$: inputs are any fluids that can be given to the patient such as oral or tube feeding and intravenous medications. Input events tables are generated from different healthcare information systems (CareVue and Metavision) but both contain information about the time when a medication intake is occurred, for example enteral feeding is recorded and its value. Some more transactional events are supported by Input-events-MV table such as the time when intake is ended or an intake order is updated.

5. Billing events contain a list of medical procedures that are performed on patients that are used for billing services. Billing events can be extracted from CPT-events table.
   $CPT - events\ table$: this table has a list of Current Procedural Terminology (CPT) codes for medical billing purposes. It contains information that shows the time of performed procedures.

6. Report events include different types of reports such as nurse notes and radiology notes. Report events are captured in the Note-event table.
   $Note - event\ table$: this table has information about different types of notes, the date of reported notes and the ID of the caregiver who reported it.
   It should be noted that, these events are distributed in various tables however, all tables have the basic requirements of process mining such as, a unique subject id, which corresponds to patient id, and a unique admission id, event, event time, some event attributes and some resources are associated with events which can be generated from the care-givers table.

The following Table 3.1 provides a summary of process mining principle components in MIMIC-III.

Table 3.1: Summary of process mining principle components in MIMIC-III

Table	has timestamps		has duration	has observed item id	has care giver id
	Time and date	Date only			
Admissions	yes		yes	yes	no
Chart-event	yes		no	yes	yes
Input-CV	yes		no	yes	yes
Input-MV	yes		yes	yes	yes
Output	yes		no	yes	yes
Lab-event	yes		no	yes	yes
Prescription	yes		yes	yes	no
Note-event		yes	no	no	yes
Call	yes		yes	no	no
CPT-event		yes	no	yes	no
Procedure MV	yes		yes	yes	yes
Transfer	yes		yes	no	no
ICU stay	yes		yes	yes	no
Service	yes		yes	no	no
Date-time event	yes		no	yes	yes
Microbiology	yes		no	yes	no

3.5 Data acquisition from MIMIC-III

Although many modern business information systems automatically generate event logs, there are some information systems, including electronic health records that store process activities implicitly and consequently need a method for event log extraction.

The MIMIC-III database is constructed from healthcare information systems that are not process-aware oriented systems. It is an object-relational database that is built using a PostgreSQL database management system and we have therefore extracted the event log using SQL queries. Figure 3.3 shows an overview of data acquisition from MIMIC-III. The healthcare data reference model is used to guide event log extraction.

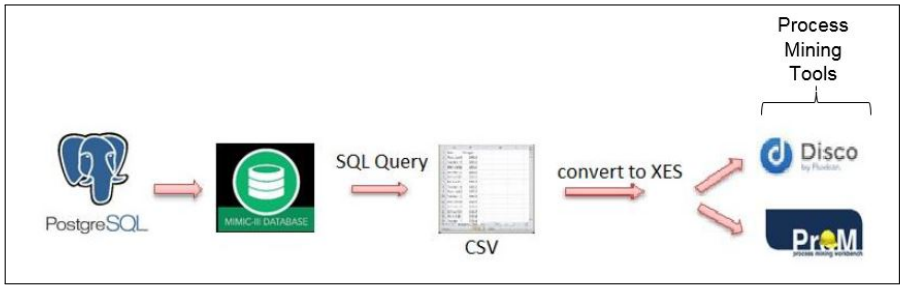


Figure 3.3: An overview diagram of getting data from MIMIC-III (XES is eXtensible Event Stream, which is the standard format for an event log)

Getting the event log from MIMIC-III can be done in two steps, the first step is creating event log from MIMIC-III and the second step is extracting event logs for specific cohorts of patients. This two steps approach is efficient because it reduces the time for multiple extractions of different groups of patients and provides efficient storage in the database. This steps are explained in detail below.

3.5.1 Creating an event log from MIMIC-III

Creating event log from MIMIC-III can done through several steps:

1. Access and install MIMIC-III
   In order to install MIMIC-III we need to get access to its server. This step requires passing a compulsory online ethics course as a preliminary step to ensure data confidentiality. For installation follow the instruction provided in
   (https://mimic.physionet.org/gettingstarted/access/).

2. Create one table that includes all healthcare events
   There are 16 tables can be used for extracting healthcare events that patient went through. Combining these events based on their temporal order will form complete process of healthcare service. The idea of extracting healthcare processes from MIMIC-III aims to provide one big table that holds all process mining essential components as shown in 3.1. The correct format of the event log is each row represents one event therefore designing the big table as this structure is essential.
   An SQL example for creating the main log table is shown in Figure 3.4.

```
CREATE TABLE "MIMIC-III-Log"
(
    hadm_id integer,
    subject_id integer,
    event character varying(200),
    start_time timestamp(0) without time zone,
    end_time timestamp(0) without time zone
    cgid integer DEFAULT 0,
    icu_id integer DEFAULT 0,
    cost character varying(200) DEFAULT 'NA'::character varying, // character because it can represent integer and double precision as in los.
    item_id integer DEFAULT 0
)
```

Figure 3.4: An SQL example for creating the main log table

3. Import the data iteratively into this table using the next query for each table that is in the reference model. Each table has its own event attributes such as cost, item id and duration. The order of the attributes is important when inserting the data into the log table. An SQL example for inserting events from admission table to the main log table is is shown in Figure 3.5.

```
insert into "MIMIC-III-Log" (hadm_id , subject_id , event , start_time  ) // the order is important must be matched.
select distinct hadm_id , subject_id , 'EDout' as event , edouttime
from mimiciii.admissions
where edouttime is not NULL  // this query for events located in the admission table .
```

Figure 3.5: An SQL example for inserting events to the main log table

4. Explore the event log from the created log table then export it to csv file. It should be noted that, in this step we have created event log for all patients in MIMIC-III with all diagnoses. Extracting a specific cohort of patients should be done in a separate step as will be discussed in the next section. General event log characteristics after creating the main log table is shown in Table 3.2

Table 3.2: General statistics of the created log table

patients	admission	emergency registration	death
46,520	58,976	30,877	5854

3.5.2 Extracting an event log for specific cohort of patients

In order to extract an event log from the created log table we need to do two steps:

1. Extract patients IDs of a specific cohort of patients from MIMIC-III. In MIMIC-III a unique ID can be either patient id 'subject id', admission id 'hadm id' or a combination of both. Selecting the IDs depends on the research questions.

2. Extract all healthcare events from the created main log table using the unique IDs.
   The extraction of healthcare process from the created log table is a straightforward step after getting the required patient IDs.

Regarding the first step, there are different ways to extract patient IDs from MIMIC-III as shown below:

(a) Extraction using free text in the admissions table
In the *admissions* table there is a column called *diagnosis* where a preliminary diagnosis is recorded. Diagnosis is recorded as free text which may require a text retrieval approach to avoid typing mistakes. An example of SQL query that is used to extract event logs using diagnosis column and admission id is shown below in Figure 3.6.

```
SELECT hadm_id
FROM
mimiciii.admissions
WHERE
(diagnosis like '%CONGESTIVE HEART FAILURE' or
diagnosis like '%congestive heart failure'
);
```

Figure 3.6: An SQL example for extracting using free text in admissions table

Using text matching as in the previous extraction method might not be efficient. This is because all cases that are extracted using text matching are correctly diagnosed with congestive heart failure, however, there might be other cases have congestive heart failure but it was recorded with other description for example, heart failure or more general description as weakness.

(b) Extraction using International Classification of Diseases version-9 (ICD9)[1]
For more accurate extraction, MIMIC-III has two tables are called $diagnoses - icd$ and $d - icd - diagnoses$ they include all information needed for patients diagnosis.
The diagnosis is coded using a standard coding system known as the ICD9 code. We can use both tables for extraction where $d - icd - diagnoses$ is a dictionary table that contains short and long title of the diagnosis and $diagnoses - icd$ links the diagnosis with patient table.

In the scope of the thesis, this method is used and we have extracted two different groups of patients from MIMIC-III as examples:

• **Diabetes type II patients**
In the first example we were interested to apply process mining on diabetes patients. We used ICD9 code to extract the log of diabetes type II patients. A list of ICD9 codes that is used for extraction is available in (Appendix A). However, the pathway of diabetes patients that includes regular checks and other clinics events can be found in primary care units such as general practise. In contrast, MIMIC-III, which is our data source, is considered as a secondary care unit that contains medical data for patients, who have admitted to the hospital. The event log of Diabetes type II patients is used in this chapter as a running example to demonstrate the extraction and preprocessing methods.
• **Colorectal cancer patients**
The second example is the event log of Colorectal cancer patients. Selecting Colorectal cancer patients as a case study is because at that time of the research we wanted to have a potential domain expert who can evaluate our method later after process discovery. We extracted this log using ICD9 code and a list of ICD9 codes that is used for extraction is available in (Appendix B). This case study will be used later in Chapter 4.

3.6 Different approaches for event log pre-processing

In this section, we demonstrate two possible approaches of event log preparation in order to provide a baseline event log for mining healthcare processes. These steps fall into two approaches which are aggregation and temporal approaches. The aggregation preprocessing steps try to prepare the order of events and it affects on the sequence of the process. On the other hand, the temporal preprocessing steps aim to prepare and clean the time aspect of events.
The motivation behind these methods is to prepare an event log with reasonable quality that helps in process modelling and in the same time without losing a lot of information about the process.

[1] http://www.icd9data.com/

3.6.1 Aggregation approaches for event log pre-processing

This method aims to improve the sequential order of the events which includes two steps; solve batch events and mapping fine-grained events into main event.

(a) Event log manipulation: solve batch events

There are some data quality issues related to MIMIC-III data such as missing accurate timestamps. This issue may be resulted from batched events. Batch processing is the execution of several events at once and recording them with the same time, for example a group of laboratory results received at the same time. The issue of batch processing also leads to a huge number of fine-grain events that increase process model complexity. In our data model, the tables Chart-events and Lab-events contain a large number of batch events which should be addressed as a preliminary step for mining patient pathways.
Each patient in the ICU has been checked on a regular basis at varying intervals. The different measurements that are taken in each check have been recorded with the same time. For process mining purposes we are focusing on the process of charted observations regardless of which items are checked therefore, all items are consolidated into a single charted event. Our hypothesis is that handling batched events as a single event simplifies the process model and improves process mining quality.
This problem is addressed in the extraction stage. Batched events are re-extracted with the same event label. The extraction includes tables that have batched events such as chart-event and lab-event. More precisely, for different chart measurements in the chart-event table such as Calcium, Glucose and Platelet count are all extracted under the name of Chart event. An SQL example for re-extraction of batched events is shown in Figure 3.7.

```
select  distinct on (charttime) charttime, hadm_id,   subject_id, icustay_id, itemid , cgid , 'Chart' as event
from mimiciii.chartevents
```

Figure 3.7: An SQL example for avoiding batch events

This method has significantly reduced the number of events which in turn reduced model complexity. It should be noted that, reducing the number of events using this method does not lead to significant information loss for our purpose in this research. We believe that, from a process mining perspective, the exact name of measurements in the ICU is less important when we aim to mine the general abstracted process model. We are able to capture the events occurred in chart-event and lab-event tables.
Although this method reduces the number of activities and events, the variation of patients pathways is still extremely high and the event log needs further manipulations.

(b) Event log manipulation: mapping fine-grained events into main activity

Another data quality issue in MIMIC-III is the different level of granularity of recorded events. The relation between these events can be represented as ontological events which have a semantic

relation with a main activity. For example, an admission activity can have a number of events where the patient may have been admitted into different wards such as Medical Intensive Care Unit (MICU) or Coronary Care Unit (CCU). Our hypothesis is that mapping fine-grained events into main activity will simplify the patient pathway model and reduce event numbers to help finding interesting patterns.
Using our data model, the categories of fine-grain events are relatively limited for some tables. Ontological events are located in Admissions and Transfer tables. Mapping the fine-grain events into main activity was done using the *Add Mapping of Activity Names* log enhancement filter in ProM. The events are mapped into main activities as illustrated in Table 3.3.

The results of this experiment shows that the number of different types of activities was

Table 3.3: Mapping ontological events

Ontological events	**Mapped activity**
admit CCU	Admission
admit CSRU	Admission
admit MICU	Admission
admit SICU	Admission
admit TSICU	Admission
transfer CCU	Transfer
transfer CSRU	Transfer
transfer MICU	Transfer
transfer SICU	Transfer
transfer TSICU	Transfer

reduced by nearly half of the previous processing step. Also, the number of events was reduced and consequently the mean of events per case is reduced likewise.
On the other hand, the number of process variations remained high and was not affected by mapping fine-grain events.

3.6.2 Temporal approaches for event log pre-processing

Outliers events can be defined as events that prevent capturing clear patterns; such events affect the quality of process mining efforts. Repeated events, which known as duplicate tasks, occur when the same event type has been executed multiple times in the same case. In critical care, for example, the incidence of repeated events is high because events include periodic monitoring (known as charting) of heart rate, blood pressure and other vital signs.
This method aims to improve the temporal aspect of the events. There are three temporal aspects of healthcare events which are the recorded time resolution, event duration and event interval. The following section will discuss these aspects and how to tackle them in detail.

(a) Recorded time resolution

An event can be stored with different time resolutions. It could be stored with a timestamp that shows the date and time or date only. Also timestamps can be stored with hours, minutes and seconds. This is considered as data quality related issue which has a strong impact on the quality of process models as mentioned by [95].
For instance, mining the process of a group of events with inconsistent temporal resolution can produce a misleading process model. This is because inconsistent temporal resolution may change the actual order of the events. In MIMIC-III, for example, the Prescription event is stored with date only while other event types are stored with different resolution that includes timestamps of hours, minutes and seconds. Therefore, the process model will allocate the Prescription event an inaccurate order.
Getting the accurate time for Prescription events from MIMIC-III database is not applicable because this depends originally on the storage schema of the MIMIC-III database where the field for storing a Prescription event is defined on a Date format only.

(b) Duration of care activity

The duration of an activity can be defined as the elapsed time since the activity started to the end of that activity. It is a feature for an activity. Some papers refer to it as execution time [97].
In MIMIC-III there is another category of fine-grained events besides ontological events which are transactional events. A transactional event is an event that provides information about the duration of an activity - when it starts, updates, comments and finishes.
This type of event is very common in healthcare processes for example, the process of transferring a patient inside a hospital which starts when a nurse creates a call for transfer, the call might be updated or cancelled, then the call should be acknowledged and the outcome should be recorded.
Transactional events are located in the Call and Input tables. Mapping these fine-grained events into the main activity was done using the *Add Mapping of Activity Names* log enhancement filter in ProM as presented in Table 3.4

Table 3.4: Mapping transactional events

Transactional events	Mapped activity
call create	Call
call update	Call
call acknowledge	Call
call outcome	Call
call first reservation	Call
call current reservation	Call
input start	Input
input store	Input
input comment	Input
input end	Input

(c) Interval of care event

The interval of care event can be defined as the time gap between two events. Interval time is a feature of an event. There are different ways it can be used to leverage this feature. In the following section, we investigate the potential of using the interval feature of same type events to reduce model complexity.

3.6.3 Interval-based event pre-processing

From a process mining point of view, repeated events is a significant confounding factor that can prevent generating useful models [44]. Typically, the handling of frequently repeated tasks has been addressed in a model discovery phase [98] [99] [100] however, most current methods are tied to specific process discovery algorithms which restrict more general use.
Dealing with repeated events as a preprocessing step has received relatively little attention in the process mining community. Moreover, to the best of our knowledge, no existing work has tackled variation reduction of repeated events using events temporal patterns. Although there are around 20 plugins in the ProM (version 6.8) process mining tool for log preparation, only two filters can be used for cleaning repeated events. These filters are called *Merge Subsequent Events* and *Remove Event Type*. They help to reduce the number of events however, no attention is paid for preserving time information about merged/removed events.
A few papers in the process mining literature have addressed repeated events as a preprocessing step. In [101], the problem of repeating tasks was addressed by refining events labels in a pre-process stage. This solution labelled repeated events based on its context for instance, 'payment' events can occur at the start of a process instance and/or at the end. Although this approach adopted accurate steps for detecting repeated events, the method is not applicable in the case of large amount of repeated events, such as those we found in healthcare data, because it increases the number of distinct events.
Two papers [102] [103] have mentioned the idea of merging repeated events into one single

event. This approach is implemented in ProM as an event log enhancement filter named *'Merge Subsequent Events'*. It aims to merge consecutive events of the same type. The merge subsequent events filter has three options of merging which are merge by keeping the first event, merge by keeping the last event or merge by considering the first as start time and the last as end time. Using this method helps to reduce the number of events however, there are a number of limitations to be discussed. The first and second options of merging ignore the time aspect between events and concentrate on reducing the number of events at the cost of losing time information. The third type of merging may result in misleading event duration. In the following section, our aim is to improve on these methods to address the specific challenges of remove outliers in healthcare periodic events.

3.6.4 The rationale for an interval-based event selection method

In this section, we define outlier events based on the time interval between events. Our starting assumption is that an event is regarded as an outlier if it occurs more frequently than a threshold interval determined from the central tendency and measure of dispersion of intervals for that event, as described in our work [20] .
We take into consideration that process mining focuses on capturing events that comply with the mainstream process. For instance, in the case of blood measurements, two successive measurements that occur within a short interval may occur because of an error in the measurement value. Therefore, removing one of those events will not lead to information loss as both events correspond to the same observation. This assumption is supported by some data observation as shown in the following Table 3.5

Table 3.5: Example of events from Input table in MIMIC-III

Admission id	Time		Item-id	Amount	Care giver id	Status
101659	**2137-02-27**	**23:00:00**	**221749**	**1.400105**	14953	changed
101659	2137-02-27	23:00:00	225158	5.833345	14953	changed
101659	**2137-02-27**	**23:35:00**	**221749**	**5.603825**	14953	changed
101659	2137-02-27	00:45:00	225158	23.34927	14953	changed
101659	2137-02-27	00:45:00	221749	6.970018	14953	changed

The above table shows events extracted from the *Input* table. The first and third highlighted rows belong to the same observed item where item id = 221749 for the same patient and the same ICU number. Based on Table 3.7, the interval pattern of input event type is 1 hour, the third row displays that this event occurred 35 mins after the previous one. It appears that this event is repeated because the carer has changed the amount of the intake item.

3.6.5 Method

In this section, some formal definitions are provided to avoid ambiguity in the method. The definitions are illustrated in Figure 3.8
Definition1: consecutive events (e1,e2), e1, e2 are consecutive events ∈ same event type E.
Definition2: interval i, is the period of time, the gap, between events (e1, e2).
Definition3: observed item x, is a distinguished attribute of an event e related to the item that was observed.

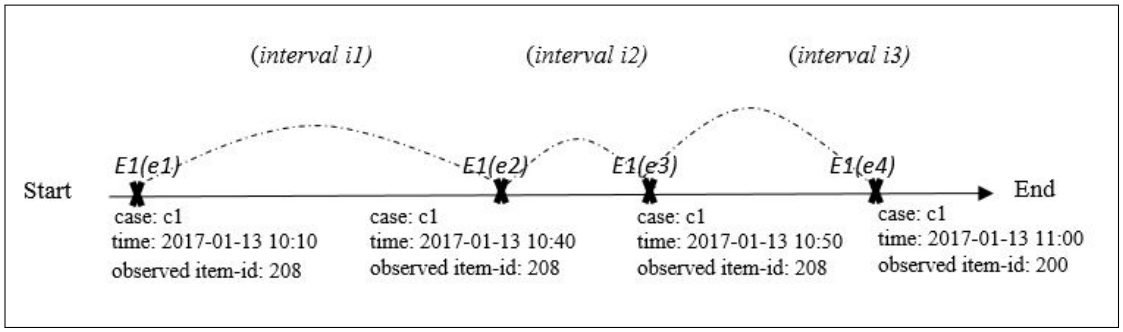


Figure 3.8: Illustration of interval-based pre-processing method

Our approach has several steps:

1. Create histograms of intervals i for each event type.
2. Use histograms to determine the central tendency and dispersion of the intervals to calculate a threshold value to identify outliers. Examples may be the mean, median and standard deviation depending on the shape of the distribution.
3. For each case c in the log, get event type E and compare the interval between its consecutive events (e1,e2) until the end of the case. The interval between each consecutive events (e1,e2) is computed by finding the time difference between e1 and e2.
4. If the interval between consecutive events (e1,e2) is less than the threshold value of that event type, remove the second event as this event can be an outlier based on our assumption.
5. Otherwise, if the interval is equal or longer than the threshold value, keep both events because they comply with the pattern.

This cleaning method aims to remove outlier events which have occurred in a time that is shorter than the selected threshold. Lets suppose three events x, y, z are consecutive care events of the periodic event 'Charting' which occurred at the times 03:54, 04:00, 04:30 respectively for the patient ID 100908 as shown in Figure 3.9.

Figure 3.9: Example of remove an outlier event from the periodic Chart event

The interval between x and y is computed which is 6 mins. This is shorter than the threshold value of 34.6 minutes that is reported in Table 3.7 for Chart event, therefore, the y event is removed as it is an outlier. After removing y, the interval between x and z is computed as they become consecutive events. The interval = 36 mins which is longer than the threshold value. Hence, event z is remained and the algorithm moves forward to compare it with the next event.

3.7 Example of event log pre-processing

The discussed preprocessing approaches are applied on the Diabetes type II event log that is extracted from the MIMIC-III database. This event log has some characteristics as presented in Table 3.6. It can be clearly seen that, the numbers of event types, event and mean , minimum and maximum number of events per case have decreased through preprocessing. This table reports event log statistics before and after applying the preprocessing steps.

Table 3.6: Event log characteristics of Diabetes type II and preprocessing steps

Event log characteristics of Diabetes type II	raw log	batch event preprocessing	mapping pre-processing	interval based preprocessing
Admissions (cases)	296	296	296	296
Patients	264	264	264	264
Variations	100%	100%	100%	100%
Event types (distinct event)	$\sim 2,300$	35	15	15
Events	$\sim 1,900,000$	252,454	210,139	208580
Mean event per case	$\sim 7,000$	853	710	705
Minimum event per case	55	28	21	21
Maximum event per case	$\sim 71,200$	10639	9246	9189

The extraction has resulted in 15 main event types which are Input, Chartevent, Output, Labevent, Prescreption, Noteevents, Call, CPTevent, Transfer, Datetimeevents, Microbology, Admission, Discharge, Edout and Edreg. Although of the clear reduction in the number of events after log preprocessing, the process variation percentage, which is computed using the following formula, is still high and reached 100% in every step of the preprocessing.

$$\text{Variation percentage} = \frac{\text{Number of process variants}}{\text{Number of total cases}} * 100 \tag{3.1}$$

This means every patient has followed a different process of care and the common pattern of care between this group of patients cannot be discovered yet. Here is more explantation about the interval based preprocessing since this step has interim results that need to be explained. The histograms are used to illustrate the interval of the periodical events. Figure 3.10 shows interval histograms for some activities such as Chart, Lab and Notes events. The threshold value is selected based on the mean for most of the activities because it represents the majority of the cases however, it depends on the interval distribution and the user preferences.

Table 3.7: Mean interval of repeated events

Care event	**Interval**
Chart event	34.6 mins
Lab event	6.0 hours
Input	1.1 hours
Note event	8.9 hours
Transfer	52.8 hours
Call	1.5 hours
Prescription	25.2 hours
CPT event	27.5 hours
Output	1.6 hours
Microbiology	66 hours

Table 3.7 shows the mean interval of repeated events in MIMIC-III. These intervals are extracted from histograms of repeated events that are presented in Figure 3.10.

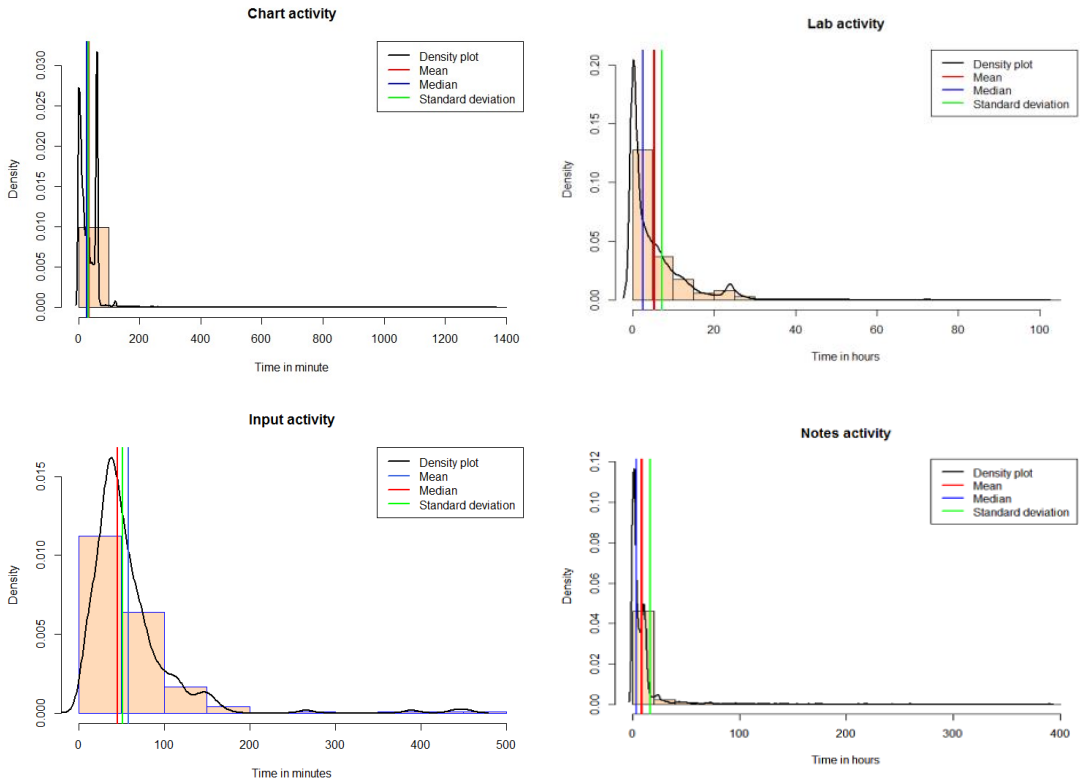


Figure 3.10: Interval histogram for some repeated events in MIMIC-III

Removing the outliers events from the event log using interval based event selection has reduced the number of events, mean and maximum events per case while other pathway characteristics such as variations, number of event types and the minimum number of events have not been affected. Moreover, this method has a different impact on different event types. Some event types have been affected strongly by removing outliers events such as Current Procedural Terminology (CPT), which is used for a payment event, where 248 outliers events are removed from the CPT event. On the other hand, Chart event and Output event have the least impact with 56 outliers events in Chart event and 64 in Output event.

Discussion

As mentioned earlier in Section 3.6, the main goal of applying event log preprocessing methods is to prepare a reasonable event log quality that helps in process modelling without losing main elements that are essential for capturing the mainstream healthcare process. Both approaches of event log preprocessing, aggregation and temporal, try to reduce the number of events in order to reduce complexity. Such approaches have reduced the number of events as reported in Table 3.6, however, these methods have failed to reduce process variations. Figure 3.11

shows that all steps of event log preparation have successfully reduced the number of events, the line illustrates the drop of events number. Unexpectedly, process variation percentage has not improved, not reduced, after log preprocessing it is still the same.

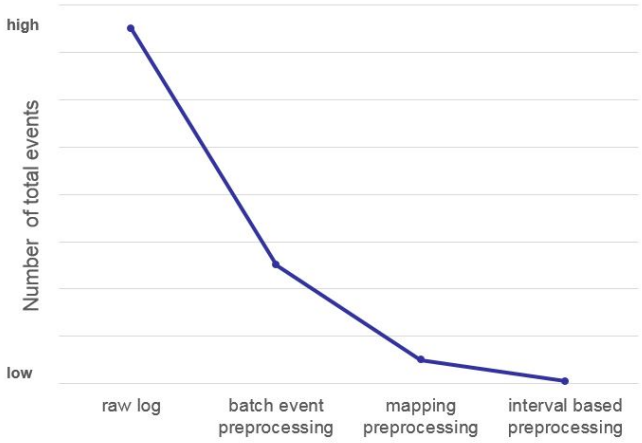


Figure 3.11: The affect of pre-processing steps on the number of events

It should be noted that, the interval based preprocessing step has improved the variation of repeated events. This can be seen in Figure 3.12 where *Input* event has the highest reduction in its variation while the *Transfer* event has the lowest variation reduction which indicates that the *Transfer* event was mostly repeated in a consistent temporal pattern.

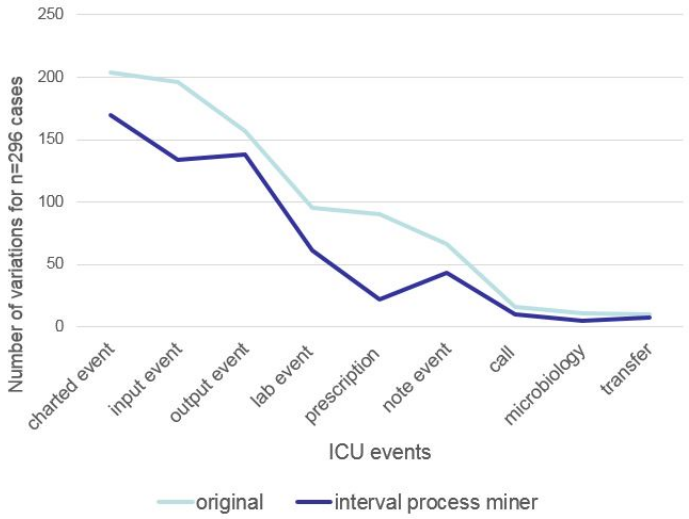


Figure 3.12: The number of variants in the periodic events in the intensive care units

The prepared cleaned event log now is presumed to be ready for process model discovery. There are different techniques in process mining that tried to discover process model of highly variable event log and address complexity by supporting events abstraction. Using these techniques on

the prepared event log is supposed to generate understandable valuable process models, however, the results were not as expected and this will be demonstrated in the following section.

3.8 The practical limitations of some techniques used for discovering unstructured processes

The aim of this section is to use the prepared event log of Diabetes type II patients for modelling the healthcare processes. Two techniques that are originally designed to cope with complex and unstructured processes are chosen. These techniques are Fuzzy miner and Local Process Models mining (LMs), that were discussed in Chapter 2. Both of these techniques support the concept of events abstraction. The results are explained below.

3.8.1 Fuzzy miner results

The fuzzy miner without abstraction has generated a complex model as presented in Figure 3.13. However, this technique supports an interactive abstraction where the nodes can be abstracted manually using a slider which provides the cut-off threshold of node significance. Figure 3.14 shows two different models (left), the top model with node significance cut-off ~ 0.5 has four events which are input, discharge, chartevent and output and two clusters of random number 18 and 19 with significance 0.051 and 0.118 respectively. The significance of cluster is the sum of the significance of related events which is basically based on event frequency. On the other hand, the bottom abstracted model has one event, input (with the highest significance = 1) , and two clusters but with different related events and significance scores which are 0.261 and 0.235 for clusters 18 and 19 respectively.

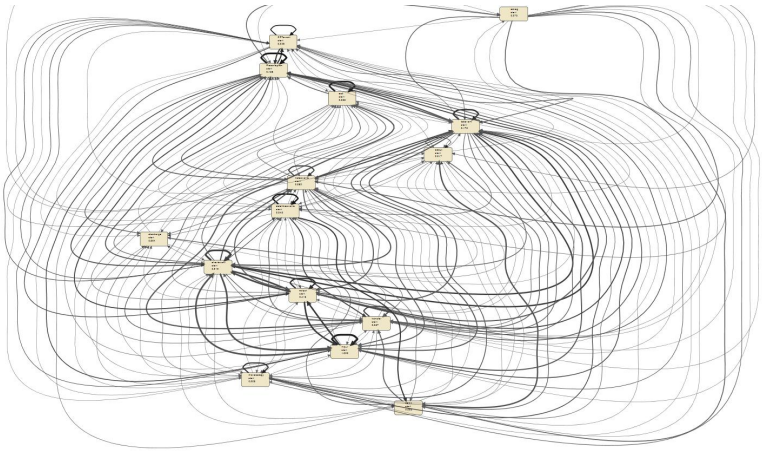


Figure 3.13: The discovered model using fuzzy miner in ProM6.8

The technique of abstracting node in fuzzy miner is simple where the high frequent event is

preserved, the less frequent and high correlated events are clustered and the low frequent events are removed. A hierarchy modelling is supported as well when clicking on the clusters. Further analysis of the events that are included in cluster 19 and 18 is shown in Figure 3.14 as well. These clusters show that the gray edges represent the links from the previous abstracted model (with 0.5 cut-off). The actual number of elements inside cluster 19 is 4 elements (Prescription, CPTevent, Edreg and Edout) while cluster 18 has 5 elements (Datetimeevents, Admission, Transfer, Noteevent and Call). The Laboratory and Microbiology events are removed after abstraction since they were not significant based on this threshold.

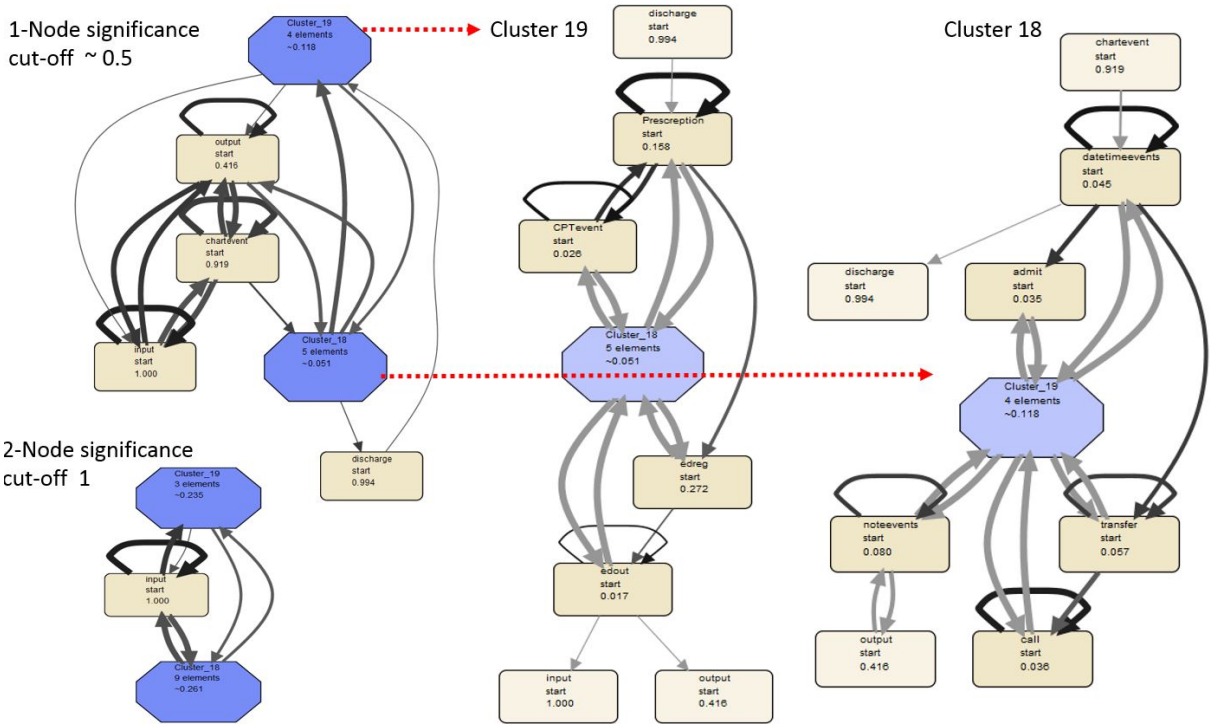


Figure 3.14: Fuzzy model with abstraction

Although fuzzy miner has reduced model complexity, it has some practical limitations can be summarized in:

1. The fuzzy miner is an interactive technique and does not discover automated process model directly where it requires manual adaptation of discovery parameters and model tuning.

2. It models the sequence of events but cannot discover different process constructors such as parallel and choice.

3. The produced fuzzy model cannot be converted to a Petri net therefore, it cannot be used

for model evaluation and conformance checking with the related event log.

4. Events are assigned into clusters in one-to-one mapping which makes the model unable to discover same events with different context. This is a major limitation where no semantic clustering is supported. For instance, the examined healthcare process here is generated form a hospital with different Intensive Care Unit (ICUs). The charted events can be recorded in different ICUs, which represent different contexts of care. Therefore, clustering events based only on the frequency as provided by fuzzy miner is not efficient for distinguishing different care contexts/states.

5. The model may start with any random node where the start node might be changed after refreshing the model and no clear start and end node are presented at the process.

6. Low frequency events are removed from the abstract model such as Laboratory and Microbiology events. We believe that, low frequency events can be significant and may have an effect on the flow of healthcare process, therefore, such events are preferable to be presented in the abstract model in order to provide comprehensive process model.

3.8.2 Local process mining results

Based on the best of our knowledge, the latest method of unsupervised pattern extraction in process mining is mining local process models that is discussed in Chapter 2. The implementation of this approach is supported by the ProM process mining tool. The aim is to discover abstract model and reduce model complexity with local process models detection. For this experiment we use the plug-in 'Mine Local Process Models' in ProM6.8.
In order to discover local models (LM), a number of settings should be set first such as; the number of local process models that will be discovered and what kind of process constructors that should be used for modelling. In this experiment, the parameters that are used before model discovery are 50 local process models and the process constructors of sequence, choice and loop. This has resulted in 50 local process model for 43 groups of events, where one group of event might be expressed using multiple local models. This will be explained in the results of LM below. The results of mining local models are ranked based on a score that is built on 4 metrics which are; confidence, determinism, language and coverage. These metrics are explained in Chapter 2 and for more information refer to [70].
Some samples of top ranked local models are presented below where, each event has two numbers in the form of (event frequency in this pattern/total number of this event):

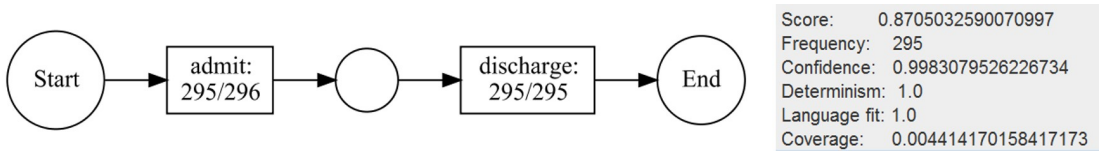


Figure 3.15: Local process models of group 1

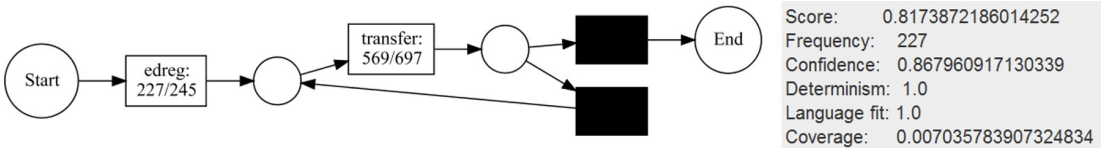
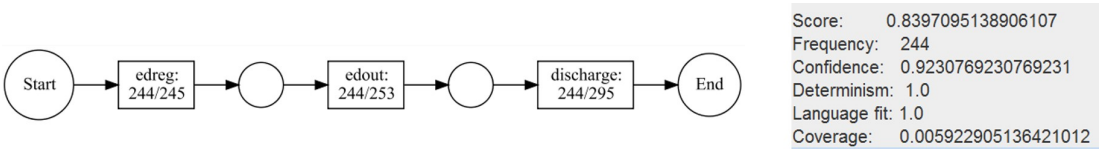
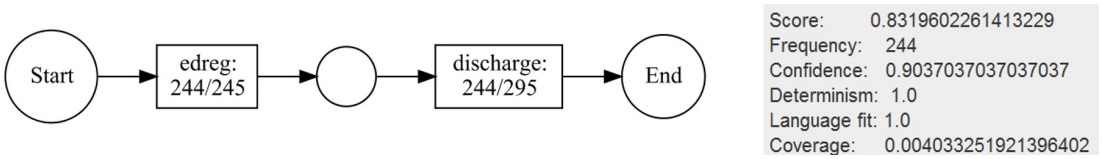


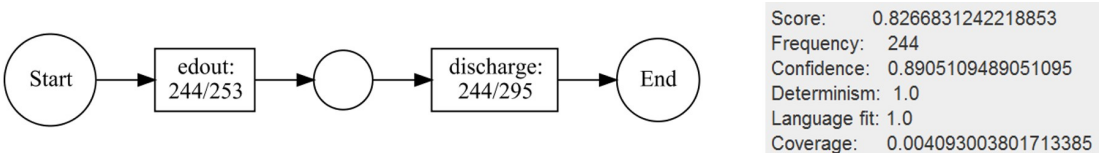
Figure 3.16: Local process models of group 2



(a) LM1 of group 3



(b) LM2 of group 3



(c) LM3 of group 3

Figure 3.17: Local process models of group 3

Figure 3.15 shows the top local process models for group 1 of events which includes admission and discharge where this local model represents the long dependency between admission and discharge. Admission event is observed 296 times in total but 295 times has occurred in this pattern. Also, discharge event is always observed in this pattern with 295 times in total. This local model has a score of 0.870 based on the different criteria such as frequency, confidence and coverage that are used in the discovery process. Figure 3.16 shows the pattern of emergency department register(edreg) then transfer to a hospital ward where a loop of transfer event is possible. The event (edreg) is observed 227 out of 245 in this pattern and transfer is observed 569 out of 697 in this patter as well.

Figure 3.17 presents the three top local models of group 3 which includes emergency department register(edreg), emergency department out(edout) and discharge events. These three local

models are ordered based on their scores which are 0.839,0.831 and 0.826. It should be noted that, similar models are discovered for the same group of events but not presented here to prevent redundancy.
It is important to note that, there are 50 local process models that are generated hence, the above models are picked manually to illustrate a sample of the results of mining local models. Although mining local model has extracted local pattern in unsupervised way and has captured the long dependency between events, it has a number of limitations can be outlined as follows:

1. Mining local process models aims to discover internal patterns which are restricted to three to five event types and cannot discover the whole process represented in a start-to-end model.

2. Mining local model requires time and careful selection of several parameters that should be set before doing process discovery. Four main parameters which are the number of local process models to be discovered, operators type (whether sequence or loop and so on), maximum and minim number of pattern occurrence and some temporal constraints such as time gap between subsequent events. These parameters have a heavy impact on the resulted models.

3. Although mining local process provides unsupervised pattern extraction, it generates a large number of local process models with the similar score and may have overlapping events. This requires a domain preference to select the most representative local model for each group of events.

4. The local process model cannot be used for hierarchy modelling such as the one that is supported by fuzzy miner.

5. Local process mining can be inefficient method and cannot scale with large event log. We have tried using this technique with the colorectal cancer patients logs however, the local models could not be discovered due to multiple crashes of ProM tool that have happened during local models discovery.

Exploring these two current techniques of process mining has emphasized the need for developing more robust abstraction method that can do the following; supports an automatic abstraction for start to end process model and discovers the general care of pattern for a complex large event log with the ability of handling process variations. Also, the required method should be able to generate a process model that can be evaluated and assessed within the available process mining frameworks in addition to distinguishing care events that may occur in different contexts of the process.

3.9 Conclusion

This chapter focused on the early steps of our methodology which is the extraction of event log and the pre-processing. An online medical database known as 'MIMIC-III' is used for the first time in this research and process mining principle components in MIMIC-III are discussed as well. Two main event log pre-processing approaches are explored which are aggregating and temporal preparations. These approaches have reduced the number of events in healthcare event log, however, further method should be used to reduce healthcare process variations and hence, improve complexity. Two techniques for process mining were used to discover understandable process model which are fuzzy miner and local process mining. Both methods provided a mean of event abstraction, however, they have some limitations. The major limitations of using fuzzy miner is the inability of using such model for evaluation since the model cannot be converted to a Petri net, which is the formal model for conducting conformance checking and model evaluation. In addition to the shortcoming of one-to-one mapping that prevented capturing different contexts based on the surrounded events. Local process mining needed a domain expert for selecting the representative local model to be used for abstraction. Also, the discovery of internal patterns does not allow for abstracting start-to-end process models. Hence, there is a need for a new method that can discover comprehensible start-to-end process model in an unsupervised way and can be used for evaluation as well. In the next chapter, a machine learning approach will be explored to apply unsupervised healthcare events abstraction to discover a start-to-end pattern of care and reduce process complexity.

Chapter 4

Machine Learning Approach for Healthcare Process Abstraction

4.1 Overview

The aim of this chapter is to investigate the applicability of using a machine learning technique, in particular HMMs, for discovering the general process model. In this chapter we present the method of using HMM for process abstraction in three main steps; learning, model selection and decoding, as shown in Figure 4.1. These steps are reordered in our research method as will be explained throughout the thesis. Some well-known information criteria metrics that are used in literature for model selection are described. Then different empirical results are discussed which in turn provide some evidences of a number of practical issues that can be found in HMM models, which are selected as best models. This chapter helps in identifying the limitations of the adopted metric that can be used for selecting the best number of states in HMMs.

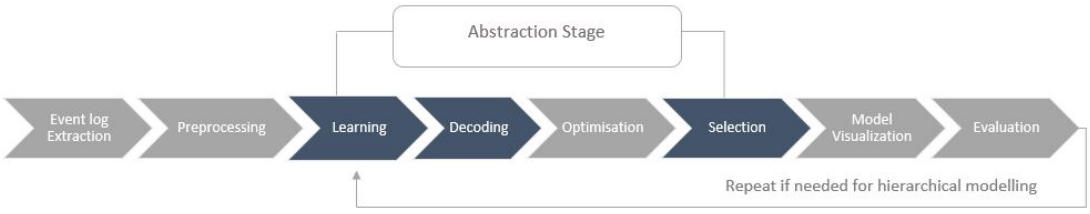


Figure 4.1: Research method and the scope of Chapter 4

4.2 Detection of healthcare hidden processes

Machine learning approaches such as Hidden Markov Models (HMM) may assist in healthcare process refinement through detecting and abstracting healthcare processes. We have searched

the literature as discussed in Chapter 2 (section 2.7.3) and we found that, HMMs have been explored in process mining research but mainly advocated for patient pathways clustering purposes. However, we argue that these models can also be utilized for detecting hidden processes and consequently, help in events abstraction and visualizing abstracted healthcare process model. In this chapter, we explore the use of an unsupervised method for detecting hidden healthcare sub-processes using HMMs, in particular the Viterbi algorithm.

4.3 Method

The following method of modelling healthcare process with state abstraction has been applied in process mining research [86] for mining customers intentions, however, it is originally adopted from other fields that have used HMMs for sequence modelling in speech recognition and bioinformatics [104].
In order to model healthcare process using state based abstraction we need to go through three main stages:

1. Learning:
   The algorithm that is used for learning is the Expectation-Maximisation (EM) as mentioned in Chapter 2. There are two parameters that should be set before starting model learning stage which are; maximum iteration and change tolerance for two iterative models. Other parameters such as transition and observation matrix are initialised randomly several times to avoid trapping in local minimum. The learning stage aims to populate a number of HMMs and this requires an iterative increasing of the number of hidden states where it starts from minimum 2 to maximum the number of event types in the process.

2. Selection:
   Selecting the best model based on different information criteria metrics, that will be discussed in the following section.

3. Decoding:
   Decoding is done by running the Viterbi algorithm over sequences to see the underlying sequences of the hidden states that are likely generated the processes.

   Model selection is a critical step built on comparing different models, hence, the following section elaborates on some well-known metrics that are used for HMMs selection.

4.3.1 Metrics for HMM models selection

In model selection, the log likelihood can be used when models have the same number of states however, models with different number of states can be compared using information criterion measures based on penalised likelihood such as AIC, BIC, ICL and cross validated

likelihood [105] [106]. These metrics are typically used for selecting models that are estimated using maximum likelihood technique. In general we can use any one of them since all of these measures try to balance between model likelihood and model complexity [107] [108]. In our experiments we are focused on BIC since it is the most widely used metric in literature and has outperformed other metrics in recognizing the true number of hidden states in the data [109],[110] and [111].

Akaike information criterion (AIC-1974)

AIC [112], is a metric for model selection that penalizes the likelihood of a model for having unneeded parameters, which leads to a complex model. It considers the predictive density of a model to generate the given data. Model with less value of AIC is a good model. AIC metric is presented in the following formula:

$$AIC = \underbrace{\text{-2 log(likelihood)}}_{\text{Term1}} + \underbrace{\text{2 par}}_{\text{Term2}} \tag{4.1}$$

where par, is the number of parameters in the model, which is discussed in detail with BIC metric below. In this formula, Term 1 decreases as we get a model that fits data which rewards fitting the data well. Term 2 increases as we increase the number of parameters in the model thus penalising selecting a complex model.

Bayesian Information Criterion (BIC-1978)

BIC [113], is one of the most widely used measures for statistical model selection. It is an extension of AIC. It aims to penalise the model with more parameters. Models with lower value of BIC are considered to be better [114]. There are different formulas of BIC however, we adopt the original formula that is supported by R statistical framework which can be computed as follows:

$$BIC = \underbrace{\text{-2 log(likelihood)}}_{\text{Term1}} + \underbrace{\text{log(n) par}}_{\text{Term2}} \tag{4.2}$$

Term 1 decreases as we get a model that fits the data which rewards fitting the data well. Term 2 increases as we increase the number of parameters in the model and also as the number of observations increase this penalising selecting a complex model with a large amount of data. Where; n is the sample size and par is the number of parameters. The parameters can be defined as every probability distribution that we need to estimate, in HMM case we need three main probability distribution which are the distribution of initial probability, state transition distribution and observation distribution.
The number of parameters, which is sometimes called the degree of freedom as an indication of the number of variables that are allowed to be varied, is a key term in BIC calculation and

for a model with k state and e distinct events, the degree of freedom can be calculated as:

$$par = k + k^2 + ke \tag{4.3}$$

Where, k^2 allows for staying in the same state. This formula can be simplified by considering probability redundancy constraints where we can calculate $k(k-1)$ rather than k^2 as the probability of the last element can be deduced from the total probability as suggested by [115].
Further simplification can be achieved by considering all zero probabilities as structural zeros as suggested by [25] therefore, the number of parameters can be calculated as:

$$par = size(\pi > 0) - 1 + size(A > 0) - k + size(B > 0) - k \tag{4.4}$$

Using the notations of HMM that are discussed in section (2.7.2) where, A is the transition probability matrix and B is the observation probability matrix. From above we can conclude that BIC tends to penalize an unstable model. In other words, if a model has a high number of transitions between states, this will consequently increase the number of parameters. BIC has outperformed AIC in most studies because it provides more realistic results in terms of involving the amount of data besides the number of parameters as stated in [107]. Also, the penalty impact of model complexity in BIC is stronger than AIC.
Despite the fact that BIC is better than AIC, some literature raised potential concerns about using BIC in high dimensional data [116]. They have reported that BIC cannot handle high dimensional data because the number of parameters is computed based on the initial model. The risk of this comes as some probability turn to be zero after converging in high dimensional data which results in a misleading BIC value [115].

Integrated Completed Likelihood (ICL- 2000)

This metric [117] aims to pay more attention to the underlying states when considering the likelihood of generating the data. It calculates the joint probability of the observations and states and prefers a model with high partitioning between states. The formula for ICL is presented here;

$$ICL = -2log(likelihood, states) + log(n)par \tag{4.5}$$

It supports an entropy based metric for calculating the separation between states. Although this metric seems to be applicable to our motivation in this research, we have not adopted it because this metric was not tested for practical uses in the literature. In addition to the limited understanding of the behaviour of this metric with large data, which is reported in [106].

Cross validation likelihood approach

Cross validation approach is a well-known technique for model evaluation in data mining. Selecting the number of state in Hidden Markov Model with cross-validated likelihood has been explored in [116]. This approach aims to select the best model based on its predictive performance where the data is used as one-sample-out for testing. The result of this work was promising and showed that cross validated likelihood is equivalent to BIC in most experiments, however, this method consumes more time for model selection and requires more computational resources for large data.
Despite the potential limitations of BIC with large data, it has been widely used as a standard metric for HMMs selection. Therefore, in the following section, we aim to adopt BIC for model selection and to investigate possible improvements areas to cope with its tendency for over-fitting with large space data.

4.4 The effect of high dimensional space on BIC

In high dimensional data when the number of events is large and cases have different lengths of observations, which results in a sparse matrix, BIC tends to favour overfitting models [118] [119] [120]. Furthermore, BIC highly depends on the prior estimation of the number of parameters which is sometimes not accurate [115].
There are several modified versions of BIC that try to cope with high dimensional data such as extended BIC (EBIC) that aims to increase the penalty of sample size into the original formula [121]. However, [122] has proved that all modified versions of BIC are equivalents in a very sparse search space.

4.5 Issues on models selected by BIC

Most studies that have mentioned BIC shortcomings with high dimensions data have only evaluated BIC based on a previously known number of clusters or states in the data and whether BIC has selected the true model or not.
In this research, we are motivated to provide an empirical investigation of issues that can be found in models selected by BIC. Initial results of our experiments have shown that there are three main issues found in models selected by BIC and trained with several size of event log. These issues are the existence of strong connected components, models ended by multiple similar states or non-significant (not representative) states.

4.5.1 Existence of strong connected components

The hypothesis of the existence of a higher level abstraction is presumed because of the strong connection between models states. In order to identify such issue, an established principle in

Graph theory that is known as 'strong connected components/community detection' is used. It helps in finding the best partitions between nodes, which are the states in our case, and highlighting the nodes that are linked strongly to each other. There are different algorithms that can be used to detect groups of connected nodes, however, we have used a simple widely used algorithm which is known as edge betweenness-community detection method [123].
The edge betweenness partitioning algorithm aims to find the highest score of edges betweenness (weights). The idea behind this algorithm is that, the edge of high weight represents the shortest path that links between different communities. Hence, removing the edge of high weight is the starting point of breaking the graph into distinct chunks. This algorithm runs iteratively and each time it breaks the highest score edge and stops when there are no edges left. For each iteration, a modularity score is calculated which aims to measure how well the graph structure is. There are different formulas for calculating graph modularity, however, all of these formulas aim to find the fraction of edges inside community with all possible edges between the nodes without partitioning. Therefore, the best partitioning is chosen based on the highest number of modularity between components [124].

4.5.2 Existence of multiple similar states

The second practical issue of models selected by BIC is the multiple presence of similar states. Before explaining this issue, a proper description of hidden states should be introduced.
From our investigation, hidden states can be characterised based on state type and state class. On one hand, four main types of states in Hidden Markov models are identified which are production, simple, composite and complex states.
Production state is a state has only one event type and it is called so conventionally with other literature of state modelling [125] that was identified this type. This type of state produces one single event type and is considered as the leaf of the model. Simple state, as we define, is a state has several event types but 80% of the state is occupied by maximum 2 event types. Hence we can say, every production state is a simple state but not vice versa. The third type is composite state which is a state has several event types and 80% of the state is occupied by more than 2 event types. Lastly complex state, which is a composite state that contains high variable processes and this variability makes the process complex and difficult to comprehend. Complex state will be used later to decide if a state needs further modelling (hierarchical).
On the other hand, state class characteristic which is determined by the event types (distinct events) that are included in that state. Same-class states are states constructed from the same distinct events. It should be noted that, same class states can be found in any type of hidden states, for instance, a model may have two simple states of same class.

Hence, the issue of multiple similar states can be detected through these two characteristics; state class or state type. Testing the similarity of states using state class perspective is trivial. It requires looking into the event types related to each state and if two states have identical

event types, they would be similar and called 'same-class' states. However, this perspective provides a definitive notion of similarity. For more explanation, events are assigned in a state based on probability where some events have a very small probability of association with a state. Therefore, if two states are almost identical but have difference of small probability events, these states are not similar because they are not 'same-class' states.
In contrast, detecting the issue of similar state by considering state type perspective provides more realistic similarity measure. It requires two steps which are; first, check state type, and second, examine the main events that occupied 80% of both states. Checking state type can be identified easily by plotting Pareto chart, then if the event types that formed 80% of the states are the same then these states are similar. It should be noted that, the state type perspective will be used for detecting the issue of multiple similar states in this research.

4.5.3 Existence of non-significant states

A state can be defined as significant if it is presented in most of the cases. It is a simple metric that shows the coverage of state in regard to number of cases who have activated that state. State importance metric is calculated as:

$$\text{State importance} = \frac{\text{number of cases in a state} * 100}{\text{total number of cases}} \geq threshold \tag{4.6}$$

In this thesis we set an initial threshold of state significance to be more than or equal to 50% of cases in order to see the states that represent at least half of the patients. This threshold can be customised based on users preference.

4.6 Empirical results of different space size

In this section we aim to demonstrate how BIC behaves with different sizes of event logs. Four different sizes and sparseness of event logs are investigated. The size depends on the number of cases, here patients ID, and number of events while the sparsity is counted by the number of nulls of variable lengths sequences. Hence, we have provided small, medium and large synthetic logs and one complex real event log.
Several experiments are conducted in order to characterise a good model and thus prevent undesirable ones. Initial results showed that the three discussed issues can be found in models trained with high size and sparse event log as explained below.

4.6.1 Method

Several hidden Markov models (HMM) are learned using four different size event logs. The package "SeqHMM" version (1.0.8-1) in R is used for learning. A number of random initializations were tested to avoid trapping in local minimum.

(a) Small event log

This log was created manually to reflect the healthcare processes inside a general Accident and Emergency room. It is a fictional data describes how a patient can be treated through a number of events such as, arrival to emergency room, initial assessment, seeing a doctor and other event types that are illustrated in Figure 4.2. The synthetic log consists of the following characteristics:

Table 4.1: Small size log characteristics

Number of cases	Number of event types	Number of events	Number of variants
10	9	84	6

The log has processes with similar length; minimum 8 and maximum 9. The log space has 10*9 dimensions and there are only 6 null values which means the space here has a very low sparsity. Figure 4.2 shows traces of the created process.

Figure 4.2: Generated traces of Accident and Emergency room (experiment 1)

The result of training HMM with different number of hidden states starting from minimum 2 to maximum 9, which is the total number of event types, is shown in Table 4.2 below.

Table 4.2: Training HMMs with different number of hidden states (experiment 1)

Number of states	Iterations =(MX)3000	Initial log	Final Log likelihood	BIC (Bayesian information criterion)
2	49	-188.0499	-155	394
3	**37**	**-191.1419**	**-125**	**392**
4	21	-191.237	-101	411
5	87	-192.6807	-76	437
6	63	-188.9694	-80	529
7	41	-188.3411	-54	568
8	26	-183.6116	-50	663
9	86	-186.2574	-40	756

The results in Table 4.2 show that the model with 3 states is the best model because it has the lowest value of BIC. Figure 4.3 below illustrates the events distribution over states and the transitions between the states of 3states model.

Figure 4.3: HMM of small synthetic data (experiment 1)

The numbering of the states is random and it is distributed from left to right where state 1 is the left most state. The process starts in state 3 where it has full probability (below the state) as the starting state. The model shows the healthcare process starts by the patient's arrival to the Accident and Emergency room and initial assessment. Then, the patient may be seen by a clinician or specialist. If a patient needs to be admitted to the hospital, a bed is requested for him. After that, some patients may be discharged or admitted to the hospital and leave the Emergency department.

Results

As was mentioned in the first chapter, the process has a natural flow. Generally the process follows directional steps, sequential events, until reaches the end. If we assumed there must be a mainstream pattern of the processes, the model should be able to identify the main transition points between blocks of events.

This model has several good characteristics for modelling processes:

• It has a natural flow of the process by providing a clear direction of the process or sub-process/states and the states have sequential transitions between them.

• Processes are abstracted into distinct states where each state corresponds to different blocks of events. In other words, this model tends to have high dissimilarity between states.
• There is no production state which has only one event types. States have allowed reasonable variance.
• All states are important because they cover all cases as shown in Table 4.3.

Table 4.3: State importance of the selected model in experiment1

	state1	state2	state3
# of cases	10	10	10
percentage	100%	100%	100%
state important	yes	yes	yes

(b) Medium event log

The goal of creating this synthetic log is to provide a controlled size of sparse space and increase the number of cases. This log is created using a simulation tool called "NETIMIS" [126]. The tool is developed to help modelling artificial processes for the aim of answering "what-if" questions. The synthetic log has a number of the characteristics that are reported in Table 4.4.

Table 4.4: Medium size log characteristics

Number of cases	Number of event types	Number of events	Number of variants
500	6	2348	6

The log has traces with different lengths; minimum 3 and maximum 6. The log space has 500*6 dimensions and there are 652 null values which make the space relatively sparse. A sample of the traces that are simulated for this experiment is illustrated in Figure 4.4.

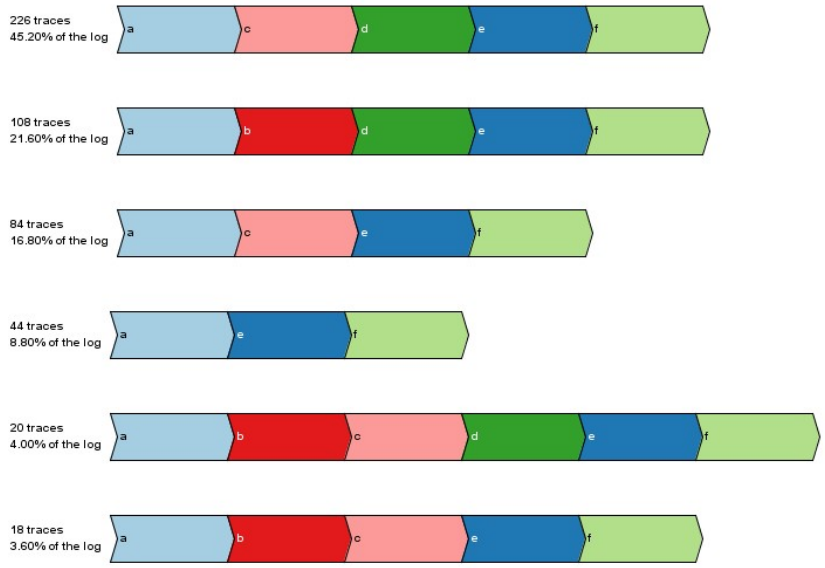


Figure 4.4: Generated synthetic traces by NETIMIS (experiment 2)

The result of training HMM with different numbers of hidden states starting from 2 to 6, which is the total number of event types, is shown in Table 4.5 below.

Table 4.5: Training HMMs with different number of hidden states (experiment 2)

Number of states	Iterations =(MX)3000	Initial log	Final Log likelihood	BIC (Bayesian information criterion)
2	70	-4512.326	-3220	6541
3	28	-4867.832	-1963	4106
4	24	-4545.877	-2013	4299
5	**13**	**-4610.926**	**-812**	**2005**
6	23	-4366.252	-812	2129

The results show that the 5states model is the best model because it has the lowest value of BIC. Figure 4.5 below illustrates the events distribution over states and the transitions between states of 5states model.

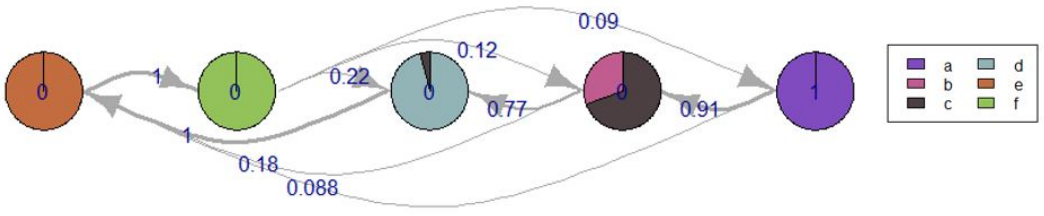


Figure 4.5: HMM of medium synthetic data (experiment 2)

The numbering of the states is random and it starts from left to right where state 1 is the state of 'e' event. The temporal order of the process started from state 5 where it has 1 probability, appeared in the centre of that state, as the starting state.

Results

It can be clearly seen that in this model when a trace reaches state 3 it will certainly transition to state 1 and next to state 2. This is because the probability of transition from state 3 to state 1 is 1 and from state 1 to state 2 is 1 likewise. Hence, we anticipate that a better generalisable state abstraction exists as suggested in Figure 4.6. We would like to see a higher level abstraction that combines state 3, state 1 and state 2 since they all linked to each other. Another possible abstraction can be revealed between state 5 and state 4 since the probability of a process that started in state 5 will transition to state 4. Although providing this kind of high level abstraction may degrade model accuracy, it prevents model over-fitting and provides better generalisability.

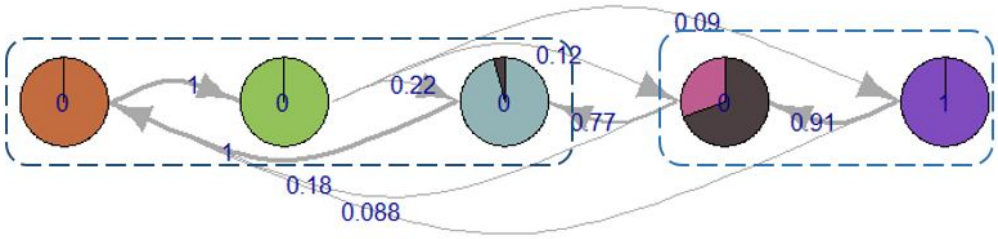


Figure 4.6: HMM of the synthetic data with possible higher level abstraction

A simple experiment has been performed to test our hypothesis of the existence of a high level abstraction between states. We used the 'strong connected components' detection approach. This helps in highlighting states that are linked to each other based on the high connectivity between them. Figure 4.7 displays two possible groups of higher abstraction and this conforms with our previous hypothesis. The first group, the red colour, combines state 1, state 2 and state 3 because they are strongly connected whereas the second group, the blue colour, includes state 4 and state 5.

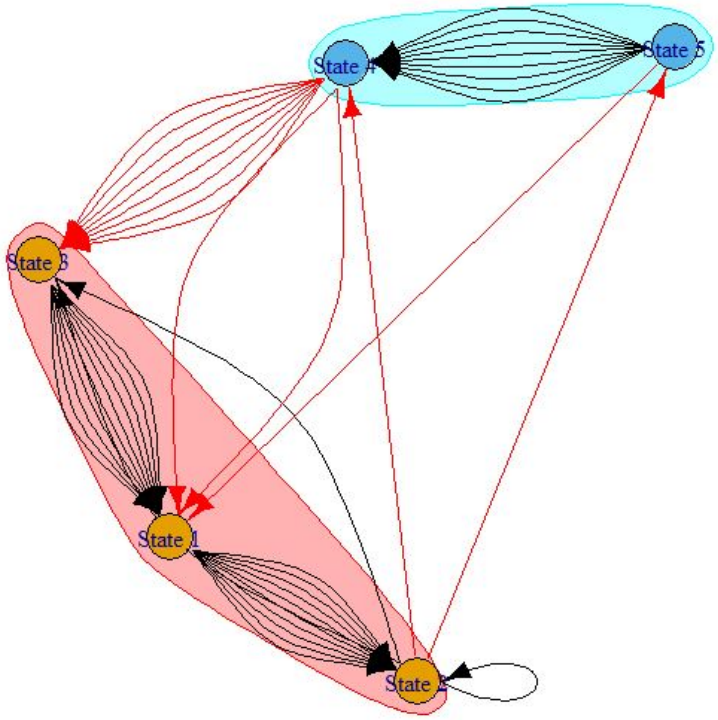


Figure 4.7: States of Figure 4.5 after using strong connected components detection

(c) Large event log

The aim of creating this synthetic log is to provide a higher process variability and sparsity than the event log used before. This log is created after adding some events using log filters in ProM tool [127] on the same event log that is used in the previous experiment (experiment 2). Increasing event numbers may come in the form of adding, swapping or repeating some events. The generated log has the following characteristics, see Table 4.6.

Table 4.6: Large size log characteristics

Number of cases	Number of event types	Number of events	Number of variants
500	6	3560	455

This log has a large number of process variants where 455 unique process instances are followed. Processes also have highly variable lengths; minimum 3 and maximum 11. The log space has 500*11 dimensions and there are 1851 null values which make the space sparse. Sample of the generated traces is presented in Figure 4.8.

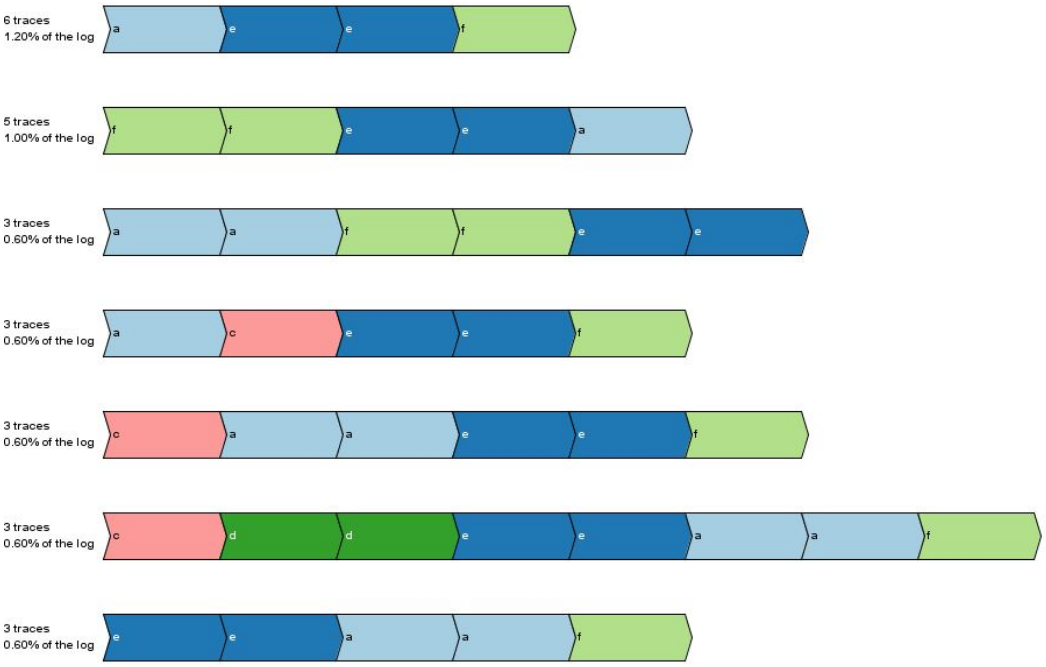


Figure 4.8: Generated traces of large log (experiment 3)

The result of training HMM with different number of hidden states starting from 2 to 6, is shown in Table 4.7 below.

Table 4.7: Training HMMs with different number of hidden states (experiment 3)

Number of states	Iterations =(MX)3000	Initial log	Final Log likelihood	BIC (Bayesian information criterion)
2	44	-6610.772	-6218	12543
3	273	-6485.913	-6152	12492
4	159	-6421.052	-6014	12315
5	**315**	**-6714.449**	**-5808**	**12018**
6	188	-6470.779	-5777	12088

The results show that model with 5 states is the best model because it has the lowest value of BIC. Figure 4.9 below illustrates the events distribution over states and the transitions between states of 5states model.

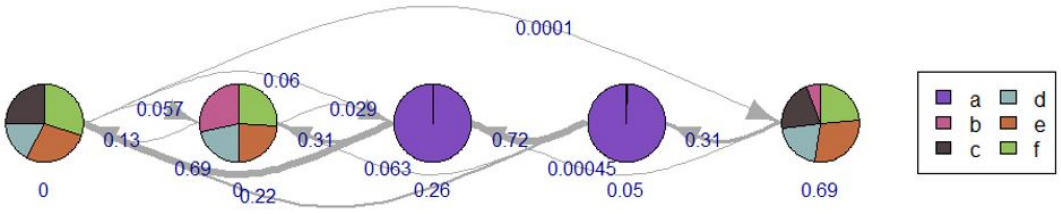


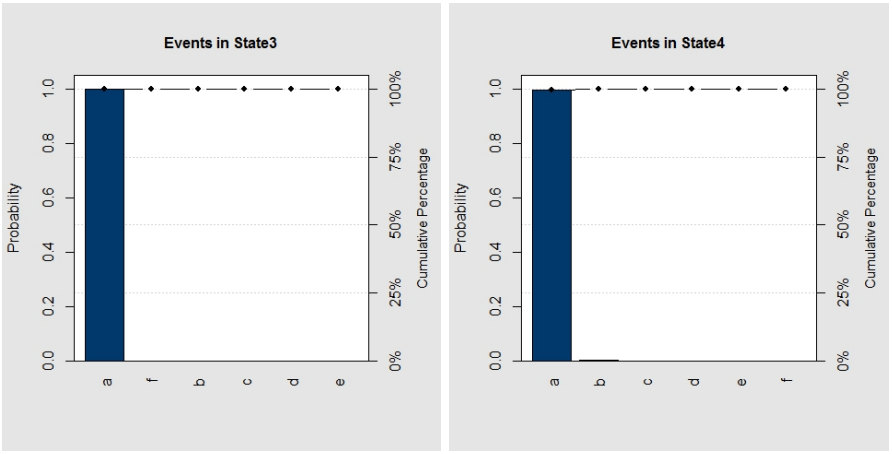
Figure 4.9: HMM of large synthetic data (experiment 3)

Results

Clearly, the presence of possible higher abstraction can be found between state 1, state 3 and state 4 due to the high probability of transitions between them. State 4 has 0.72 probability to transition to state 3 and state 3 has 0.69 probability to transition to state 1. However, there are two other issues can be found in this model :

• Similar States

In the model presented in Figure 4.9, state 3 and state 4 are similar. Although state 3 is a production state and state 4 is a simple state (contains event 'a' and very small probability of event 'b'), they are similar states. This is because more than 80% of both states are occupied by the same event 'a', see Pareto chart in Figure 4.10 below for exact probability for events in each state.



(a) Pareto chart of state 3 (b) Pareto chart of state 4

Figure 4.10: Pareto chart of simple states in large event log

• Unimportant States

The model that is shown in Figure 4.9 has 5 states however, not all presented states are significant. State importance as has been defined early, is a state that has 50% or more of cases. Therefore, states have less than 50% should not be presented in the mainstream process

model. State importance in this model is shown in the following Table 4.8.

Table 4.8: State importance of the selected model in experiment3

	state1	state2	state3	state4	state5
# of cases	367	104	389	297	346
percentage	73.4%	20.8%	77.8%	59.4%	69.2%
state important	yes	no	yes	yes	yes

(d) Complex real event log

The goal of using real event log here is to provide a larger space and observe the behaviour of BIC with high sparsity and distinct process instances (extremely variable processes) where every patient has followed a unique healthcare process. Using the MIMIC-III as explained in Chapter 3 , we have extracted an event log of a group of patients with 'colorectal cancer'. This real event log consists of the following characteristics in Table 4.9.

Table 4.9: Colorectal cancer log characteristics

Number of cases	Number of event types	Number of events	Number of variants
1197	15	270,429	1197

This log includes 15 event types including administration events such as admission, transfer and discharge in addition to other intensive care events such as bedside charting and nurse noting. For more description of the care events that are included in this healthcare process refer to Chapter 3. This event log has 1197 traces with highly variable lengths; minimum 32 and maximum 1188. Log space has 1197*1188 dimension and there are 1,151,607 null values which make the space very sparse. Sample of colorectal cancer processes is illustrated in Figure 4.11.

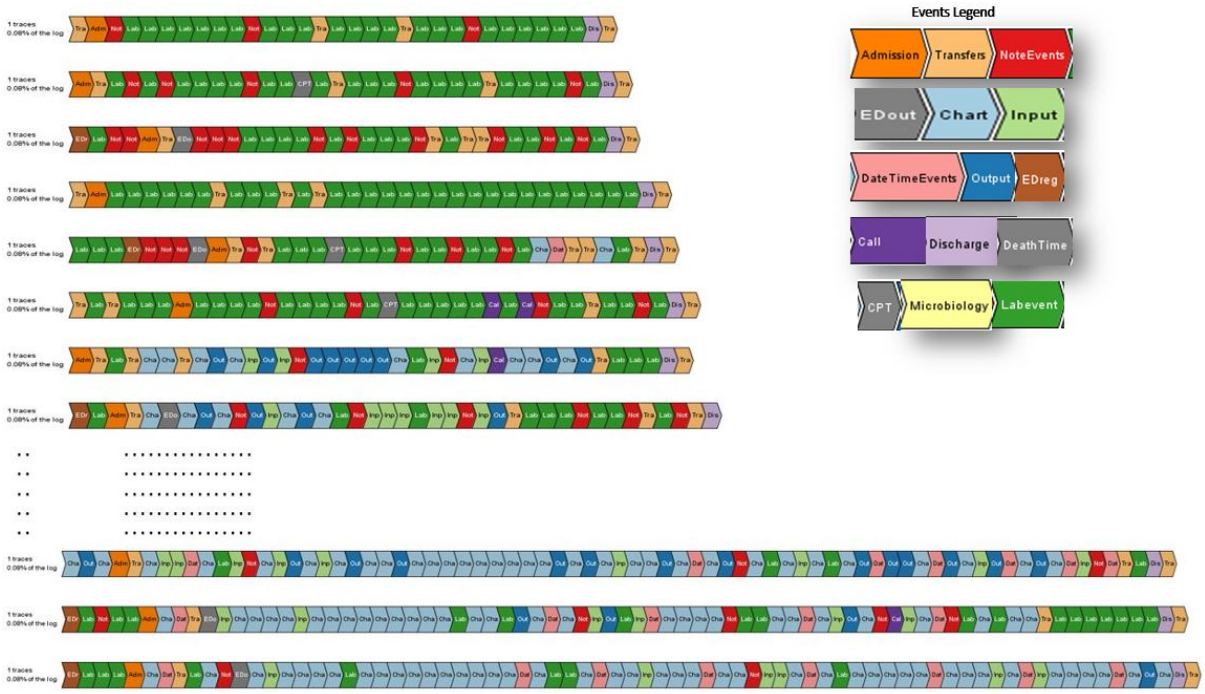


Figure 4.11: Colorectal cancer real traces (experiment 4)

The result of training HMM with different numbers of hidden states starting from 2 to 15, is shown in Table 4.10 below.

Table 4.10: Training HMMs with different number of hidden states

Number of states	Iterations =(MX)3000	Initial log	Final Log likelihood	BIC (Bayesian information criterion)
2	78	-733528.7	-439262	878912
3	182	-781375.1	-432114	864852
4	348	-790453.2	-424301	849489
5	462	-757942	-420207	841588
6	530	-723263	-415769	833025
7	1340	-717309	-415519	832864
8	849	-698582	-412234	826657
9	901	-710080.1	-411706	825987
10	1022	-720699.8	-409399	821786
11	**885**	**-732276.1**	**-401820**	**807066**
12	1142	-737459.4	-402900	809689
13	1028	-707600	-402944	810265
14	974	-738786.1	-404369	813628
15	1412	-741162.2	-403276	811979

The results show that model with 11 states is the best model because it has the lowest value of

BIC. Figure 4.12 below shows the events distribution over states and the transitions between states of the best model with 11 states.

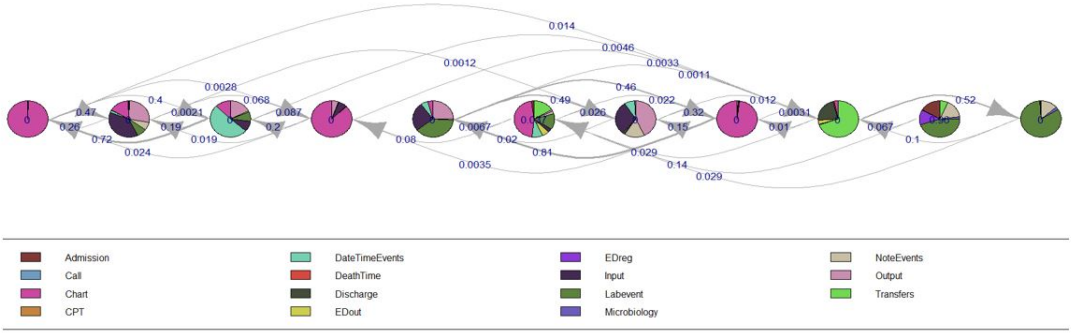


Figure 4.12: HMM of colorectal cancer data (experiment 4)

The majority of the processes started in state 10 with 0.96 probability while few processes started in state 6 with 0.037 probability.

Results

There are three issues found in this model:

• The first issue is the potential of the existence of higher level abstraction among states as shown in Figure4.13.

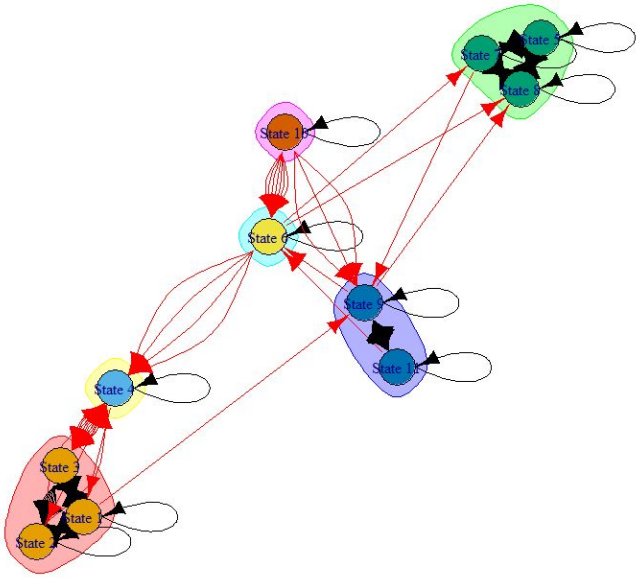


Figure 4.13: States of Figure 4.12 after using strong connected components detection

The technique of connected components detection has found three possible groups for abstrac-

tion which are, red group includes state 1, state 2 and state 3, green group includes state 5 , state 7 and state 8 and blue group that consists of state 9 and state 11. Hence, this model provides detail that hinder capturing the main pattern of care in colorectal cancer process.

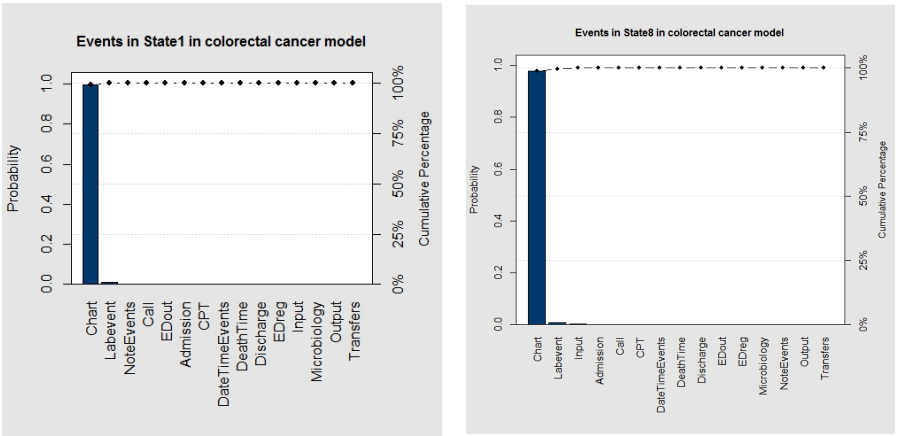
• The second issue is the presence of unimportant states. Based on Table 4.11, there are two states, state 5 and state 8, that provide undesirable details in a model that is supposed to present the mainstream process for colorectal cancer patients.

Table 4.11: State importance of the selected model in experiment 4

	state1	state2	state3	state4	state5	state6	state7	state8	state9	state10	state11
number of cases	671	669	676	692	545	1166	560	643	1197	1147	1067
percentage	56%	56%	56%	57%	45%	97%	46%	53%	100%	96%	89%
state important	yes	yes	yes	yes	no	yes	no	yes	yes	yes	yes

• The third issue is the presence of similar states. By looking at the model of Figure 4.12, we can see there is a high similarity between state 1 and state 8 and suspected similarity between state 2 and state 7.

In order to test this similarity we first check the type of states for both state 1 and state 8, and second for state 2 and state 7. Plotting Pareto chart is used here to test the states similarity.



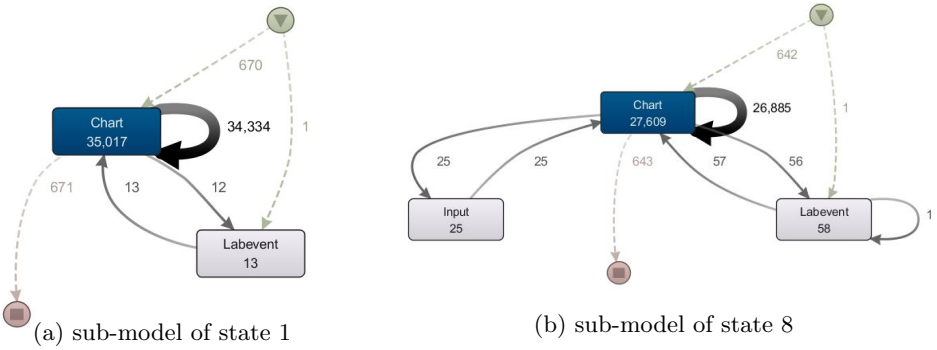
(a) Pareto chart of state 1 (b) Pareto chart of state 8

Figure 4.14: Pareto chart of simple states in complex event log

State 1 and state 8 appear to be simple and similar states as shown in 4.14. Simple states because 80% of both states is occupied by maximum 2 event types. Similar since the event types that cover 80% are the same which is Chart event.

Visualizing the sub-models of that states using Disco, a process mining tool, has proved that there is no significant difference in terms of their process, see Figure 4.15. The process of state

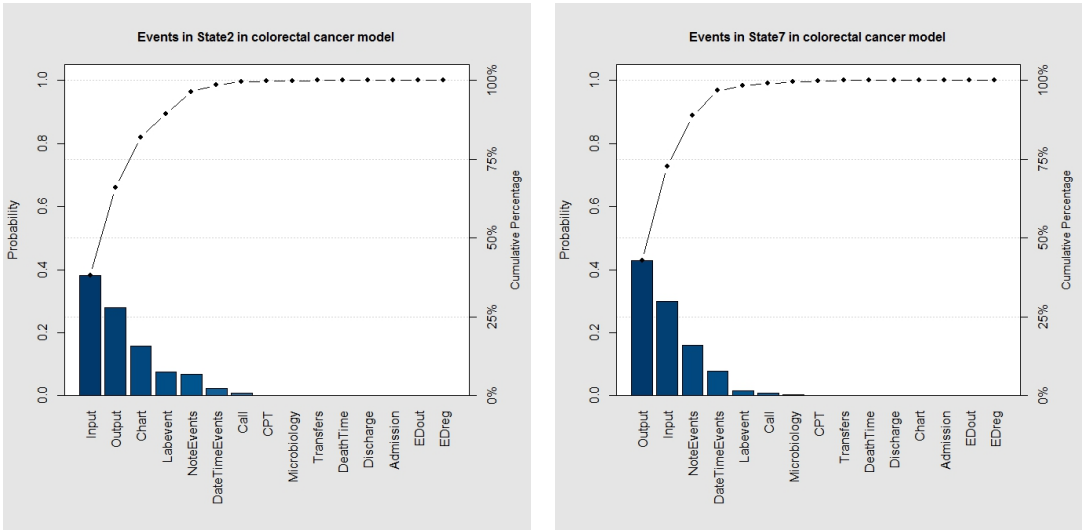
1 (a) is very similar to the process of state 8 (b) except 'Input' event, which has low frequency. It might be preferred for these two states to be combined in one single representative state that includes the three events; Chart, Labevent and Input events.



(a) sub-model of state 1 (b) sub-model of state 8

Figure 4.15: Sub-models comparison of similar states in complex event log

On the other hand, investigating the suspected similarity between state 2 and state 7 has revealed that both states of type composite as shown in Figure 4.16. However, they are not similar since the main event types that covered 80% of both states are different. In other words, the main event types of state 2 Figure 4.16 (a) are (Input, Output and Chart events) while the main event types of state 7 Figure 4.16 (b) are (Output, Input and Note events) where states differ in the third event type.



(a) Pareto chart of state 2 (b) Pareto chart of state 7

Figure 4.16: Pareto chart of composite states in complex event log

4.6.2 Discussion

The adopted method of state modelling, which includes; learning, selecting and decoding, provides a valid approach for automated abstracting of healthcare processes. However, it requires a metric for selecting the good model. BIC is a well-known metric and widely used for selecting models that are generated from learning HMMs of different states. Although BIC finds the best of candidate models with the aim to balance between fitness and complexity, it takes the models under the assumption that there must be a good model among the candidates. In other words, BIC does not examine the goodness of the states.
The empirical findings of the presented experiments confirm that models selected by BIC can suffer from some issues such as overfitting which leads to the potential of higher abstraction, the existence of multiple similar states and unimportant states.
There are several reasons that may trigger these issues. Our experiments in this chapter have focused on data size, process variability and sparsity. However, a considerable amount of literature has already proved that the EM algorithm, which is used in HMM learning, is guaranteed to optimise around the given initialization. Therefore, the deficiency of the models here is likely to be influenced by the initialization of model parameters in addition to event log size and sparsity.
However, the need for new method is essential in order to provide better abstract process model. Table 4.12, summarizes the three issues found in the previous experiments. We can conclude that BIC is sensitive to the size and sparsity of the data. It has effectively selected a good model for only the first experiment that has a small size and non sparse space.

Table 4.12: Qualitative logs description and issues found in models selected by BIC

Experiment	size	sparse	connected components	similar states	unimportant states
Experiment 1	small log	not sparse	no	no	no
Experiment 2	medium	limited sparsity	yes	no	no
Experiment 3	large	sparse	yes	yes	yes
Experiment 4	complex	high sparsity	yes	yes	yes

It should be noted that, the method of state abstraction using HMMs has a computational processing related issue. The required time for learning HMMs in our experiments is affected by the event log size and the number of hidden states. Therefore, the learning stage needs longer time as the sample size and the number of hidden states increase.

4.7 Conclusion

This chapter has provided machine learning method based on HMMs for using state abstraction modelling. Several model selection metrics are discussed for example, AIC, BIC and ICL. We have conducted some empirical experiments using toy data and colorectal cancer real log that

is extracted from MIMIC-III. The used event logs were ranging of size, process variability and sparsity. Three main issues were detected on models selected as the best models. These issues are; the potential of higher abstraction, existence of multiple similar states and unimportant states. Interestingly, BIC metric could select a good model with small size event log and this enlighten us for characterising what a good model should be.
In the next chapter, a new method for model selection is suggested. This method is motivated by finding a good model that is free, or has less suffering, of the limitations that were identified in BIC models. The proposed selection method is based on some criteria that are inspired from the empirical results of this chapter.

Chapter 5

Multi-objective Optimisation for Process Abstraction

5.1 Overview

This chapter explores four proposed criteria that may help in selecting the most desirable abstracted process model. These criteria are linearity, state compactness, cross state similarity and state importance. The rationale behind these criteria and their calculations are presented here. In addition to demonstrating the criteria properties and steps for designing a multi-objective function. This chapter focuses on the optimisation role of the designed multi-objective function, which is an important step in our methodology, see Figure 5.1. We conclude this chapter by adopting and improving the method of modelling complex process using state abstraction that is discussed previously in Chapter 4.

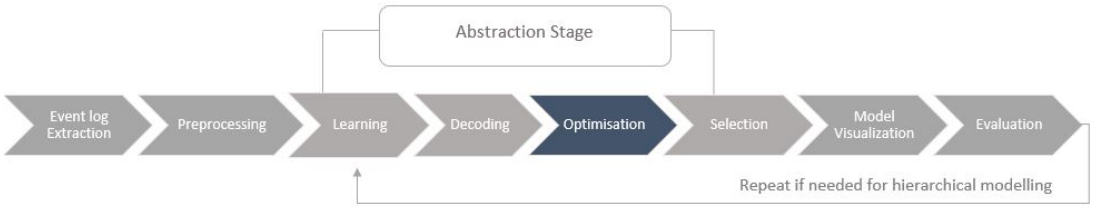


Figure 5.1: Research method and the scope of Chapter 5

5.2 Multi-objective optimisation

Multi-objective optimisation is a class of optimisation technique that is firstly discussed in the mid of 1990s as reported in [128]. It has been initially explored in different fields such as economy and engineering. Adopting this type of optimisation in machine learning recently has

received more attention because of it is inherit matching of real world optimisation problems. The main difference between this kind of optimising and other more common optimising techniques for instance, single objective function, is the representation of the objective function that is used in the optimisation process. The multi-objective function is represented in a vector form rather a scalar function. Vector representation provides better capability for optimising a problem with different or conflict objectives, which is the case of most real world problems.

In this research, the search space of a complex event log is used to populate a number of converged HMMs, then the proposed multi-objective function optimises the space of candidate HMMs models to the space of best models that are selected based on our criteria, as illustrated in Figure 5.2.

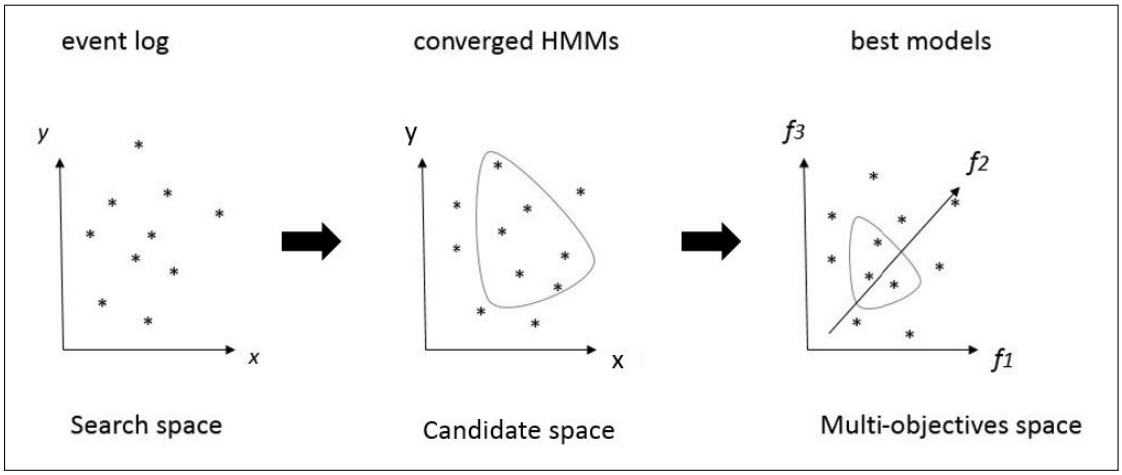


Figure 5.2: Search spaces transformation in our method

5.2.1 Pareto Optimal Solutions

Generally the aim of using multi-objective function is to optimise multiple solutions for the problem to be solved. This collection of solutions in multi-objective optimisation is known as Pareto solutions, which is named after the economist "Vilfredo Pareto" [128]. The Pareto solutions can be categorised into two types of solutions which are; optimal and feasible solutions. The optimal solutions guarantee the best trade-off between optimisation criteria among candidates space and they dominate other feasible solutions. It should be noted that, some practical search space techniques may populate candidates that are non globally optimal and hence, Pareto optimal solutions that are resulted in the optimisation step might not be optimal as well [129].

In this research and as mentioned earlier, the 'EM' algorithm is used to generate the candidates space of HMMs and this algorithm cannot guarantee finding global optimal solutions since it might be converged to locally optimal models. Thus, the results of our multi-objective opti-

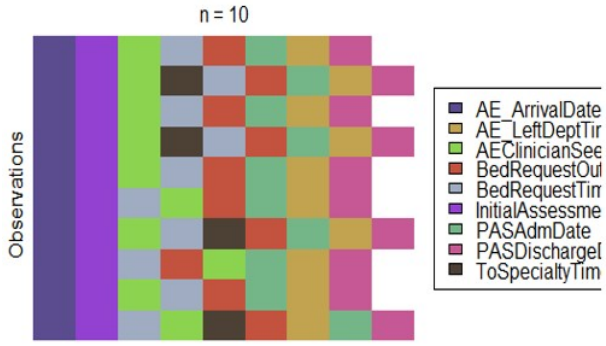
misation are not necessarily Pareto optimal solutions in regard to the search space. In this research we call objectives as criteria, which are discussed in the following section.

5.3 Proposed criteria for model selection

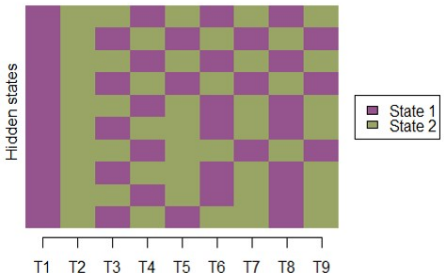
In multi-objective function each function correspond to a particular criteria that is significant in model selection. Despite the exploratory nature of our experiments in Chapter 4, the findings offered some insights into desirable model characteristics. We identify four key criteria that play significant roles in the selection of the best process model which are; linearity, state compactness, cross state similarity and state importance. The rationale of proposed these criteria and calculation methods for them are discussed below.

5.3.1 Rationale behind the proposed criteria

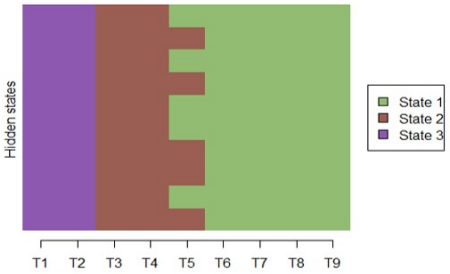
In order to avoid or mitigate issues found in models that were selected by BIC, we need to understand possible factors that may control such issues. We are motivated by adopting the same principles of the ideal model that was selected in experiment 1 in Chapter 4 due to its good characteristics for modelling processes as discussed earlier.
Investigating the structure of the hidden states of our desirable model has revealed important insights into the effect of increasing the number of states to a model. Figure 5.3 shows the projection of hidden states over sequences for models with different number of states. We used the event log that has generated the ideal model with 3 states and it is characterised in Table 4.1. It has simple tractable processes as shown in 5.3 (a) Accident and Emergency fictional observations (A&E) which help us in analysing the process qualitatively. An iterative decoding using the Viterbi algorithm is applied starting from 2 hidden states to 9 hidden states which is the total number of distinct events and presented in Figure 5.3 (b - i).



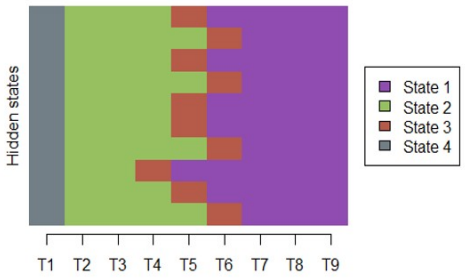
(a) Accident and Emergency room fictional processes



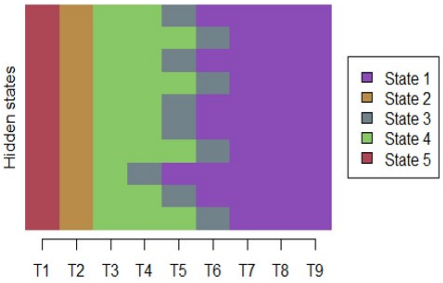
(b) 2state model



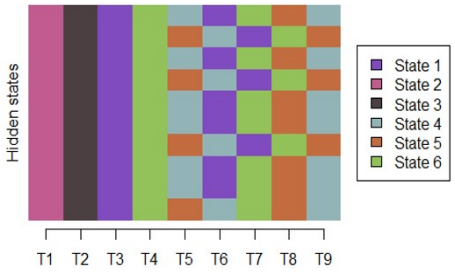
(c) 3state model



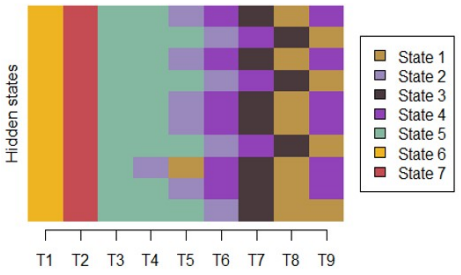
(d) 4state model



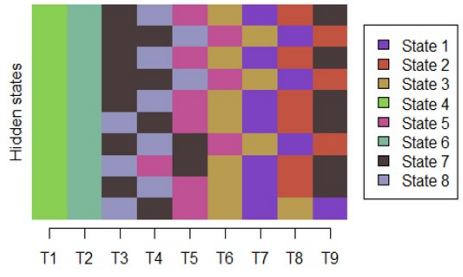
(e) 5state model



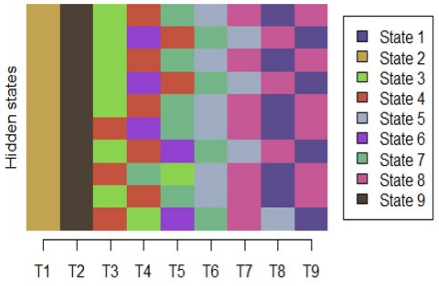
(f) 6state model



(g) 7state model



(h) 8state model



(i) 9state model

Figure 5.3: The impact of increasing the number of states to a model

In Figure 5.3 (b), two hidden states are used which results in a bad process modelling. Too few states leads to a model of high inverse transition rate (unstable model) where a process changes rapidly from one state to another. Moreover, state 1 and state 2 contain groups of highly different events, for example events occurred in state1; arrive , seen by clinician, request a bed, admission to ward and also discharge. This makes both states have a high variance which contributes to under-fitting the model.
In Figure 5.3 (c), three states are used for training. This model represents the natural flow of the process which helps in providing a good segmentation of the process into blocks of events. The model is stable which results in a linear flow of the process. There are no overlapping events between states which leads to desirable variance in all states.
In Figure 5.3 (d), the model was trained using four hidden states. Although this model seems to be stable and provides a good segmentation of the process, it has a production state which is state 1. A production state, as defined earlier, is a state has a single event type. This type of state is characterised with a very low state variance.
The phenomena of production state is expected here since we deal with a small scale event log and it might be an indication of over-fitting.
In Figure 5.3 (e - i), the model was trained by 5,6,7,8 and 9 hidden states respectively. The phenomena of production state is growing in all models which leads to highly non-preferred over-fitting models.

5.3.2 The proposed criteria

Taking the previous analysis into consideration, we propose several criteria that may work as control factors. These factors contribute effectively in characterising a good model. The suggested criteria are linearity, state compactness, cross state similarity and state importance.

(a) Linearity

Linearity can be defined as the sequential flow of the process where staying in the same state is accepted but no inverse flow is allowed. Linear HMM may include both well known structures 'left-to-right' or 'right-to-left' . More linearity means less inverse transition is preferable.
The linearity principle is natural idea with the intuitive understanding of healthcare processes. In other words, a patient is exposed to a series of healthcare steps starting from the need for healthcare, traversing intermediate investigation and ending by a healthcare outcome. This flow is highlighted by [130] in the description of a general clinical pathway guide that implies the movement from one stage to another.
We argue that although a healthcare process model looks complex at the first glance, there must be a mainstream pattern of care followed. Taking into consideration the process nature in terms of sequential direction of events and under the assumption of the hidden linearity in healthcare process, it would be a good idea to prefer a model that has captured the highest

linearity between states. This model is anticipated to have the best cut off points between blocks of care/state but is not necessarily the best fit for the data.

(b) State compactness

State compactness aims to measure the similarity of inner state processes. It is an important metric for demonstrating the validity of process clustering and quantifying state variance. There are different internal cluster validation metrics that can be used, such as entropy based metrics, however, we aim to use a metric that is more appropriate for sequence clustering validation. It is preferable to have a high compactness score which means high similarity of processes.

(c) Cross state similarity

Cross state similarity aims to measure the similarity of processes between states. This is to ensure relatively distinct states and to reduce the chance of overlapping events between states. Cross state similarity is measured by the number of common nodes and common edges. Models with high dissimilarity score between states are desirable.

(d) State importance

A state can be defined as significant if it is activated by most of the cases. In this thesis we set a threshold of state importance to be more than or equal to 50% of cases in order to capture the main process followed by at least half of patients.

5.4 Calculation of the proposed criteria

This section explains in detail the calculation for our criteria; linearity, state compactness, cross state similarity and state importance.

5.4.1 How to calculate linearity

Left-to-right topology of a HMM can be constructed by controlling the upper triangle of the transition matrix and right-to-left HMM is controlled by the lower triangle of the transition matrix. Both topologies represent a form of linearity. For instance, a transition probability matrix T of a 3states right-to-left HMM is constructed as:

$$T = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0.6 & 0.4 & 0 \\ 0 & 0.5 & 0.5 \end{pmatrix} \end{array}$$

Where the sum of probabilities of each row = 1. The linearity of this model can be calculated as:

$$L = \frac{\sum \text{probability of each row}}{\text{number of states}} \text{ , } L = \frac{3}{3} = 1$$

In a non-linear HMM model such as the case of a model with inverse transitions that has a probability transition matrix T_1

$$T_1 = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{pmatrix} 0.6 & 0.2 & 0.2 \\ 0.7 & 0.3 & 0 \\ 0 & 0.5 & 0.5 \end{pmatrix} \end{array}$$

First, we need to know what the main direction of the model is. We can deduce the direction of the model from the probability of transition matrix.
If the sum of the transitions of upper triangle part is higher than the sum of the transitions of lower triangle part then the model is mostly linear with left-to-right direction and vice versa. For instance, in T_1 the sum of the transitions of upper triangle (left-to-right direction) is = 1.8 whereas the the sum of the transitions of lower triangle (right-to-left) direction is = 2.6 therefore, model linearity is;

$$L = \frac{\sum \text{sum of the transitions of the most intensive traingular}}{\text{number of states}}$$

$$L = \frac{2.6}{3} = 0.86$$

Practical steps for linearity calculation

After learning an HMM, the model has converged and the parameters can be extracted from model's object as described in the following R code lines:
$require(seqHMM)$
$require(gdata)$
$sum(lowerTriangle(model\$transition\_probs, diag = TRUE, byrow = TRUE)) = 2.152835$
$sum(upperTriangle(model\$transition\_probs, diag = TRUE, byrow = TRUE)) = 3.095071$
% divide the largest value / number of state
3.095071/4 = 0.7737677

5.4.2 How to calculate state compactness

The similarity of processes inside a state can be measured using an optimal string alignment (OSA) score which is an extension of the edit distance score that is suggested in [131]. Optimal string alignment is designed for measuring text similarity therefore, the event log must be in the form of horizontal sequences (all events that are belong to a particular case must be in one

single row). OSA transforms a sequence to be identical to another sequence. It allows four transformation operations; inserting, deleting, substituting and transposing. For example:

- admission, blood test, charting $->$ blood test, charting: insertion of admission.
- admission, blood test, charting $->$ admission, blood test, noteevent: substitution of charting to noteevent.
- admission, blood test, charting $->$ admission, charting, blood test: transposition of blood test to charting.
- blood test, charting, discharge $->$ blood test, discharge: deletion of charting.

Each operation can be weighted hence, in healthcare application we believe that the weight of transposing should be more tolerant because transposition between events is expected and often happens in healthcare processes. The setting used for operations weight is; insertion, substitution and deletion = 1 while transposition = 0.5.
OSA calculates the pairwise alignment score between sequences. For example, an event log with N sequences will generate an $N \times N$ alignment matrix. Then the average score will be taken as the score of state compactness.

Practical steps for state compactness calculation

1. Install "*stringdist*" package in R.
2. Extract processes of each state individually.
3. Calculate the alignment score using *seq_distmatrix()* function in '*stringdist*' package.
4. Take the average score as the score of state compactness.

5.4.3 How to calculate cross state process similarity

Computing cross state similarity using the same method of OSA is not efficient as a result of the pairwise similarity calculations that are required for each sequence from one state against all the sequences in the other states. For cross state similarity we aim to measure the similarity between groups of processes/sequences between states. The work in [132] provides a review of different measurements for process comparison. We have adopted one measure which is based on the similarity of common nodes and common edges between processes.

Practical steps for cross state similarity calculation

1. Extract processes of each state individually where each state is considered as a sub-log (L).
2. Calculate similarity and this includes:

- Extract common nodes (event types).
- Calculate node (n) similarity using:

$$sim(L_1, L_2) = 2 * \frac{\text{number of } (n_1 \cap n_2)}{\text{number of } (n_1 + n_2)} \tag{5.1}$$

- Extract common edges using the method of 2-gram extraction between two consecutive nodes.
- Calculate edge (e) similarity using:

$$sim(L_1, L_2) = 2 * \frac{\text{number of } (e_1 \cap e_2)}{\text{number of } (e_1 + e_2)} \tag{5.2}$$

- Calculate total similarity score as:

$$\text{Total sim}(L_1, L_2) = \frac{\text{node similarity score + edge simialirty score}}{2} \tag{5.3}$$

3. Take the average score of the symmetric matrix that holds cross state similarity of all sub logs as a score of that model.

5.4.4 How to calculate state importance

State importance is a simple metric and is calculated as:

$$\text{State importance} = \frac{\text{number of cases in a state} * 100}{\text{number of all cases}} \geq threshold \tag{5.4}$$

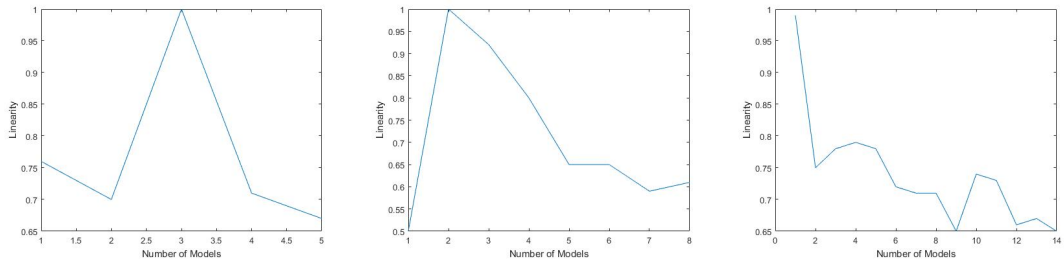
This metric returns how many non-significant states a model has.

5.5 Criteria Properties

The investigation of the property of the proposed criteria aiming at understanding how these criteria can be used to cope with the issues found in HMMs. It should be noted that, state importance is a constraint criteria, since it has a predefined threshold, and will play a penalty role in our multi-objective function. Therefore, we explore here the property of unconstrained criteria which are; linearity, state compactness and cross state similarity.
The figures used here are related to event logs that were used in experiments 1,2 and 4 in Chapter 4. The event log of experiment 3 is discarded to avoid redundancy since it has trained with same number of hidden states in experiment 2.
Models in X axis in all figures started from the smallest number of hidden states to the largest for, instance, model number 1 is trained with 2 hidden states and model number 2 is trained with 3 hidden states and so on.

5.5.1 Linearity

High linearity models are likely to be models with a small number of states. The more states the lower the linearity will be. Figure 5.4 shows that, high linear process is found in model number 3 in (a), model number 2 in (b) and model number 1 in (c) where these models have 4, 3 and 2 hidden states respectively.

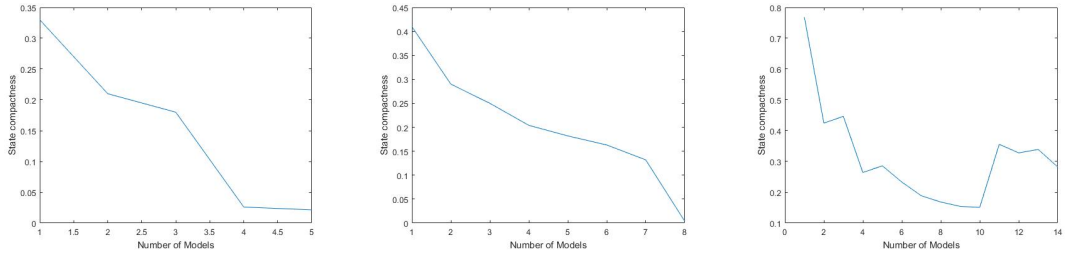


(a) Synthetic process (b) Emergency room process (c) Colorectal cancer process

Figure 5.4: Linearity criteria property

5.5.2 State compactness

Models with highest compactness are likely to be models with a large number of states. The higher the number of states, the better compactness where the distance between processes inside a state approaches 0. The compactness keep improves as we go positively in X axis as shown in Figure 5.5. The best compactness is model number 5 , 8 and 10 for (a) synthetic process, (b) emergency process and (c) colorectal cancer process respectively.



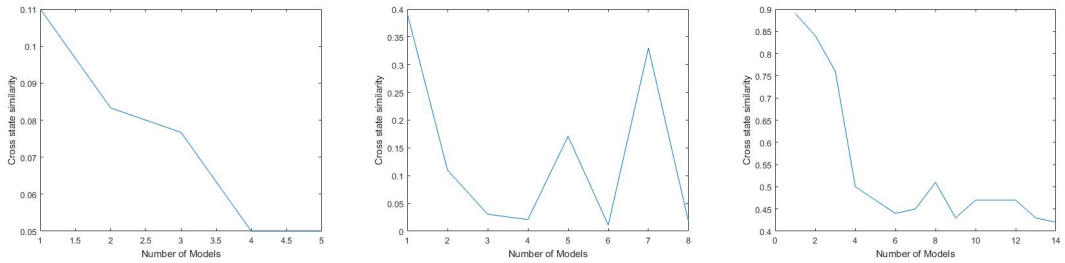
(a) Synthetic process (b) Emergency room process (c) Colorectal cancer process

Figure 5.5: State compactness criteria property

5.5.3 Cross state similarity

Cross state similarity has also a fluctuating pattern where it starts with high cross similarity score, then drops with low score after that some high scores might be found again. Generally, models with desirable low cross state similarity are likely to be captured before approaching

best compactness points.
It can be clearly seen in Figure 5.6 that several minimum points of cross state similarity are reached before approaching the point of best compactness. For example, in (a) synthetic process, the cross state similarity has dropped in model number 4 before model number 5, which has best compactness.
Also, in (b) emergency process, low similarity between states is detected in model number 3, 4 and 6 before model number 8, that has best compactness. Likewise in (c) colorectal cancer process has several models with a very low cross state similarity score such as model number 6, 7 and 9 before best compactness model, model number 10.



(a) Synthetic process (b) Emergency room process (c) Colorectal cancer process

Figure 5.6: Cross state similarity criteria property

5.5.4 Discussion

Exploring the criteria properties helps in understanding the potential impact of the proposed criteria on HMMs issues. For more clarification, the empirical results in Chapter 4 have shown that the higher number of states in a model, the more likely of detecting strong connected components. Also, high linearity model is likely to be a model with few number of states. Thus, linearity criteria may act as an effective factor that helps in choosing the model with less number of connected components.
On the other hand, results of previous experiments reported that similar states issue is not only restricted to models with high number of states but can be found in models of few number of states likewise, see Figure 5.6. This might be because of the bad initialization which is a consequence of the lack of knowledge about the data. Adopting a metric that prefers a model with high dissimilarity between states might help in coping with this similar states issue. Hence, the criteria of cross state similarity could play a role for addressing such issue.
An attention should be paid for preferring a model with high dissimilarity between states since it may result in choosing a model with a large number of states where each state holds one single event type. Therefore, the risk of preferring a model with high dissimilarity between states can be mitigated by favouring a model with a reasonable states variance along with dissimilar states. State compactness here may play a role in preferring a model of moderate

states variance.
Moving to the third issue that is related to the presence of unimportant states where this problem seems to be resulted form insufficient data along with bad initialization. Some concerns may rise regarding the EM algorithm, that is used for learning HMM models, which is not smart enough to decide if a state is significant or not. Therefore, we intend to use a penalty term where a process miner can set a threshold for state importance that should be presented in the model.

5.6 Designing the multi-objective optimisation function

In order to quantify the optimal model we need to design an objective function that combines and represents our criteria. We are interested in finding a model with high linearity (l), moderate state compactness (sc), low cross state similarity (css) and includes only important states (si). The idea of weighted parameters is widely used in multi-objective function optimisation [133] which is in the form of

$$max \sum_{i=1}^{n} w_i * f_i(x), \text{ where } w \text{ is the weight and } f(x) \text{ is the desired function}$$

Adopting this principle can help us in designing our multi-objective function especially when the goal is not for finding a global optimum solution.
We have developed the following multi-objective functions to find the most desirable model for process abstraction. Two types for optimisation are developed which are soft optimisation and strict optimisation.

1- Soft optimisation uses multi-objective function that is presented in the Equation(5.5) for optimising candidate models space. This function represents only unconstrained criteria; linearity, state compactness and cross state similarity. It has a tolerance toward unimportant states. The purpose of this flexibility is to pick a model that represents the mainstream process regardless how many cases related in each state, hence model selected using this type of optimisation may hold a level of detail about the process.

$$f(1) = max\{ \underbrace{2\ (l_i)}_{\text{Term2}} - \underbrace{\frac{1}{2}(sc_i) - css_i}_{\text{Term1}} : i = 1..n\} \tag{5.5}$$

2- Strict optimisation uses multi-objective function that is showed in the Equation(5.6) for optimising candidates model space. This function represents both constrained and unconstrained criteria; linearity, state compactness, cross state similarity and state importance. As its name means, this type of optimisation takes state coverage into consideration and tends to penalize model with unimportant states. This function is developed to pick the best model that

represents the mainstream process without detail about the process.

$$f(2) = max\{\underbrace{1 - si_i}_{\text{Term3}}\underbrace{[2\ (l_i)}_{\text{Term2}} - \underbrace{\frac{1}{2}(sc_i) - css_i]}_{\text{Term1}} : i = 1..n\} \tag{5.6}$$

state importance here is normalized using the following simple normalisation method:

$$\text{normalized state importance } (si) = \frac{\text{original value - minimum value}}{\text{maximum value - minimum value}}$$

The proposed multi-objective functions have main terms:
Term1 consists of two functions which are state compactness and cross state similarity. State compactness decreases as we get high similarity inside states which leads to low state variance. knowing that state compactness keeps decreasing has derived us to reduce the weight of state compactness to half. This helps in penalising a model of having very low state variance. Cross state similarity decreases as we get less overlapping processes between states which is preferable.
Term2 is linearity which increases as we get a more linear model and this rewards a model for providing better segmentation/partitioning. It is multiplied by 2 to ensure equal forces in the space. For more clarification, choosing this weight for linearity is a result of several empirical investigations as will be explained in the next section.
Term3 further penalises a model that has unimportant states.
This final version of the strict and soft multi-objective optimisation is a result of several attempts to understand each function individually and how all criteria may affect on each other.

5.6.1 Steps for designing the proposed multi-objective function

Before demonstrating our steps for designing the multi-objective function, it is important to take into consideration the property of each criteria and the relations between them. This has helped in criteria weightings and explaining how the criteria could contribute in a trade-off relation.

Criteria trade-off and weightings

Finding a solution in search space requires finding a balance between conflict criteria. This is because a model with high linearity is unlikely to have high state compactness or high dissimilarity between states. The weighting here focuses on unconstrained criteria since state importance will be treated as a penalty term after optimising the space using soft optimisation.

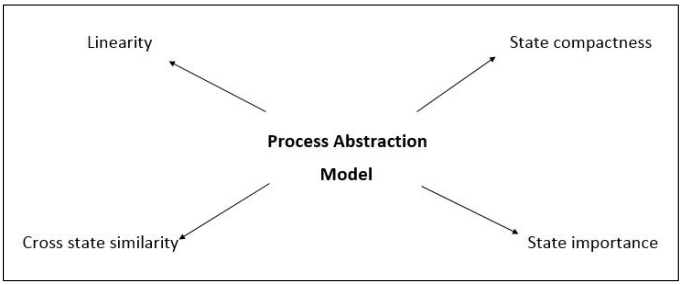


Figure 5.7: Process abstraction model and different forces

We have applied simple heuristic for criteria weighting tuning which has done using our synthetic logs and real data of colorectal cancer log. The following presented figures are generated from the colorectal cancer event log to avoid redundancy, however, other synthetic event logs that were presented in Chapter 4 have shown the same behaviour of criteria tuning. The calculation of the criteria using colorectal cancer data is reported in Table 5.1.

Table 5.1: Criteria calculation of colorectal cancer event log.

Model	1	2	3	4	5	6	7	8	9	10	11	12	13	14
States	2s	3s	4s	5s	6s	7s	8s	9s	10s	11s	12s	13s	14s	15s
Linearity	0.99	0.75	0.78	0.79	0.78	0.72	0.71	0.71	0.65	0.74	0.73	0.66	0.67	0.65
compactness	0.76	0.42	0.44	0.26	0.28	0.23	0.18	0.16	0.15	0.15	0.35	0.32	0.33	0.28
cross sim.	0.89	0.84	0.76	0.50	0.47	0.44	0.45	0.51	0.43	0.47	0.47	0.47	0.43	0.42

During weights tuning process, we aimed to investigate the behaviour of state compactness and cross state similarity with linearity. Basically, combining state compactness and linearity in one function shows the dominance of the compactness over linearity where the model of the maximum value is the model of the best compactness score as shown in Figure 5.8.

$$f_0 = max\{linearity_i - state\ compactness_i : i = 1..n\} \tag{5.7}$$

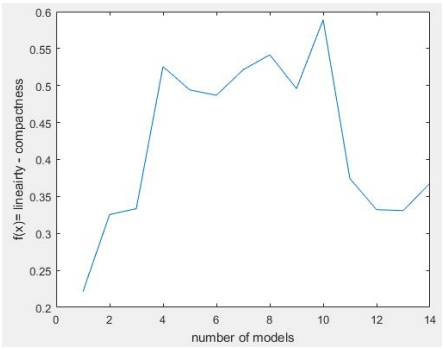


Figure 5.8: State compactness and linearity in colorectal cancer log

Hence, we have minimised the weight of state compactness to the half as below;

$$f_1 = max\{linearity_i - \frac{state\ compactness_i}{2} : i = 1..n\} \tag{5.8}$$

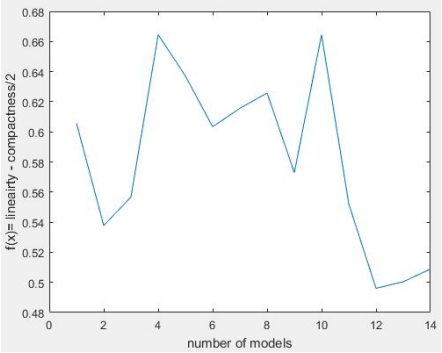


Figure 5.9: State compactness and linearity after weighing

It can be clearly seen in Figure 5.9, that linearity has started to affect on function behaviour where model number 4, which has high linearity, seems to be equivalent to model with high compactness.
On the other hand, the relation between cross state similarity and linearity without weights has provided a balance between their forces where no one of these criteria has a dominance effect on the function behaviour.

$$f_2 = max\{linearity_i - cross\ states\ similarity_i : i = 1..n\} \tag{5.9}$$

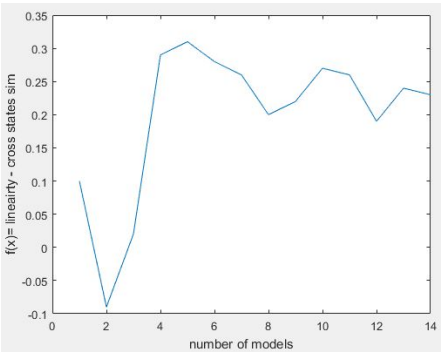


Figure 5.10: Cross state similarity and linearity criteria

The best model as presented in Figure 5.10, is model number 5, which is free of controlling by linearity or cross state similarity thus, there is no need to tuning their weights.
Investigating the behaviour of our multi-objective function after involving the three criteria has shown the dominance of the compactness again. This result seems to be plausible since the equation includes two forces which are compactness and cross state similarity and both pull the selection of the best model towards the right, high number of state as presented in Figure 5.11

$$f_3 = max\{linearity_i - \frac{1}{2} * (state\ compactness_i) - cross\ state\ similarity_i : i = 1..n\} \tag{5.10}$$

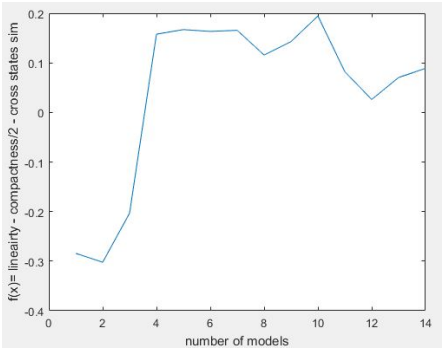


Figure 5.11: The three criteria in one function

Therefore, increasing the weight of linearity is suggested to provide a balance point in the space. Hence, the candidate space now has two forces (compactness and cross state similarity) that encourage the selection to go to high number of states and one force (linearity) with double weight to prefer the model of less number of states, see the following equation.

$$f_4 = max\{2*linearity_i - \frac{1}{2}*(state\ compactness_i) - cross\ states\ similarity_i : i = 1..n\} \quad (5.11)$$

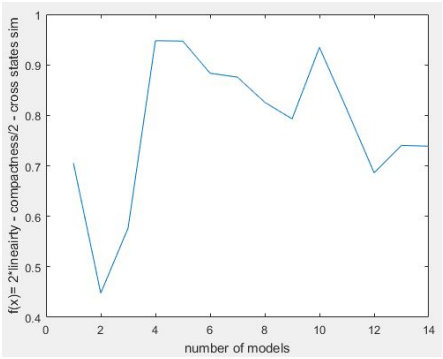


Figure 5.12: Fitted weights for all criteria

Modifying the weight of linearity as suggested above has helped our multi-objective function to select a balanced model that is non-dominant by one of the criteria and represents resistance to all forces as illustrated in Figure 5.12.
This equation is the proposed multi-objective function which guarantees non of the criteria is dominant and assumes a balance of different forces is found.

5.7 Putting it all together: The proposed improved method for state abstraction modelling

In order to develop a robust method for state abstraction modelling, the method that is discussed in Chapter 4 (Section 4.3) is adopted and improved. The improvements include:

- Involving our multi-objective optimisation as a main step in the method before the selection stage.
- Replacing the traditional metrics for selecting the best models by the results that are suggested using our optimisation.
- Applying a hierarchical modelling if the best selected model has a special type of states that is defined as a complex state. The reason for that is to provide better modelling since the complex state usually represents complex processes and to compensate model fitting.

The improved method of state based abstraction for healthcare process modelling has eight stages as follows:

1. **Extraction**: The required event logs are extracted using the methods that are explained in Chapter 3, which are applicable to most electronic health records data.

2. **Preprocessing**: The event logs need to be prepared and cleaned. The prepared logs should be converted to horizontal sequences format to be suitable input for the next stage.

3. **Learning**: This stage is the start of the abstraction stage. The algorithm that is used for learning is the Expectation-Maximisation (EM).

4. **Decoding**: Decoding is performed by running the Viterbi algorithm over sequences. The number of hidden state for each relevant event is extracted to enrich the event log with state numbers as an added attribute.

5. **Optimisation**: Optimising the space using our proposed multi-objective function and applying both types of optimisations in order to provide some sort of flexibility. We suggest that, the strict optimisation is applied first in order to get the mainstream process model and improve the understandability of the process, then the soft optimisation is performed which may provide further detail about the process.

6. **Selection**: Selecting the best model based on the maximum score for both optimisation types and at this step the abstraction stage ends.

7. **Model Visualization**: Visualizing the selected model using our new state based abstraction model. Process model visualization using the available package in R,'Seqhmm', is not helpful because of the lack of the understandability in terms of the start and end of the process. In addition to the non desirable probability that is shown on the edges where the probability of incoming edges are not necessarily equal to the outgoing edges. This may cause a confusion in tracking the flow of the process. We think this probability may not help in increasing the understandability of the process. Also, the layout of the model is not suitable for modelling a process as it is plotted in one line where the

transitions between states become hard to follow. Therefore, we have implemented a new visualization function as an enhancement of the old function in 'Seqhmm' to cope with these limitations. The new function is used to visualize all process models of our case studies that are illustrated in Chapter 6 and Chapter 7.

8. **Evaluation**: Models are evaluated using three different aspects which are models selection validation, process model quality metrics and lastly domain expert evaluation.

As a further step is applying a hierarchical modelling if needs. These eight steps are repeated if the best selected model has a special type of states that is defined as a complex state. Checking the type of states is needed to provide a better abstract process model through a hierarchical modelling for complex states.

5.7.1 Multi-objective optimisation algorithm

Steps of the proposed multi-objective function for selecting the best abstracted model are presented in the following pseudocode, see **Algorithm 1** below.
This algorithm explains the sequence of steps should be taken to optimise HMMs candidate space. Line(8-11) show the first step for optimisation which is the generation of HMMs candidates. The generated models start by at least two hidden states and increase iteratively to the upper bound which is the number of distinct events in event log.
Line (12-27) show the primary contributions of this thesis where the proposed criteria are calculated for every model. Then two types of optimisations are computed and a vector that includes the optimisation scores and the best model's index for each optimisation type is returned. In Line (24-26), checking if the best model has a complex state, a state is complex when 80% of that state is occupied by more than two event types and has a highly variable processes. If a complex state is found, then start hierarchical modelling and do the same steps again.

Algorithm 1

1: Given: N; the number of distinct events
2: Lg; the event log
3: i; index for model
4: j; index for decoder
5: s; state
6: w; a threshold for state importance
7: θ ; HMM initial random parameters
8: **for** $state\_number : 2, N$ **do** ▷ Generating HMM candidates
9: $M(i) \leftarrow model\_learning(\theta, Lg)$
10: $D(j) \leftarrow viterbi\_decoding(M(i), Lg)$
11: **end for**
12: **procedure** OPTIMISATION(M, D) ▷ Optimising candidates space
13: **for** $i : 1, M$ **do**
14: $l(i) \leftarrow linearity(M(i))$ ▷ see criteria calculation in Section 5.4
15: **end for**
16: **for** $j : 1, D$ **do**
17: $comp(j) \leftarrow state\_compactness(D(j_s), Lg)$
18: $cross(j) \leftarrow cross\_similarity(D(j_s), Lg)$
19: $importance(j) \leftarrow state\_coverage(D(j_s), Lg, w)$
20: **end for**
21: $soft[score, indx] \leftarrow max\_soft\_optimized(l, comp, cross)$ ▷ Equation 5.5
22: $strict[score, indx] \leftarrow max\_strict\_optimized(l, comp, cross, importance)$ ▷ Equation 5.6
23: **return** $soft,\ strict$
24: **if** (s is complex) **then** ▷ for hierarchical modelling
25: extract events of this state and go to line 8
26: **end if**
27: **end procedure**

5.8 Conclusion

This chapter discussed four suggested criteria for modelling state abstracted healthcare model. The criteria are linearity, state compactness, cross state similarity and state importance. The rationale of selecting these criteria is explained and the calculation is discussed elaborately. Exploring criteria properties and the relations between the criteria have helped in tuning appropriate weights, based on simple heuristic, and designing our multi-objective function. Two types for optimisations are suggested which are soft and strict optimisations. Soft optimisation used unconstrained criteria while strict optimisation used unconstrained criteria in addition to constrained function which is state importance. Lastly, a robust method is developed to include the suggested optimisation in the process of healthcare state abstraction modelling. In this chapter, criteria exploring and weights tuning have done using synthetic data and real event log that is extracted from MIMIC-III database. In the next chapter, the developed method will be tested on a different source for healthcare process namely PPM database in order to evaluate the results with a domain expert as the last stage of our method.

Chapter 6

Case Study 1: Chemotherapy cycles of breast cancer patients

6.1 Overview

The aim of this chapter is to explain how to test our improved abstraction method on discovering the mainstream model using real world data for healthcare processes. A case study of breast cancer patients is extracted from the PPM electronic healthcare system. The experiment in this chapter has followed the steps of our method that is discussed in Chapter 5 and illustrated in Figure 6.1. Both types of optimisations are applied. The strict optimisation model is used for describing the main healthcare processes while the soft optimisation models is used for process outcomes analysis since this model provides more detail about the process. Process model is visualized using our new enhanced visualization method and evaluated through different aspects.

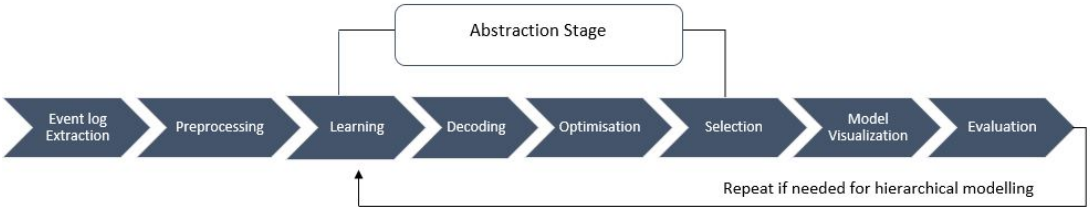


Figure 6.1: Research method and the scope of Chapter 6 and Chapter 7

6.2 Breast Cancer Healthcare Process in the UK

According to World Health Organisation (WHO, 2019), breast cancer is the second leading cause to death in the world. However, "Although the incidence of breast cancer is increasing

in the UK, mortality is decreasing thanks to significant achievements in the organisation and delivery of breast cancer care, with multidisciplinary working at its centre." [134, pg.1]
Based on [134], breast cancer patients generally start the healthcare journey following a referral. Then a patient case is discussed with a multidisciplinary team of specialists to agree on treatment type and strategy. There are several interventions that can be taken. Cancer is usually controlled first by surgery if it is suitable for patient condition and this is considered as a primary treatment. Then, the adjuvant treatment type is planned which aims to prevent cancer reoccurring such as radiotherapy, chemotherapy or hormone therapy.
If chemotherapy is described, based on [135], the chemotherapy treatment is not only one session. The oncologist decides the treatment course and how many cycles a patient needs in addition to identifying a treatment plan, which is known as the chemotherapy regimen. Each regimen usually includes one or more chemotherapy drugs with determined doses. The oncologist selects a chemotherapy regimen based on different factors such as the type and part of body where the cancer is located and if it has spread or not. If a primary treatment such as surgery was not applicable due to the size of tumour or other reasons, a neoadjuvant treatment type is planned, which is given before primary treatment to help shrink the tumour and, consequently, improve the chance of managing and removing it. There is another type of treatment called palliative treatment which is used to improve the quality of life for patients who cannot be completely cured from cancer due to the spread of the tumour.

6.3 Patient Pathway Manager (PPM)

Patient Pathway Manager [24] is an Electronic Health Record (EHR) used to store clinical and coded data of all patients who have cancer at the Leeds Cancer Centre, which is one of the largest cancer centres in the UK. It was developed by Leeds Teaching Hospitals Trust (LTHT) in 2003 and extended to include different kinds of healthcare data that is integrated from different sources for example, Patients Administration System (PAS) to access inpatients and outpatients data, Chemocare system for cancer treatment data and other healthcare systems to include radiotherapy and laboratory results.
The PPM data includes diagnosis of patients from 1921 to now (2019). It has clinical records of more than 2.39 million patients [16] which contain a considerable amount of routine healthcare data that might be used for mining patients pathways.
In this chapter we use an extract data of the PPM EHR that represents all care events for several visits of patients where these care events can be tracked using patients IDs as shown in Figure 6.2. The extract of the data was guided by experts interests on some tables of the PPM system.

Figure 6.2: Healthcare process perspective in PPM

6.4 Data acquisition from PPM

An event log is the core item of process mining research and the first step of applying process mining is accessing the system and extracting required event logs. The PPM system is not a process-aware system which means event logs do not automatically exist in the system but a process miner needs to extract them through several steps as mentioned early in Chapter 3. The steps for acquiring event logs from PPM are explained below:

6.4.1 Creating an event log from the PPM

Creating an event log from the PPM can be done through a number of steps:

1. Firstly, getting access to the data. Access to the data is given after arrangements with PPM data providers. This research forms part of the work funded in the SBRI- 1 grant, (project number 1203SBRSB2DANRSBRI Application. doc 20504-149147). The work was hosted by Leeds Teaching Hospitals NHS Trust (LTHT). This work was sanctioned according to local LTHT research and development policy. Data extraction was carried out under strict information governance procedures, including anonymisation of patient-level data. The extract data of the PPM database has been made accessible to us in 13 individual files in the format of comma separated files. All files are stored on an encrypted pinned secure hard drive that is accessible only for authorised researchers. A form contains the Standard Operating Procedure (SOP) for using this pinned secure drive is written and signed to ensure the appropriate use of the data. Also, patients data were anonymized to ensure patients confidentiality.

2. Secondly, creating the local database. In order to extract event logs from PPM files, a local version of the extract data of PPM is created and located on our secure drive. The 13 provided files are imported into the created database.

6.4.2 Extraction an event log for the required cohort of patients

PPM data reference model is constructed in this research using the Entity Relationship Diagram in PostgreSQL Database editor (ERD). This Diagram helps to extract the event logs and identify possible care events in the EHR. Figure 6.3 shows all the tables that have temporal fields which are needed for process mining. A process can be tracked using a patient's ID. The extraction criteria for each case study and the description of care events are discussed in this chapter for case study 1 and Chapter 7 for case study 2 and 3.

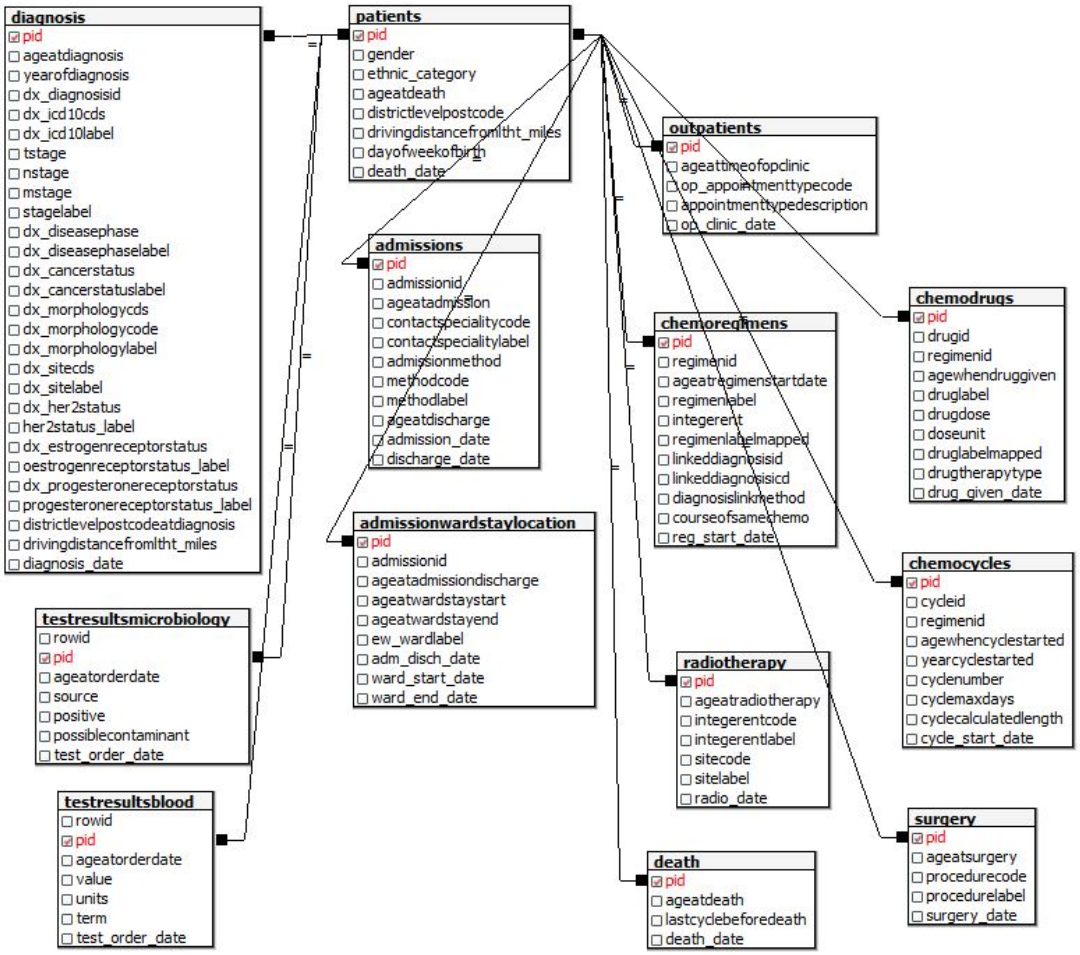


Figure 6.3: PPM data reference model generated in this research

It should be noted that, the PPM extract data that we obtained might be affected by the provenance chain of getting the data where it was extracted for a previous research project, that is discussed in [16], with different aims. Initial exploring for the tables has shown that the table of 'chemodrugs', which contains data about the drugs labels, drugs doses and other drugs related information, has an ambiguous date of when a drug is given. In other words, the

event of taken a drug is recorded with a date that was before a chemotherapy cycle starting. A discussion with an expert who works on the PPM data has suggested that this date is confusing where the chemodrug should be given on the same day of getting chemotherapy and the recorded date might be resulted from data quality issues. Therefore, events generated from 'chemodrugs' table are excluded in this research to avoid misleading results.

Time issues in the PPM extract data:

Although the extract of the PPM data has the three components of mining patients process which are patient id, event name and event date, the temporal fields were recorded in a number of days (integer format) as a step of date manipulation to protect patients confidentiality. This number corresponds to the age of a patient when an event happened. Therefore, a method for a valid timestamped format reconstruction is required.

(a) Reconstructing 'timestamped' format from age determined event

In order to construct 'timestamped' format from the number of days of an event occurring, we need to set a default day as a start point for that. The date **'2020-01-01 12:00:00'** is chosen as an artificial default day and then the number of days is subtracted from the default date.
This method has converted the age in number of days into a valid timestamped format that can be parsed by process mining tools.
Despite the successful reconstruction of the time format, a further issue relating to event order is raised as discussed below in (b).

(b) Reconstructing events order

One of the major event logs quality issues as discussed by [95] is the level of temporal resolution. Some events are recorded by time resolution which can be down to the second or time resolution down to the day only. We found that all events are recorded on 'day only' time resolution. Consequently, knowing the order in which events have happened on the same day is not applicable.
Hence, the order of same-day events is given based on an expected sequence of those events which have been validated by a domain expert.
Examples of same-day events are:
1- 'Admission' and 'Ward stay', as a patient should be admitted first and then stay in a ward.
2- 'Regimen start' and 'Chemotherapy session', Regimen of treatment should be discussed and approved then chemotherapy treatments are given.

6.5 Extraction criteria of case study 1

The first case study focuses on extracting a relatively complex healthcare process of breast cancer patients who have chemotherapy treatment. The extraction here follows the breast cancer extraction criteria that has done in a related work in [16] where we believe this extraction provides a relatively complex processes since it has a limited scope of patients selection and event selection as will be discussed below. We aim to extract a specific cohort of patients who have been diagnosed with breast cancer (ICD-10 code 'C50') between 2004 and 2013 and received epirubicin and cyclophosphamide (EC90) chemotherapy as adjuvant treatment.

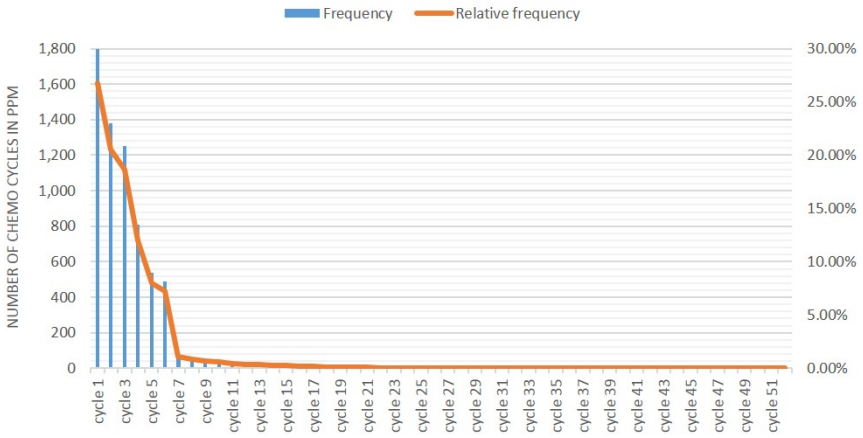


Figure 6.4: Chemotherapy cycles frequency

Based on [135] and [16], a chemotherapy course is expected to last for 6 cycles and this may increase or decrease based on patient condition. We found that there are a large number of chemotherapy cycles which reached 52 cycles for few patients as shown in Figure 6.4. This number of cycles is validated later with the domain expert as will be discussed in the evaluation section (6.9.3). It can be clearly seen that the number of cycles drops dramatically after cycle 6. Thus, we have focused on mining the process of chemotherapy patients using the 6 chemotherapy cycles only. The selection of care events concerns about the main events for healthcare process such as admission, discharge, chemotherapy cycles, blood test, death and acute event such as neutropenia sepsis. Neutropenia sepsis is a potentially fatal event and can be identified by a neutrophil count that is less than $1.5 * 10^9/L$ according to [16].

Table 6.1: Log characteristics of single type of treatment for breast cancer

total cases	distinct event	total events	variants	variation (%)	nulls	case length		
						min	avg	max
739	13	30996	640	86	208440	8	40	324

The inclusion criteria of this case study as shown in Table 6.1 has extracted 739 patients and

13 distinct events. The length of case is based on the number of events where the longest process has 324 events whereas the shortest process has 8 events only. The extracted event log is sparse as indicated by the number of nulls that are generated from the variable length process instances. The frequency of events is presented in Figure 6.5. 739 cases demonstrate 640 variants of processes which results in 86.6% of process variation.

Activity	▲ Frequency
Discharge	11,794
admission - Elective	10,562
cycle 1	1,798
cycle 2	1,379
cycle 3	1,252
admission - Emergency	1,228
blood test	1,017
cycle 4	809
cycle 5	539
cycle 6	489
death	79
Neutropenia sepsis	46
admission - Other	4

Figure 6.5: Screenshot of case study 1 events and frequency

6.6 Models learning and decoding

Applying our method to this event log has resulted in populating 12 possible models with 13 hidden states, which is the upper bound based on the number of distinct events, as shown in Table 6.2.

Table 6.2: Learning HMMs with different number of hidden states in case study 1

Model number	Number of states	Iterations =(MX)3000	Initial log	Final Log likelihood	BIC (Bayesian information criterion)
1	2	151	-81625.06	-49059.08	98397.39
2	3	236	-75183.38	-34022.01	68499.05
3	4	198	-76121.36	-30501.01	61653.55
4	5	521	-87077.66	-30441.22	61751.13
5	6	762	-77192.73	-29246.65	59599.85
6	7	440	-74412.32	-28418.14	58201.36
7	8	1345	-75817.65	-28677.09	58998.50
8	9	584	-81967.37	-27933.81	57811.83
9	10	608	-81395.04	-27331.45	56927.72
10	11	387	-80045.33	-28177.09	58960.26
11	**12**	**1217**	**-79262.91**	**-26930.91**	**56829.86**
12	13	678	-75377.57	-26809.92	56970.52

The standard metric of selecting best model of HMM, BIC, has selected a model with 12 states(highlighted) as the best model. However, we believe less number of states can represent the main pattern of care since BIC tends to select an over-fitting model. Following the steps of the proposed method, all models are decoded using the Viterbi algorithm as a pre-step for models candidate optimisation.

6.7 Optimisation

We have run both versions of our multi-objective function to optimise models candidates space:

1- Strict optimisation for models' candidate space

Selecting the best model with considering state importance is achievable using the strict optimisation of our proposed multi-objective function (Equation 5.6). In this case, we would like to see the process model of most patients where each state should have no less than 50% of cases. Criteria calculation using the strict optimisation is reported in Table 6.3.

Table 6.3: Criteria calculation in case study 1 (strict optimisation)

Model	1	2	3	4	5	6	7	8	9	10	11	12
States	2s	3s	**4s**	5s	6s	7s	8s	9s	10s	11s	12s	13s
Linearity	0.99	0.66	**0.73**	0.62	0.66	0.69	0.55	0.74	0.63	0.60	0.57	0.60
Compactness	0.172	0.034	**0.022**	0.017	0.014	0.014	0.009	0.009	0.008	0.013	0.014	0.012
Cross sim.	0.835	0.687	**0.433**	0.419	0.368	0.345	0.311	0.298	0.287	0.337	0.338	0.306
Normalized importance	0.8	1.0	1.0	0.8	0.8	0.8	0.8	0.2	0.6	0.0	0.4	0.2
Strict optimisation	0.862	0.635	**1.023**	0.656	0.764	0.825	0.640	0.237	0.584	0.000	0.324	0.179

Plotting the three functions in a scatter plot is shown in Figure 6.6. Models can be selected based on a specific criteria for example the model with best linearity (model of 2 hidden states - black dot at the top) or the model with the best compactness and cross state similarity, which in this case study is found in the same model (model of 10 states - black dot at the bottom). The red lines show the direction for criteria best values. Optimising the candidate space of 12 models using the strict optimisation has selected model of 4 states is the best model with maximum value of 1.023.

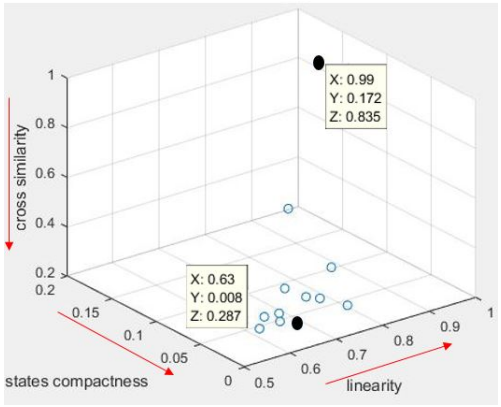


Figure 6.6: 3D visualization of the criteria in case study 1

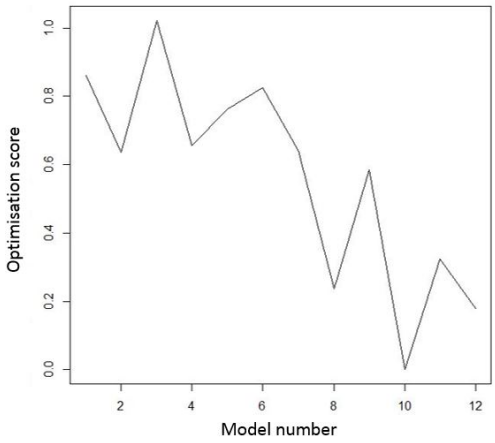


Figure 6.7: Strict optimisation scores in case study 1

In Figure 6.7, the X-axis represents the number of models where the first model consists of two hidden states and the last model consists of 13 hidden states. Clearly, model of 4 states is the highest whereas the 10th model, which has 11 states, is the worst model since it has 5 unimportant states. State importance threshold that is used in our experiments is 50% however, different thresholds can be tested as presented in Table 6.4.

Table 6.4: State coverage and importance in case study 1

Model	State coverage percentage													# of unimportant states of two thresholds	
														50%	30%
2s	9.60	96												1	1
3s	100	99	99											0	0
4s	94	94	92	50										0	0
5s	94	7	92	94	50									1	1
6s	48	94	94	98	98	96								1	0
7s	22	50	92	92	92	91	92							1	1
8s	90	92	46	88	94	88	50	94						1	0
9s	11	33	23	92	92	91	41	89	92					4	2
10s	56	92	91	92	92	49	92	41	91	50				2	0
11s	93	18	80	36	22	90	92	13	15	89	92			5	4
12s	94	54	98	87	90	91	41	11	11	88	87	53		3	2
13s	94	84	93	91	91	92	36	34	17	14	77	53	92	4	2

Based on the 50% threshold, 11states model has the highest number of unimportant states where 5 out of 11 states represent less than 50% of cases. Both 9states model and 13states model are also non-representative models where both of them have 4 unimportant states. On the other hand, 3state model and 4states model are the best model since all of their states represent more than 50% of cases. However, 4state model has the best score to balance between other optimisation criteria; linearity, compactness and cross state similarity.

6.7.1 Healthcare process analysis using the strict model

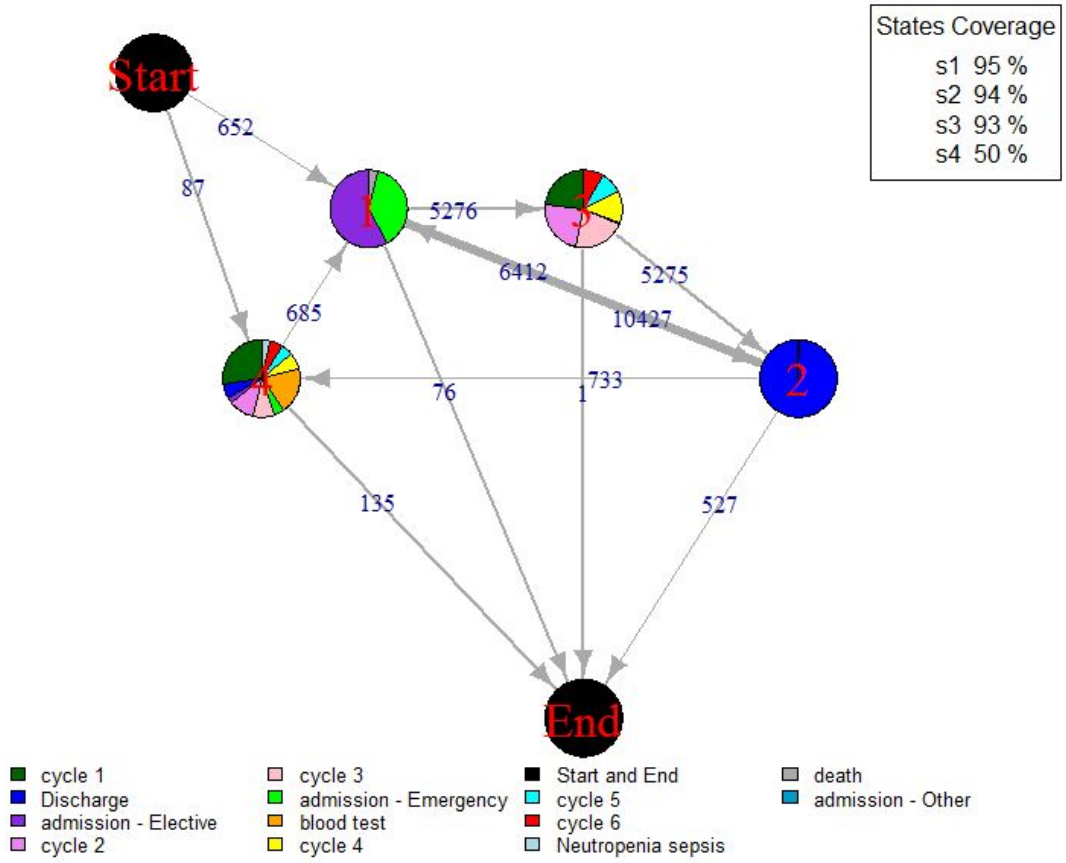


Figure 6.8: The best model of case study 1 selected by the strict optimisation

The strict model, which is the model that is selected by the strict optimisation, is presented in Figure 6.8 and demonstrates the mainstream pathway of breast cancer who had chemotherapy sessions as adjuvant treatment between 2004 - 2013. This model shows the healthcare process for multiple visits of breast cancer patients. It should be noted that, the number on the edges linked to the Start and End nodes represent case frequency, which shows how many number of patients were on that edge, however, the number of the edges in between states represent the absolute frequency that reflects how many times this transition has happened in the data. The state number is random and does not hold any meaning also, loop on same state is removed to simplify the model. Moreover, very low frequency events of less than 5 occurrence might not appear in the graph node.

The majority of the patients, n=652, have started their healthcare journey through admission either elective or emergency as shown in state1. Interestingly, the transition from (admis-

sions(s1) → Discharge(s2)) means that, most of the patients have been discharged directly after admission where no care event was recorded in between. A possible hypothesis for interpreting this pattern is that it might be an indication to the primary treatment that was given to the patients before starting chemotherapy treatment, which is not our focus on extraction this case study since the patients here have adjuvant chemotherapy that is given post a primary treatment. Another possible interpretation of the pattern; admission then discharge without care events in between, is that the incompleteness of PPM data extract that was given to us. Digging deeper to the data has confirmed our first hypothesis, this pattern of care, admission followed directly by discharge, has mostly happened in the beginning of patients records where the primary treatment has occurred.

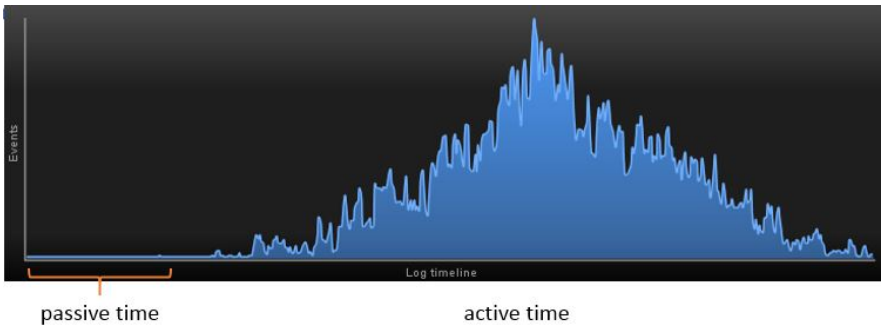


Figure 6.9: Active and passive time for breast cancer process in PPM

Figure 6.9 illustrates that the process of breast cancer patients go through active time and passive time. Active time is when the treatment starts usually by getting chemotherapy sessions for this case study in particular, to the end of the chemotherapy course which is mostly 6 cycles. Passive time is the time when a patient admitted to the hospital for any other reason except chemotherapy session. The absolute frequency of the passive time pattern, which is the transition from state 1 to state 2, in this case study is 6412 and this pattern could happen before starting the treatment. Time gap between passive time and active time may vary where it might be months or weeks before chemotherapy.
On the other hand, the pattern (Discharge(s2) → admission(s1)) is an expected sequence of the multiple visits for patients. After the admissions state, a patient moves to chemotherapy sessions which include cyclel, cycle2, cycle3, cycle4, cycle5 and cycle6. Once a patient finishes their chemotherapy session they will be discharged. A blood test may be taken at the beginning of the healthcare process and/or for the upcoming cycles before taking their chemotherapy session to avoid an acute event such as neutropenia sepsis.
Interestingly, the model has allocated a particular state (s4) for highly different patterns of chemotherapy process. In order to have a deeper insight of the process inside each state we can use any process mining tool and explore the processes of each states individually.
For instance, patterns of care inside state 4 in our model can be investigated using the Dotted chart in ProM as illustrated in Figure 6.10. This Figure shows vertically the number of patients

and horizontally the process instances where care events are ordered by the time when a case started and coloured by states numbers. It can be clearly seen that, two main different patterns represent two different groups of patients.
Using the Traces explorer plugin in ProM, the first group, which is the top pattern A, are patients who have started their treatment by chemotherapy cycles directly where no admissions event are recorded and they represent 44 cases. The second group, the bottom pattern B, are patients who had blood test to check if they may experienced an acute event such as neutropenia sepsis and they represent 43 cases while 284 cases have blood test in the middle of their treatment. This state represents 50% of cases.

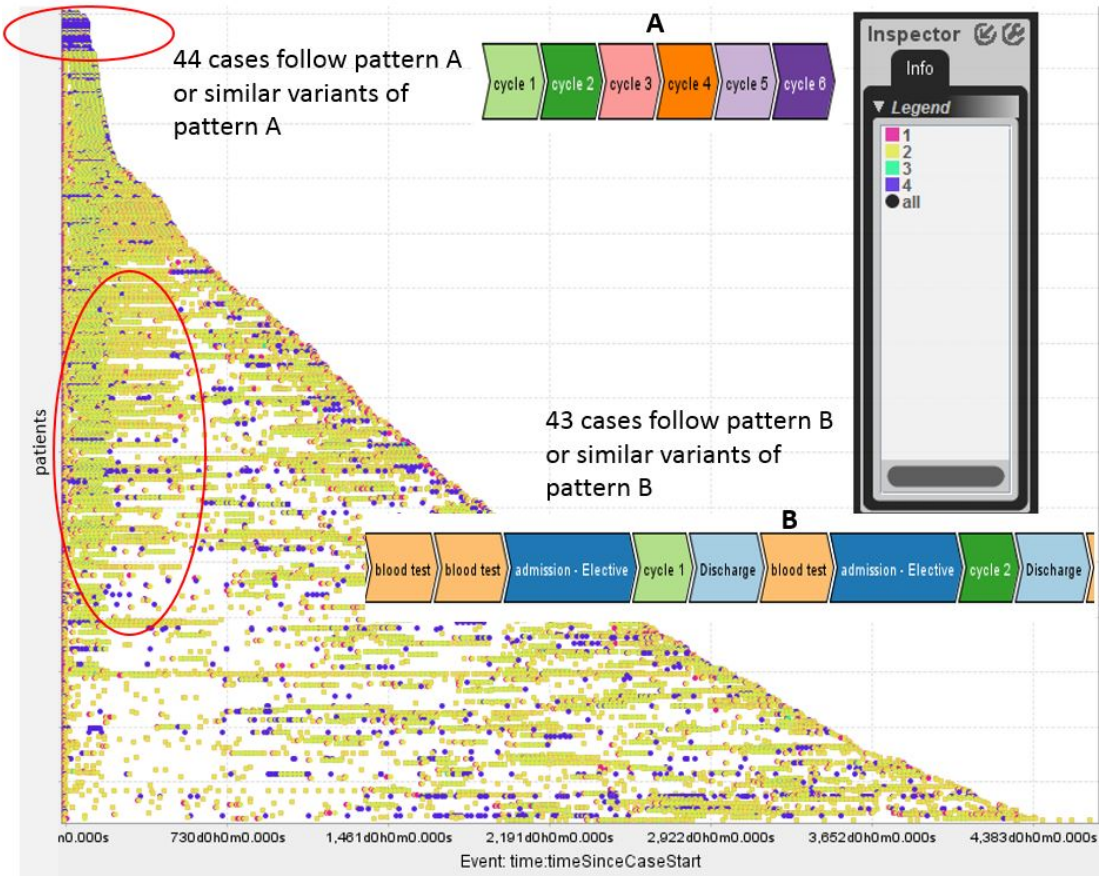


Figure 6.10: Dotted chart for examining patterns inside state 4 combined with two main patterns generated from Traces explorer

By analysing the healthcare model and investigating the distinct events of each state we could relabel the states initially based on the main events that contribute in forming the states, see Figure 6.11 which shows the model with initial states labels.

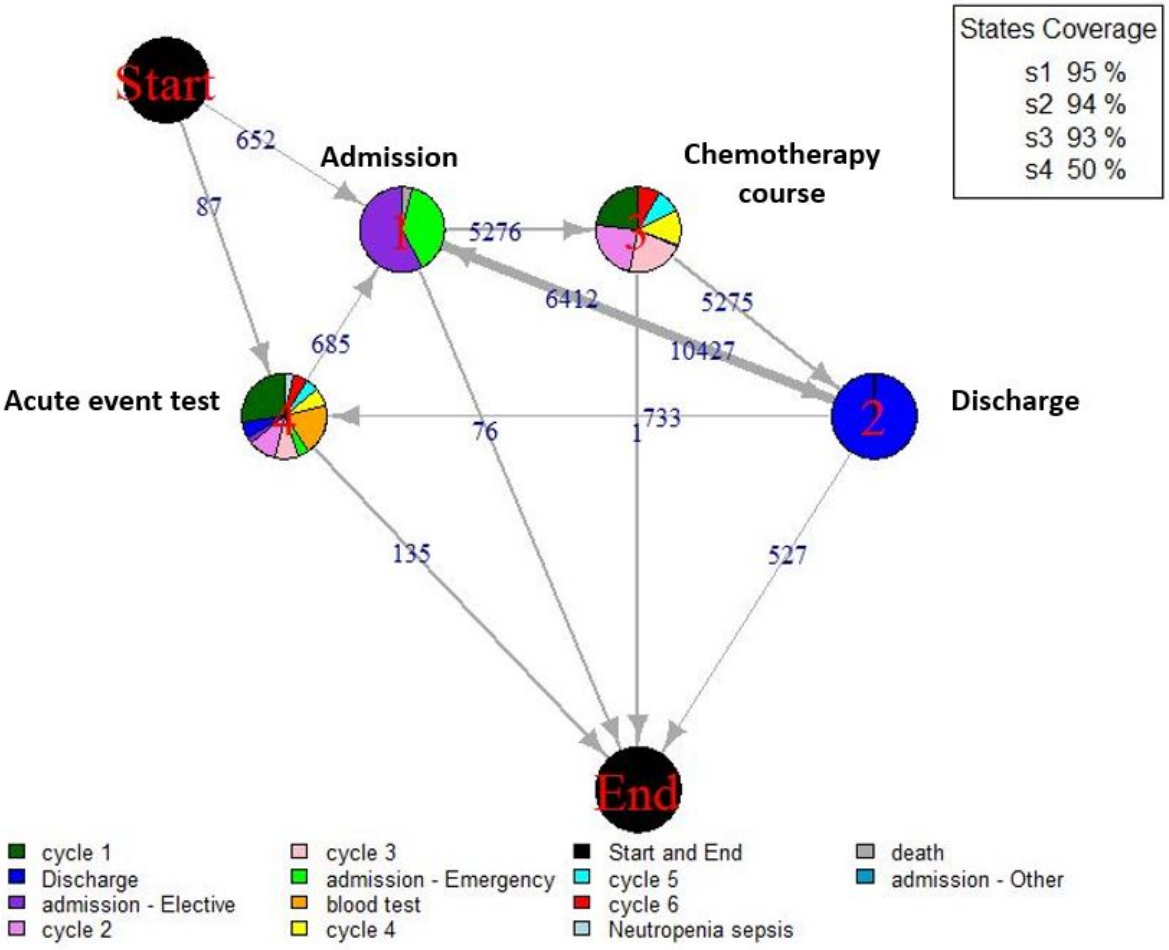


Figure 6.11: Initial labelled states of breast cancer process in case study 1

2- Soft optimisation for models' candidate space

Selecting the best model with flexibility toward state importance can be done using our soft optimisation (Equation 5.5). Our method has optimised the space of candidate models for this case study and the criteria are calculated as displayed in Table 6.5.

Table 6.5: Criteria calculation in case study 1 (soft optimisation)

Model	1	2	3	4	5	6	7	8	9	10	11	12
States	2s	3s	4s	5s	6s	7s	8s	**9s**	10s	11s	12s	13s
Linearity	0.99	0.66	0.73	0.62	0.66	0.69	0.55	**0.74**	0.63	0.60	0.57	0.60
Compactness	0.172	0.034	0.022	0.017	0.014	0.014	0.009	**0.009**	0.008	0.013	0.014	0.012
Cross similarity	0.835	0.687	0.433	0.419	0.368	0.345	0.311	**0.298**	0.287	0.337	0.338	0.306
Soft optimisation	1.078	0.635	1.023	0.820	0.955	1.031	0.800	**1.183**	0.973	0.861	0.810	0.895

The best model is a model of 9 states, which is presented in Figure 6.13, where it has the maximum score of the optimisation function which is 1.183. The worst model is model number 2 which is the model of 3 state where it has 0.635 score as plotted in Figure 6.12

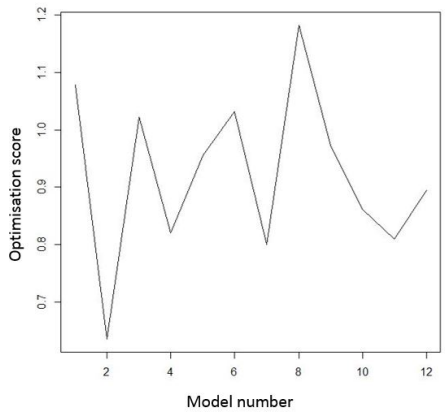


Figure 6.12: Soft optimisation scores in case study 1

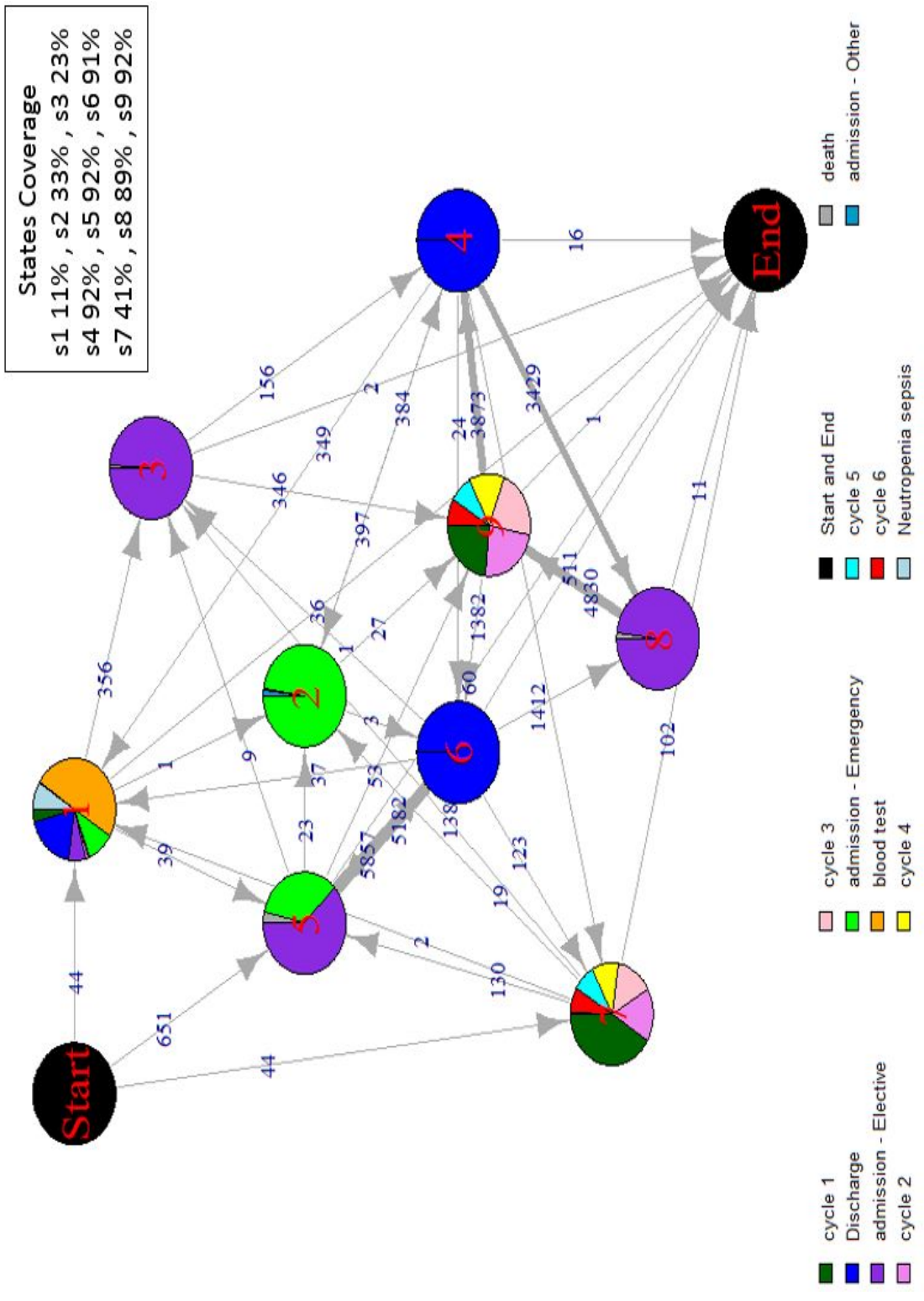


Figure 6.13: The best model of case study 1 selected by soft optimisation

6.7.2 Healthcare process analysis using the soft model

By relaxing the constraint of state importance, our optimisation has selected the 9states model as the best model. The soft model, that is visualized in Figure 6.13, represents the whole journey of breast cancer patients starting from the need for care to the end of recorded treatment. Hence, the model demonstrates multiple visits of the patients. Interestingly, state 4 in the first model has been split into two different states; state 1 and state 7. State 7 is dedicated for patients who started their chemotherapy directly without admission, which is an unexpected pattern. State 1 shows the pattern of care for acute event patients and blood test.
This model provides further details in terms of discharge and admissions events. The model has distinguished between two contexts of discharge event. The first context is that a discharge follows admissions directly with no care event in between and this was presented in the strict model as well. This context is shown in the link between state 5 and state 6, state 2 and state 6, state 3 and state 4 and lastly between state 2 and state 4. This state, as discussed earlier, may represent history record of the number of times when a patient needed care previously. The second context is discharge that happens after finishing follow up chemotherapy cycles as shown in the transition between state 9 to state 4.
On the other hand, admissions are split over 4 states which are state 5, state 2, state 3 and state 8. Firstly, state 5 which includes both types of admission (elective-emergency) this state is allocated for the first visits for most of the patients. State 2 represents only emergency admission that might happen in the follow up visits. State 3 is for elective admission that mostly happen after recovering from an acute event, which was occurred in state1. Lastly, state 8 which shows elective admission as well but this state is for patients with non-acute event who are for following visits.
Clearly, this model provides details that are unneeded for discovering the mainstream pattern in addition to the low coverage of some states as shown in Figure 6.13.

6.8 Discussion

In order to identify possible critical healthcare states and get interesting insights on the given healthcare process, we suggest using the soft model for analysing process outcomes since this model provides further detail about the processes. Analysing the model that is shown in Figure 6.13 resulted in three interesting pathways we would like to consider these pathways as outcomes of the experiments. We can categorize the outcomes into good and bad outcomes. Bad outcomes are the result of the presence of two events: death or acute event. Pathways that do not include any of these events are considered a good outcome.
1- Death event:
There are 79 death cases out of 739 patients = 10% of this case study. The death event can occur in several states; state 5, state 8, state 7, state 3 or state 2. Mining the direct relations between events for cases where a death event occurs might help in understanding when the

death is likely to happen.

In order to do this, cases with death events are selected, then events are augmented with their state number. Using the BupaR tool [136] (version 0.4.2) in R, a precedence matrix (sometimes known as dependency matrix) is generated as displayed in Figure 6.14. The precedence matrix basically demonstrates the direct relations between events. Here, events are associated with states numbers. As we can see in the red rectangle, most of death events were recorded after discharge (57 after discharge in state 6 and 11 after discharge in state 4). Few of death cases were recorded after admissions. Apparently, this dependency matrix does not help in capturing the care events that have occurred before discharge.

Therefore, we have filtered out discharge and admissions events to explore previous care event as shown in Figure 6.15.

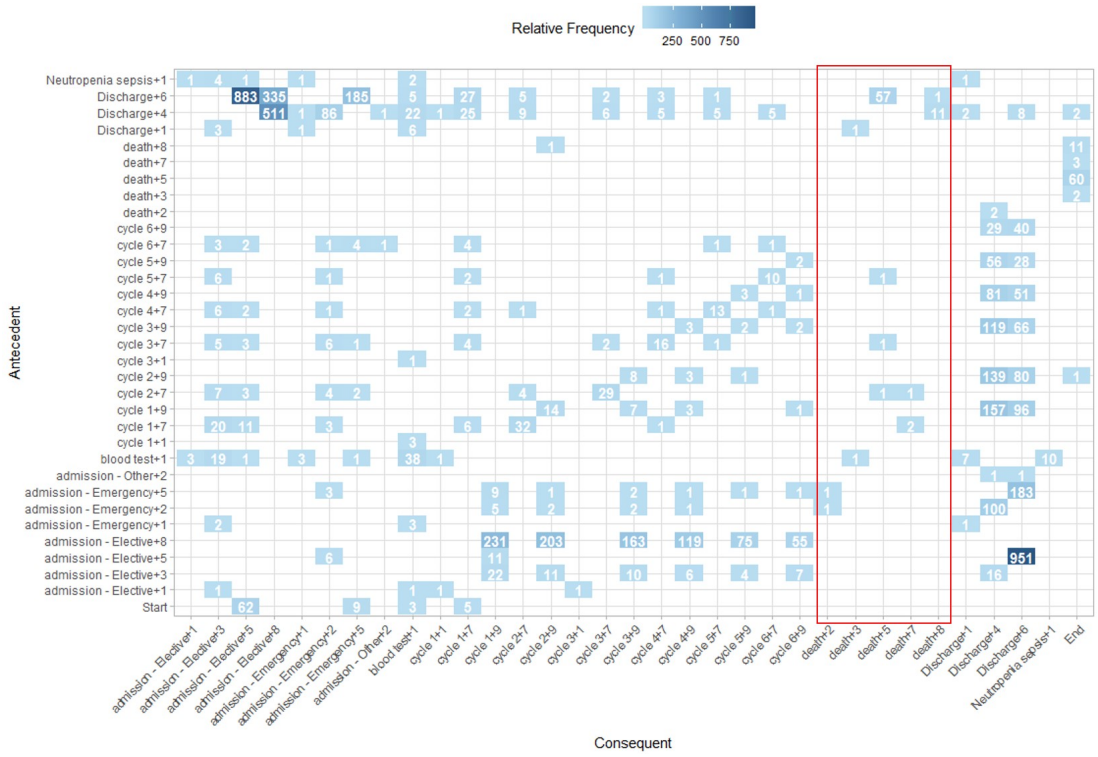


Figure 6.14: Direct relations of events associated with states for cases have death event (axes label is in the form of 'event+state number')

Figure 6.15 shows different care events associated with state numbers that happened before death. The number of deaths in patients who transited from state 9 is higher than the number of deaths for patients who transited from state7 with total of death 57 and 18 for state 9 and state 7 respectively, while only 4 death cases occurred after blood test that happened in state 1. However, this might not be a significant increase of state 9 over state 7 since the number of patients who came through state 9 is 431 whereas the number of cases who came from state 7

is 51 cases.

Analysing the correlation between chemotherapy cycles and death has shown that death is likely to happen after cycle 6 for patients who came from state 9 with 13 cases. In contrast to patients who came from state 7 where the death was correlated equally for cycle1 and cycle6 with 5 death cases in each.

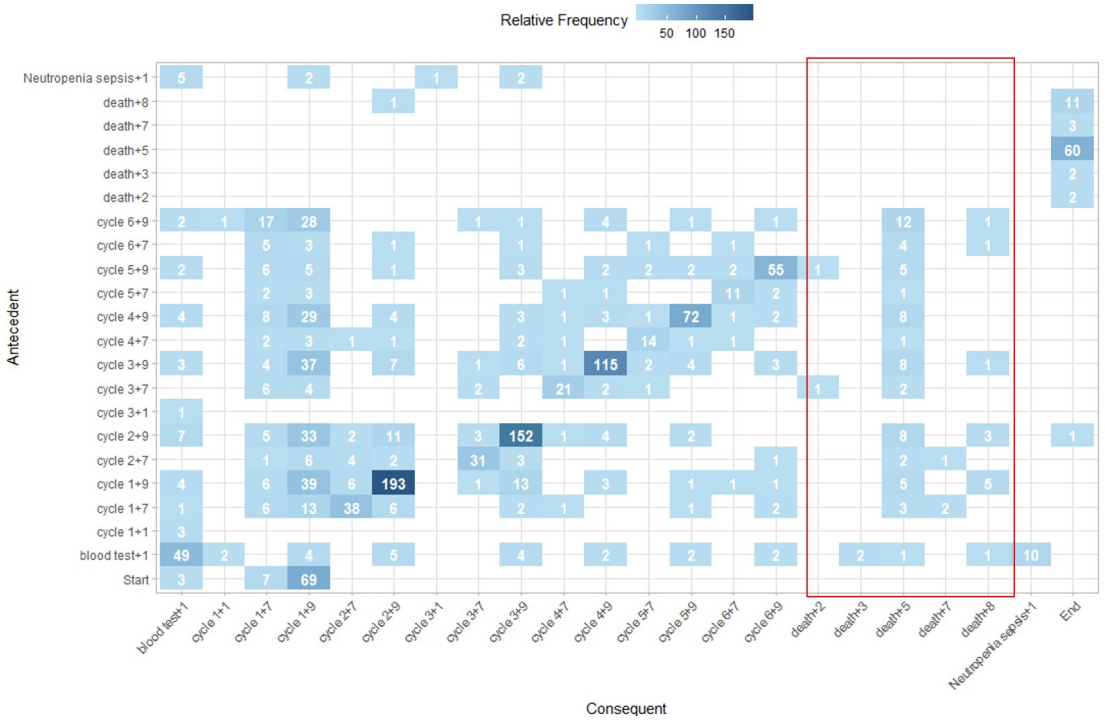


Figure 6.15: Direct relations of death events associated with states

2- Acute event:

There are 20 patients who have an acute event such as neutropenia sepsis which represents 2.7% of this case study. The neutropenia sepsis event is recorded 46 times and the maximum repetition was 5 times for one single case who died. There are 3 patients who had neutropenia sepsis have died as shown in Figure 6.16 which represents 15% of this case study. The occurrence of neutropenia sepsis has no clear correlation with a particular chemotherapy cycle. It may happen after any cycle as shown in Figure 6.16.

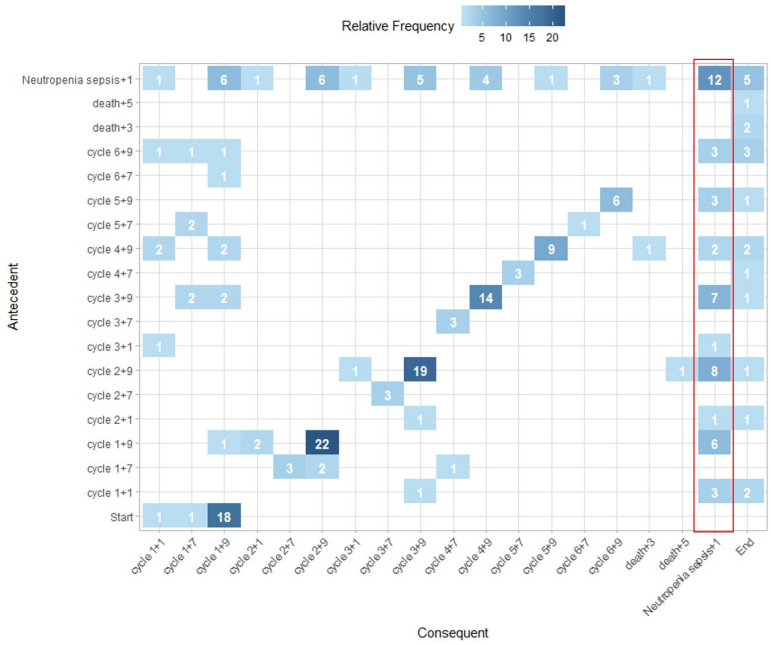


Figure 6.16: Direct relations of neutropenia sepsis events associated with states

In summary:
1- A good outcome is a pathway that is free from two events either death or acute event. Example of a good outcome can be captured in this pattern: start, state 5, state 9, state 4 and for the upcoming visits starting from state 8 through state 9 then state 4.
2- Bad outcome can be seen in two types of events either death or acute event. Example for bad outcome is the pathway of ; start, state 1, state 5 or end which shows that patient has a cute event and then ended by death or there is no recorded events.

6.9 Model Evaluation

In this section, three aspects for models evaluation are discussed which are models selection validation, model evaluation using process mining conventional metric and evaluation based on domain experts.

6.9.1 Model selection validation for case study 1

In this section we aim to check the validity of model selection of this case study against issues which were key motivations for our optimisation. Also, the same model validation metrics are applied on models that are selected by the Bayesian Information Criteria (BIC) in order to provide a comparison between the two selection methods.
The models presented in Figure 6.19 are visualized using the original function of the 'SeqHMM'

package. The reason for not using our new visualization function here is because the goal is to investigate validation aspects not to visualize the process models. Hence, using the old visualization function is sufficient for this purpose.
Model validation is an important step that comes after model selection. As discussed earlier in Chapter 4, there are three main issues that may generate undesirable abstraction model which are the existence of highly connected states that reflects the potential of higher level of abstraction, similar states and unimportant states.
We will validate model selection against these issues. The way of quantifying these issues are provided in Chapter 4 and here is a brief reminder:
1- The issue of strong connected states is identified using the graph theory technique for strong components detection.
2- The issue of multiple similar states is detected using state type similarity measure, where states are similar if they have same state type for instance, simple, composite and complex and same event types that occupied 80% of both states. A list of all states types for this case study and case studies that will be discussed in the following chapter is provided in Appendix C. The presence of similar state is scored by counting how many same-type similar states a model has.
3- The issue of unimportant states corresponds to state converge where it shows the percentage of how many cases are involved in a state.

In the following section, the validation for models selection is presented and the methods that are used for identifying validation metrics are explained in detail likewise. The proposed optimisation methods both strict and soft have selected fewer number of states compared with BIC. The best model that is selected using the strict optimisation is the 4state model and using the soft optimisation is the 9state model whereas BIC has selected a model of 12 states as the best model. Figure 6.17 shows the best model for our optimisation with maximum score of optimisation whereas the best model using BIC has the minimum value.

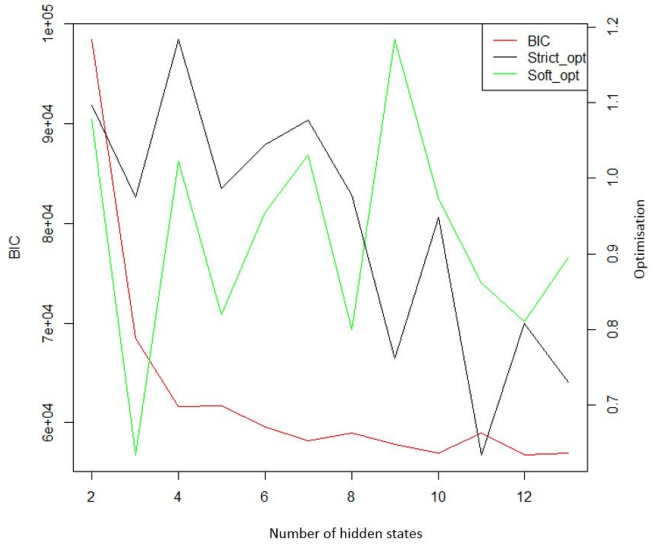
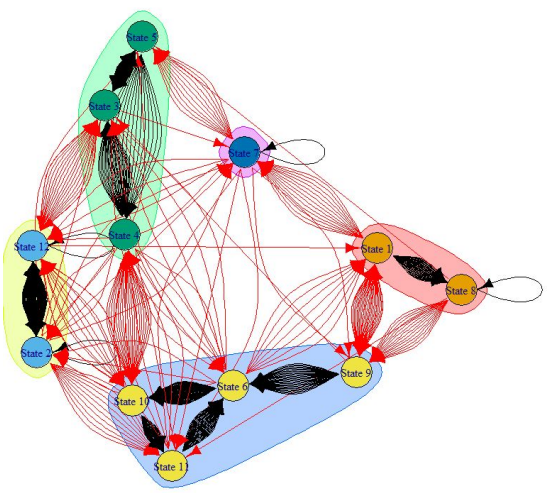


Figure 6.17: Best models of different metrics in case study 1

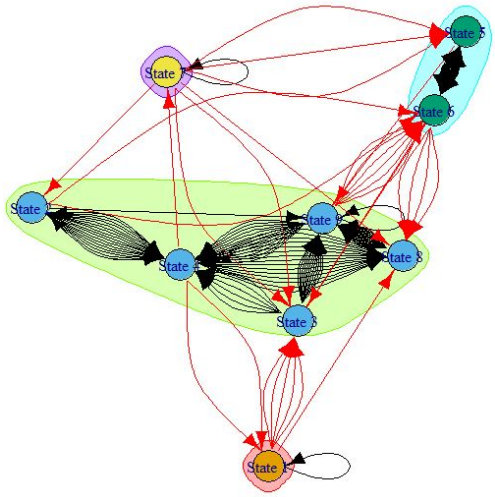
1- Connected components:

The highly connected states are abundantly observed in the 12states model that is selected by BIC, as shown in Figure 6.18(a). There are 4 possible higher abstraction that can be detected in this model where each cluster must have at least two states.

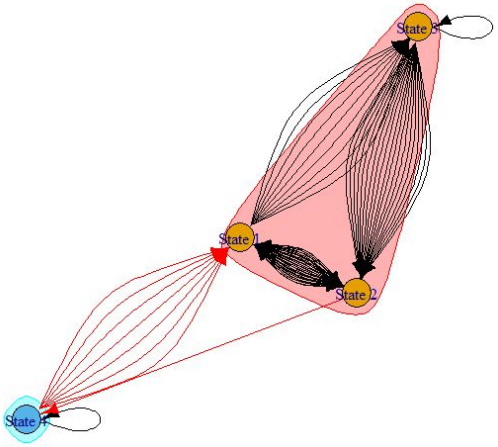
In contrast to Figure 6.18(b) and (c), the number of connected states is few where there are 2 and 1 clusters of states in the models selected by soft and strict optimisation respectively.



(a) model selected by BIC



(b) model selected by soft optimisation



(c) model selected by strict optimisation

Figure 6.18: Connected components detection in case study 1

2- Similar states:

The model in Figure 6.19(a) has three same-type similar states. Detecting multiple similar states for all these types have resulted in finding:

1- Production states (Discharge) are shown in state 1, 2, 3 and 10.

2- Simple states (Admission-Elective) are state 5, 9 and 11.

3- Composite states (Chemotherapy cycles) are state 6 and state 7.

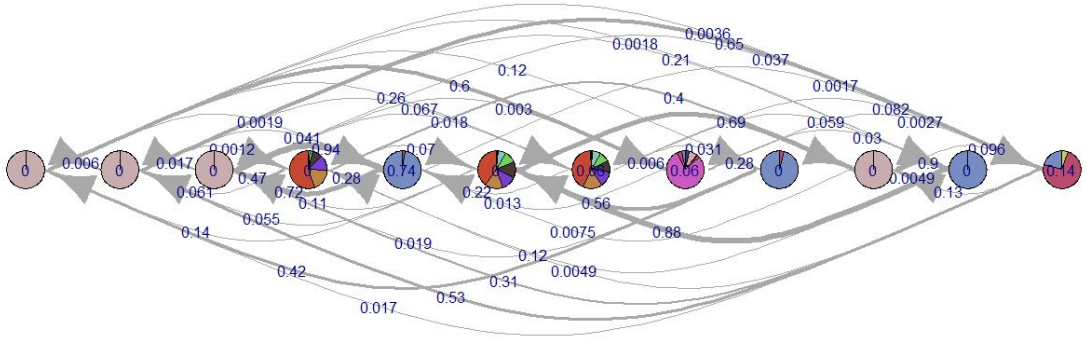
The model presented in 6.19(b) has three same-type similar states as follows:

1- Production states (Discharge) are state 4 and state 6.

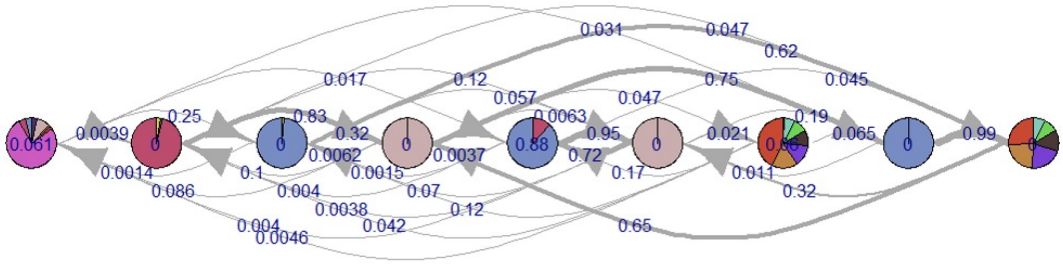
2- Simple states (Admission-Elective) which are states 3, 5 and 8.

3- Composite states (Chemotherapy cycles) are state 7 and state 9.

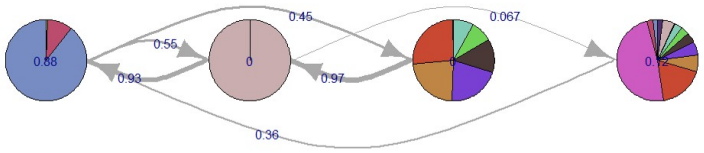
The model presented in 6.19(c) has no similar states all states are constructed from different event types.



(a) similar states model selected by BIC



(b) similar states of soft optimisation model



(c) similar states of strict optimisation model

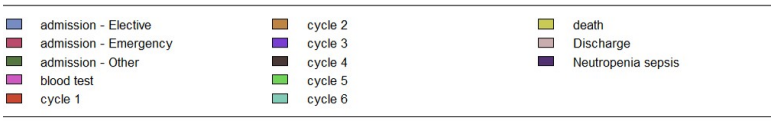


Figure 6.19: Similar states detection in case study 1(states numbering starts from left to right)

3- Unimportant states:

Based on state importance percentage that is extracted and reported in Table 6.4, we use the threshold of 50% to determine if a state is important or not. The threshold can be adjusted based on user preference. The result showed that, as expected the states in model selected by the strict optimisation were all significant. Unlike to states in soft model where this model contained 4 non-significant states. It should be noted that, as discussed in Chapter 5, the issue

of unimportant states might be related to bad model initialization that cannot be addressed in the optimisation.
A summary of model selection validation metrics is presented in Table 6.6. Clearly, the three issues are more likely to be found in the model that is selected by BIC whereas these issues are hardly observed in the model that is selected by the strict optimisation. Although the soft model and BIC model have the same number of similar states, the number of states for each similar type in BIC model is higher than soft model. For example, there are 4 states of production type (Discharge) in BIC model whereas only 2 states of production type (Discharge) in the soft model. Also, the soft model has better scores of all the proposed criteria compared to BIC model.

Table 6.6: Validation metrics of case study 1

Issues	**strict optimisation**		**soft optimisation**		**BIC**	
	found	count	found	count	found	count
strong connected components	yes	1	yes	2	yes	4
similar states	no	-	yes	3	yes	3
unimportant states (<50%)	no	-	yes	4	yes	3

6.9.2 Models evaluation based on process mining metrics

In this section we present the results of evaluating our models using all conventional process mining metrics that are discussed in Chapter 2 section 2.8 which are fitness, precision, f-measure, generalisation. In addition to the complexity metrics that are discussed in section 2.4.3 which are size, control-flow-complexity (CFC) and structuredness (struct). According to the review study of process mining discovery evaluation that is conducted by [47], there are three main algorithms that can generate reliable results for process model evaluation. These algorithms are Inductive Miner, Split Miner and Evolutionary Tree Miner and they have outperformed other algorithms in terms of model evaluation metrics. Thus, we have used Inductive Miner (IM), which is discussed in Chapter 2, and Split Miner (SM) [137], which is built on the same principles of inductive miner but it aims to include the choice construct of the process as well, for process model discovery. However, the Evolutionary Tree Miner is computationally expensive and required high performance computer which is not applicable in our case, hence it is excluded from our experiment.
Table 6.7 shows process evaluation metrics of the extracted event log. These metrics concern about complexity, accuracy and performance (execution time). We aim to compare the process models metrics of the raw event logs and the process models of these logs after abstraction using both strict and soft optimisations. Table 6.8 reports the results of the discovered process models using IM and SM after using the strict and soft optimisation models for abstraction. In all three complexity metrics that are used for evaluation, there is a noticeable improvement in the process model complexity for the abstracted models. The size and CFC have improved in both models. However, in soft optimisation with the SM, the struct metric has a small degra-

dation compared to the original log of case study 1. This might be due to the shortcoming of the SM in building struct model as discussed in [137].
On the other hand, the accuracy metrics except generalization show an overall increase. In contrast to what has been mentioned in [138], where the author indicated that the precision metric is not significantly important since the process in healthcare are very flexible. Our results show that despite the flexibility of healthcare processes, the abstraction has improved the model precision in both strict and soft models. There is a small decrease in the fitness of case study 1- and consequently on the model generalization- after soft abstraction. However, compared with the increase of the precision and F-measure, we believe this decrease in fitness does not have a significant affect on model accuracy measurements. Furthermore, the execution time that is needed for discovering process models using the IM and SM in both models has improved dramatically after abstraction which is an expected result for smaller size models. Generally, the abstraction methods have improved model complexity, accuracy and performance metrics. Analysing the process variation percentage before and after abstraction has showed that the variation percentage has clearly decreased using both models from 86% in the original event log to 76% and 75% in strict and soft models respectively.

Table 6.7: Process evaluation of case study 1 before abstraction

Logs	Variation	Discovery algorithm	Complexity			Accuracy				Execution time
			size	CFC	struct	fitness	precision	f-measure	generalization	
Case study 1	86%	IM	38	29	1	1	0.482	0.651	1	1089 ms
		SM	42	33	1	0.91	0.50	0.64	0.917	206 ms

Table 6.8: Process evaluation of case study 1 after abstraction using strict and soft optimisation

Logs	Variation	Abstraction	Discovery algorithm	Complexity			Accuracy				Execution time
				size	CFC	struct	fitness	precision	f-measure	generalization	
Case study 1	76%	Strict	IM	17	12	1	1	0.653	0.790	0.999	601 ms
			SM	18	13	1	0.963	0.711	0.818	0.963	101 ms
Case study 1	75%	Soft	IM	27	19	1	1	0.600	0.75	0.99	926 ms
			SM	28	17	0.821	0.899	0.739	0.811	0.902	229 ms

6.9.3 Models evaluation based on domain experts

As a final step of the evaluation, we aim to evaluate the utility of our method using domain expert's opinion. For this purpose we arranged a meeting with two domain experts who work closely with the PPM system and have a strong background of cancer healthcare processes in the PPM and we presented the results of the abstracted models. To simplify the discussion in the meeting, the case study 1 was used first to explain the steps of the research method and the criteria of model optimisation in detail. Then the results of the abstracted models for both optimisation types are demonstrated and discussed. The discussion mainly aimed to ensure that the models are sensible and indeed described the general healthcare processes for breast

cancer in the PPM data.
The experts have agreed on the correctness of the modelled processes. Also, they have clarified some possible reasons for the pattern (Admission then Discharge) without care events in between which has been observed in both models. The possible reason for this pattern is that a patient needed a care for other health conditions that are not related to the cancer or that represent the primary treatment, as we have anticipated. Moreover, the experts have confirmed that the chemotherapy sessions cannot be given without admission to the hospital. In contrast to our model, this pattern is shown in case study 1 in the model that is selected by the strict optimisation in state 4 and in the soft optimisation in state 7. Discovering this pattern in a specific state in HMM has shown the usefulness of using HMM to identify data recording issues, where the domain expert indicated the potential of not recording the care events by healthcare practitioners as supposed to be. Also, we have asked the experts about the large unexpected number of chemotherapy cycles that is shown in Figure 6.4. They justified the number of cycles in which might be affected by the way of how the data is recorded. Some practitioners may record a new cycle of chemotherapy, that is given after finishing the first course that was suggested, as a follow up cycle. This may happen often especially when the chemo-regimen would be the same for both courses. Finally, the domain experts have asserted the usefulness of our method and agreed on the improved understandability of the models provided. Our method has discovered the main pattern of care and provided conceptually-valid abstractions on complex healthcare processes. In addition of that, this method has successfully reduced the time of experts involvement in this research.

6.10 Conclusion

Testing the proposed unsupervised abstraction method on the case study of chemotherapy cycles of adjuvant breast cancer patients has shown promising results of discovering the healthcare process models without the need for domain experts in the abstraction stage. Models that are discovered using the strict optimisation have provided the general picture of the breast cancer healthcare processes. However, the soft optimisation tend to be more tolerant toward process detail that may happen for few cases. Models are evaluated and have reduced the complexity of original healthcare process based on complexity and accuracy metrics that are discussed. Issues found in BIC model might be found in soft model but they were mitigated in terms of the number of appearance of these issues. Unlike the strict model where was free of such issues. Finding a good or bad process outcome in our discussion is dependable on the presence or absence of an interesting event, such as death, or acute event such as neutropenia sepsis. After discovering the process models using state abstraction method, the task of similar cohort selection became easier. This chapter provided a relatively complex real world healthcare for applying our proposed unsupervised abstraction method. Our method has successfully discovered the mainstream process models and reduced model complexity. In the next chapter,

we aim to extract two further case studies from the PPM healthcare data in order to provide larger scale of complexity and test the proposed method as well.

Chapter 7

Further Experiments: Case study 2 and case study 3

7.1 Overview

The aim of this chapter is to explain how to provide more complexity of breast cancer healthcare process in order to test our method for larger scale of complexity than the one discussed in Chapter 6. Two different case studies are extracted from the PPM data extract. We apply the same proposed method, that is applied in the previous chapter, on these case studies to reduce process models complexity and consequently discover the general process of the given healthcare. Then the abstract process models will be validated and evaluated as well. To highlight a further functionality of state based model, a new strategy that helps in selecting similar patients is discussed. The focus of this strategy is on process perspectives similarity.

7.2 Case study 2: Different treatment types of breast cancer patients

Our hypothesis for adding further healthcare process complexity is that; including larger sample size and wider scope of care events increases the complexity of healthcare processes. Adding larger sample size of patients can be done by extracting patients who treated with different treatment types of breast cancer chemotherapy such as the type of neoadjuvant and palliative treatments in addition to the adjuvant treatment, that was used in case study 1. On the other hand, including wider scope of care events can be achieved by extracting different care events that can show the complete process of care. In contrast to case study 1, in which care events were limited to chemotherapy cycles only.

7.2.1 Extraction criteria

The same inclusion criteria from the previous case study is used in addition to different treatment types of chemotherapy. Three types of treatments are extracted using SQL and based on the field of *integerent* in *chemoregimens* table of PPM database, see Figure 6.3. Treatment types differ based on the order of chemotherapy event; for instance, if chemotherapy is given before surgical intervention (neoadjuvant) or after surgery to prevent recurrent cancer (adjuvant) or (palliative) treatment type to improve patient life quality. Figure 7.1 shows that the adjuvant treatment is the more frequent treatment that is given to the PPM extract patients.

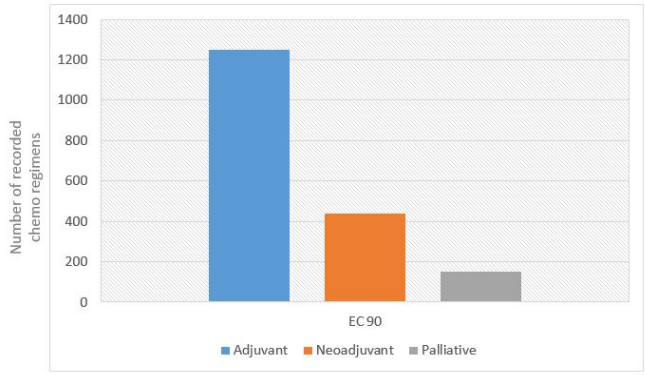


Figure 7.1: 'EC 90' regimen with different treatment types

The focus of extracting care events in this case study is broader than the scope of healthcare events in case study 1. The care events in this case study include admission, ward stay, discharge, diagnosis, visiting outpatient clinic, chemo-regimen start, blood test, microbiology test, surgery, radiotherapy, death and chemotherapy session, which represents chemotherapy cycles in general. The extraction criteria has selected 981 patients with high process variation percentage 99% and high variable length traces that resulted in high sparse space. The length of the process instances is ranging from very short processes with only 8 events to very long processes with 926 events as shown in Table 7.1.

Table 7.1: Log characteristics of different treatment types of breast cancer

total cases	distinct event	total events	variants	variation (%)	nulls	case length		
						min	avg	max
981	14	79453	979	99	828953	8	81	926

There are 14 distinct events based on our selection scope. Events names and frequency are displayed in Figure 7.2 below.

Activity	▲ Frequency
Discharge	18,738
ward stay	18,727
admission - Elective	16,814
chemotherapy session	11,501
regimen start	3,521
diagnosis	2,175
admission - Emergency	1,919
blood test	1,645
microbiology test	1,454
visiting outpatient clinic	981
surgery	973
radiotherapy	812
death	188
admission - Other	5

Figure 7.2: Screenshot of case study 2 events and frequency

7.2.2 Models learning and decoding

Applying our method on this event log has resulted in populating 13 possible models with maximum 14 hidden states, which is the upper bound based on the number of distinct events, as shown in Table 7.2.

Table 7.2: Learning HMMs with different number of hidden states in case study 2

Model number	Number of states	Iterations =(MX)3000	Initial log	Final Log likelihood	BIC (Bayesian information criterion)
1	2	277	-214510.6	-139404.2	279135.7
2	3	128	-206149.0	-114101.10	228732.5
3	4	246	-207841.3	-92955.16	186666.3
4	5	186	-229435.4	-82194.50	165393.2
5	6	222	-215143.9	-74725.21	150725.4
6	7	291	-204938.5	-74555.86	150680.0
7	8	197	-215066.3	-72251.40	146387.0
8	9	221	-221676.6	-71336.22	144895.2
9	10	421	-206018.9	-70062.12	142708.0
10	11	1061	-207672.0	-69126.59	141220.6
11	12	840	-211616.9	-67370.60	138114.8
12	13	1404	-204456.4	-66764.45	137331.2
13	**14**	**1036**	**-216796.5**	**-66426.67**	**137107.0**

The standard metric of selecting the best model of HMM, BIC, has selected a model with 14 states as the best model. However, we believe that, less number of hidden states can discover the mainstream pattern of the process. All models are decoded using the Viterbi algorithm as a pre-step for models candidate optimisation. The results of our method are explained below.

7.2.3 Optimisation

We have run the two different versions of our proposed multi-objective function to optimise models candidate space:

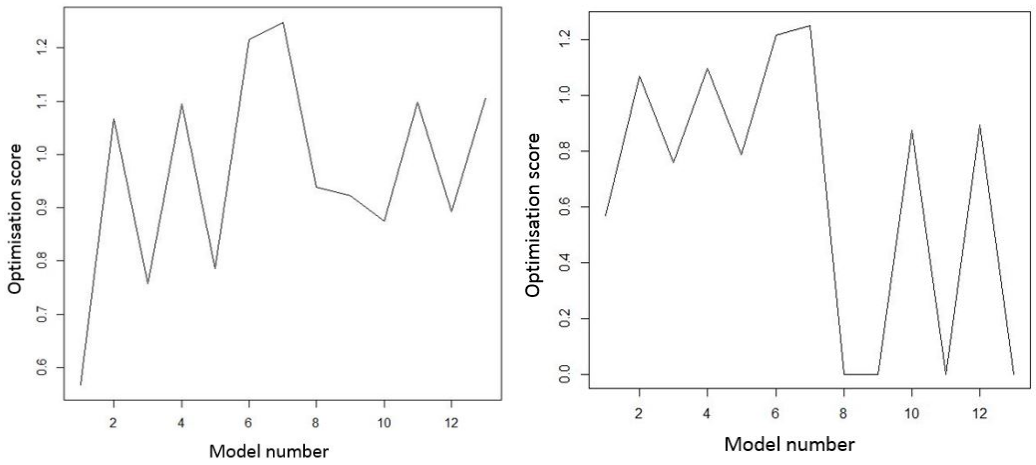
Strict and soft optimisation for models' candidate space

Using our method both types of optimisation have selected the same model as shown in Table 7.3. Model of 8 states is the best abstracted model based on our criteria. This model has the maximum score of strict optimisation and soft optimisation which is 1.248 because all the states in this model are important, based on the threshold.

Table 7.3: Criteria calculation in case study 2 (strict and soft optimisation)

Model	1	2	3	4	5	6	7	8	9	10	11	12	13
States	2s	3s	4s	5s	6s	7s	**8s**	9s	10s	11s	12s	13s	14s
Linearity	0.68	0.79	0.61	0.69	0.56	0.75	**0.72**	0.63	0.61	0.56	0.68	0.57	0.69
Compactness	0.246	0.086	0.036	0.023	0.015	0.013	**0.011**	0.011	0.012	0.017	0.019	0.024	0.035
Cross sim.	0.670	0.470	0.444	0.273	0.327	0.279	**0.186**	0.315	0.291	0.236	0.252	0.235	0.258
Normalized importance	1	1	1	1	1	1	**1**	0	0	1	0	1	0
Strict optimisation	0.567	1.067	0.758	1.095	0.786	1.215	**1.248**	0.000	0.000	0.876	0.000	0.893	0.000
Soft optimisation	0.567	1.067	0.758	1.095	0.785	1.215	**1.248**	0.939	0.923	0.875	1.098	0.893	1.104

Due to the absence of unimportant states in most of the models, the optimisation scores have not changed in these models. In contrast of models with a single unimportant state, where they have been penalised and their scores became zeros, see Figure 7.3(b).



(a) Soft optimisation scores in case study 2 (b) Strict optimisation scores in case study 2

Figure 7.3: Multi-objective optimisations scores in case study 2

The percentage of state importance over all models is presented in Figure 7.4. We can see that, four models only have one single unimportant state based on 50% threshold which are 9states model, 10states model, 12states model and 14states model.

Table 7.4: State coverage and importance in case study 2

Model	State coverage percentage														# of unimportant states of two threshold	
															50%	30%
2s	100	100													0	0
3s	100	100	100												0	0
4s	100	99	100	100											0	0
5s	100	100	100	100	99										0	0
6s	100	100	98	100	100	99									0	0
7s	100	98	98	100	99	99	100								0	0
8s	100	100	99	98	100	100	67	51							0	0
9s	83	99	98	100	99	100	23	63	100						1	0
10s	100	99	99	66	98	99	42	99	99	98					1	0
11s	100	99	99	98	100	65	67	98	99	99	99				0	0
12s	100	82	98	85	64	99	87	99	84	99	99	49			1	0
13s	100	100	99	99	99	98	95	66	68	80	99	99	98		0	0
14s	100	100	99	99	95	68	98	48	98	98	97	99	98	98	1	0

7.2.4 Healthcare process analysis

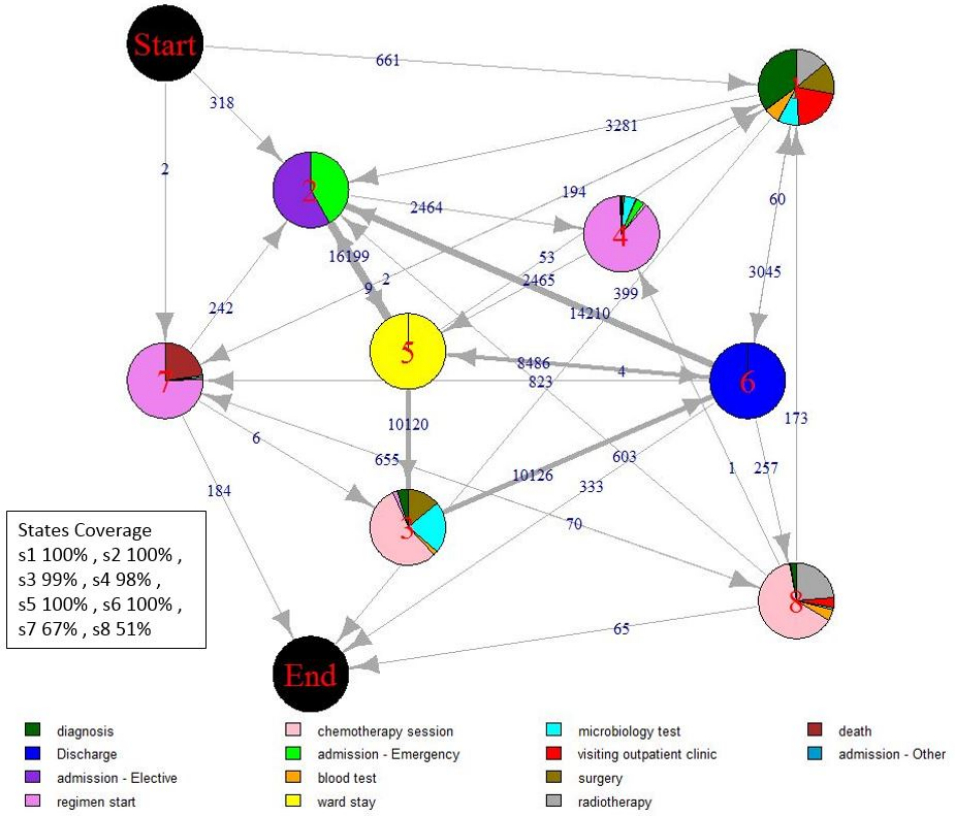


Figure 7.4: The best model of case study 2 selected by both strict and soft optimisations

Visualizing the process model using state abstraction is shown in Figure 7.4. This model shows the process of different types of treatment for breast cancer patients in PPM healthcare system. The general common pattern is that: patients start their treatment by visiting outpatients clinic in state 1 and have different laboratory tests and diagnosed. Then, patients can be admitted to the hospital either elective or emergency in state 2. After that, patients stay in a hospital ward in state 5. Inside the ward, patients may get several medical interventions depending on their need for instance, surgery or chemotherapy as shown in state 3. At the end of the medical interventions, patients are discharged to home either in the same day or after a while. Interestingly, this model has distinguished between different contexts of chemotherapy, radiotherapy and regimen start events where theses events are located in multiple states. The analysis of different states is done by the same way in case study 1 using the precedence matrix to show the direct relations between different events.

Analysing Chemotherapy states:
There are two main related states for chemotherapy which are state 3 and state 8. 552 of patients who have chemotherapy course only in state 3, 8 of patients took their chemotherapy only in state 8 and 409 of patients where overlapping between state 3 and state 8. Chemotherapy is also captured in state 4 but with a very few cases for example, there are 5 patients have their chemotherapy session in both state 3 and state 4 and only 7 cases have their chemotherapy in overlapping between state 3, state 8 and state 4.
The main difference between chemotherapy in state 3, state 4 and state 8 is that chemotherapy in state 3 must happen inside hospital ward, chemotherapy in state 4 happens mostly after admission emergency whereas chemotherapy in state 8 always happens without admission to the hospital. Chemotherapy in state 3 is mostly correlated with microbiology test however, chemotherapy in state 8 is mostly correlated with radiotherapy.

Analysing radiotherapy states:
In this case study, there are 812 of cases have experienced radiotherapy in their treatment. In contrast, 169 cases have treatments without radiotherapy. There are two main related states of radiotherapy which are state 1 and state 8. 624 cases have radiotherapy in state 1 and 188 cases have radiotherapy in state 8.
Both of the radiotherapy states happen outside the hospital in outpatient clinics. However, the main difference between them is that radiotherapy in state 1 is given after discharge and it is given as expected in outpatients clinics. Some exceptions have occurred such as, 2 patients have started with history record of radiotherapy and 49 cases have ended the treatment by radiotherapy.
Radiotherapy in state 1 is more correlated with diagnosis or visiting outpatients clinic. On the other hand, radiotherapy in state 8 mostly followed by chemotherapy, that is under taken in state 8.

For more explanation, (54%,n=101) of patients have radiotherapy directly after chemotherapy in state 8, however, the time gap between chemotherapy and radiotherapy is 23 days. Few cases, n=3, have instant sequences between chemotherapy in state 8 and radiotherapy in state 8 which means they are recorded with the same day. We believe this was recorded for a given care that happened in the past tense because the time gap between radiotherapy and the following events (diagnosis or admission electively) is 13 months as average.

Analysing chemotherapy regimens states:

In the model presented in Figure 7.4, the care event of chemotherapy regimen is located in three different states; state 3, state 4 and state 7. The difference between these states is that regimen in state 3 has always proceeded by being in a ward, regimen in state 4 has mostly happened after admission electively to the hospital whereas regimen in state 7 has mostly occurred after visiting outpatients clinics.

There are (41%, n=403) patients who have started their regimen in state 4 and only 6 patients who started their regimen in state 7. Interestingly, both of them =(41.6%, n=409) have been exposed to one single regimen throughout their care process.

In contrast,(58.3%, n=572) have changed their chemotherapy regimen multiple times where 10 patients have their chemotherapy regimen in state 7 and 562 have first regimen in state 4 and the follow up regimens in state 7.

Interestingly, the temporal pattern of regimen change is different in each state. The temporal pattern of regimen change for patients in state 7 is faster than patients who have started their regimen in state 4 as shown in Figure 7.5. In other words, patients in state 7 are likely to stay with their regimen just for 8 weeks as an average time, however, patients in state 4 are more persistent to their regimen where they stay 12 weeks as an average.

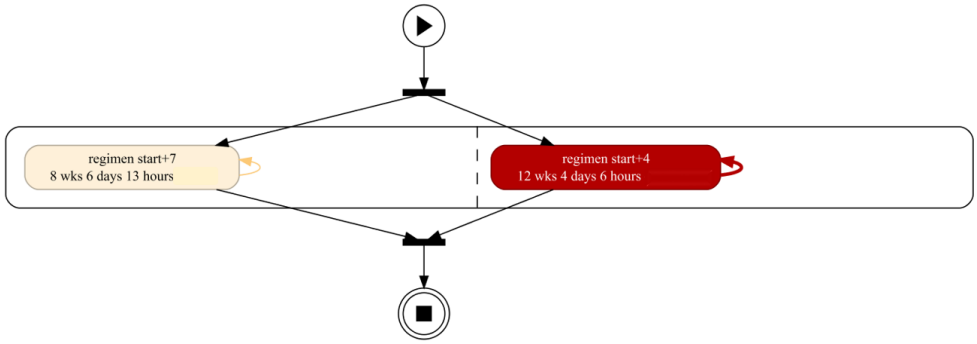


Figure 7.5: The temporal pattern of changing chemo-regimen. This Figure is generated from ProM log explorer

Based on the above analysis of the hidden states and by examining the distinct events related to states, we have provided initial labels for the states that will be validated later with the domain expert, see Figure 7.6.

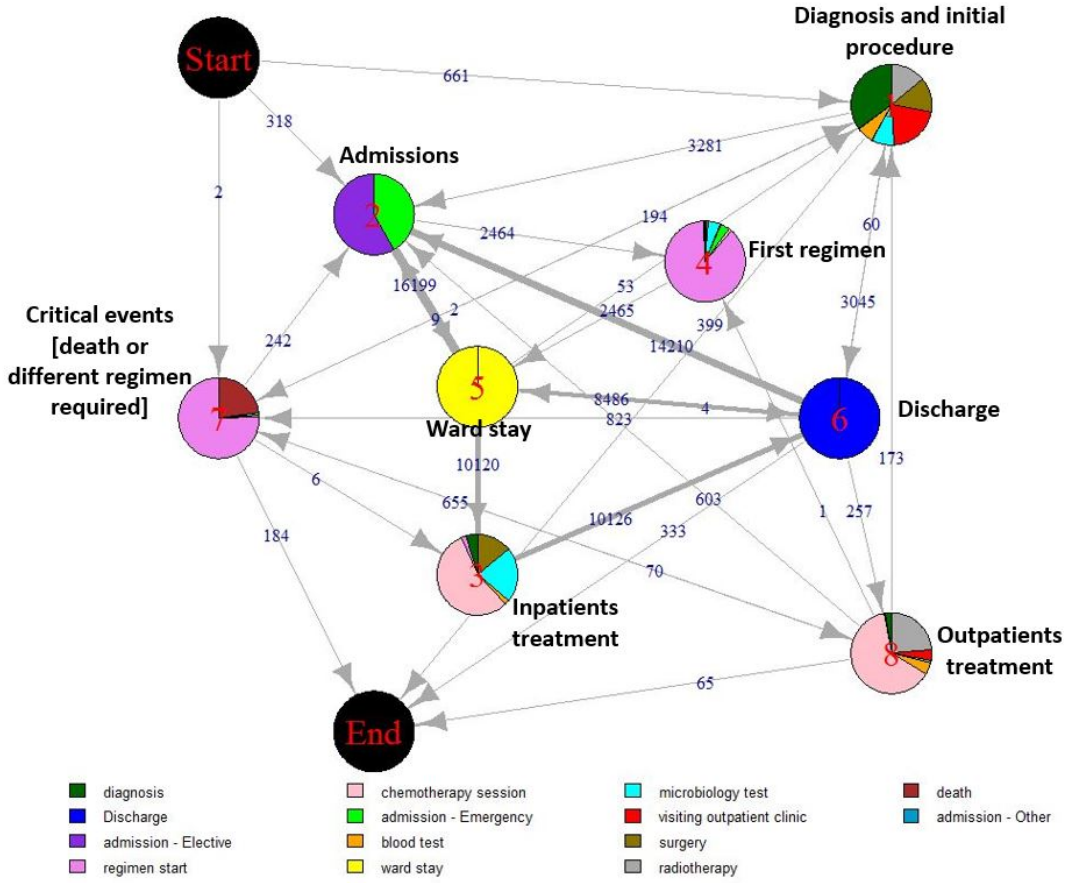


Figure 7.6: Initial labelled states of breast cancer process in case study 2

Discussion

The discussion of different outcomes here is derived by identifying possible critical states that can be used to provide healthcare precautions. Approaching such critical states may affect badly on patient pathway such as a pathway that may have death event or cancer reoccurring. The event of cancer reoccurring is based on the number of times that a patient is diagnosed with cancer after first diagnosis.

1- Death event:

In all treatment types there are 188 death cases (19%). Death event is observed in state 7 and most likely to occur after patient was discharged from the hospital as the model shows, however, this is subjective to how accurate the time of the death is recorded in the system.

For instance, a patient come to take chemotherapy and died in the hospital as shown in Figure 7.7. The exact time of death is not known since all events, in this case, are recorded with same admission day. Therefore, death could also happen in a ward then a patient is discharged.

In this case study, palliative treatment has the highest percentage of death among all other therapy types where 45% of the patients have died. In contrast, neoadjuvant treatment is the lowest percentage of death with only 11% as reported in Table 7.5.

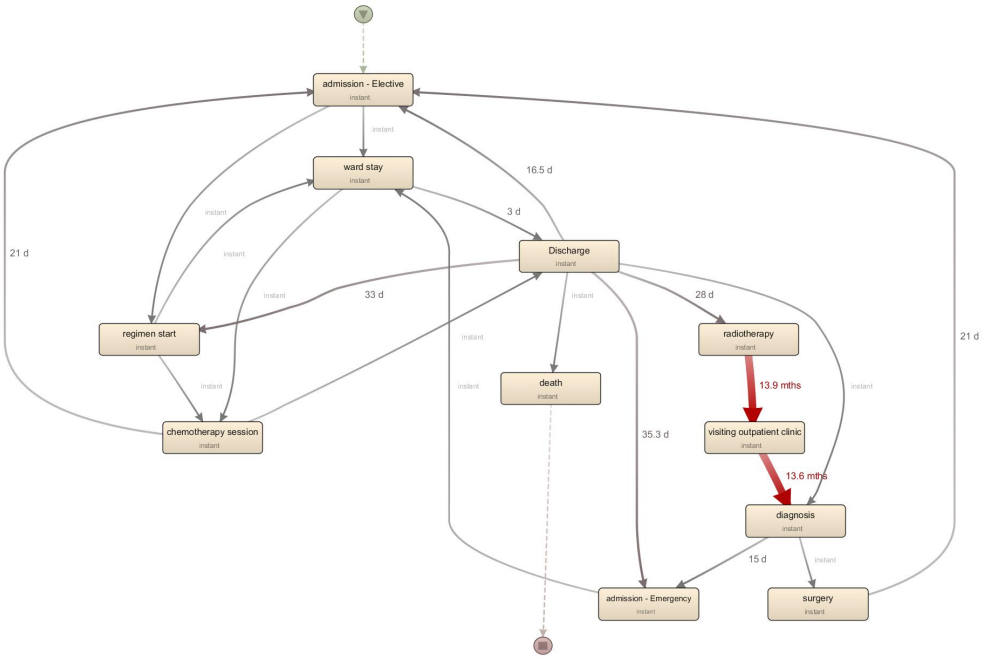


Figure 7.7: Example of events that are recorded with the same day of death (time is shown as instant)

2- Cancer reoccurring:

There are 626 (63%) of patients have a cancer diagnose event at least twice in this case study. The event of cancer reoccurring is based on the number of time that a patient was diagnosed with cancer after first diagnosis. Cancer reoccurring is discovered in state 3 or state 8.

The potential correlation between cancer reoccurring and death can be worthy to explore. Numbers in the following table reports that the majority of death, more than 80% of cases (n= 158), have happened for patients with reoccurring cancer in all treatment types; adjuvant, neoadjuvant and palliative. On the other hand, two cases have been diagnosed with cancer 8 times but they have survived.

Table 7.5: Bad outcomes in different treatment types of cancer therapy

Treatment type	Patients	Death	Cancer re-occurring	average of reoc-curring	Death in reoc-curring cancer patients	Temporal pat-tern of cancer reoccurring
Adjuvant treatment	495	77 (15%)	280(56%)	2	61 (79%)	32 months
Neoadjuvant treatment	322	36 (11%)	220(68%)	2	30 (83%)	31 months
Palliative treatment	164	75 (45%)	126(25%)	2	67 (40%)	12 months

Neoadjuvant treatment is the highest number of cancer reoccurring with 68% of patients whereas palliative treatment has the lowest number of cancer reoccurring. Despite the fact that palliative treatment has the lowest number of cancer reoccurring, this does not mean a good outcome. A possible reason for this low percentage is that due to the palliative patients have an advanced stage of cancer disease and the cancer is not expected to be cured. The number of average cancer reoccurring for all treatment types is the same. The time gap for cancer reoccurring is roughly the same for adjuvant and neoadjuvant which is a round 30 months while in palliative treatment cancer can reoccur after 12 months.
It should be noted that, the selected abstract model of case study 2 will be evaluated and discussed at the end of this chapter in section 7.5.

7.3 Strategy for selecting a cohort of patients using state-based abstraction model

The description of the above case studies, 1 and 2, is motivated by analysing the general process models. However, analysing process variations for similar group of patients is also important and this has been emphasised by the process mining reviews papers that were discussed in Chapter 2. The applicability of analysing the process variations comes at the next step of discovering the mainstream pattern. Hence, in this section we describe a strategy that helps in selecting a cohort of patients based on process similarity. This strategy provides the link between the abstraction method that is developed for modelling complex processes and other process mining approaches that were designed for relatively structured processes. The following top-down strategy is designed to guide the selection of a cohort of patients, who are presumably have similar process variants, using different similarity perspectives. We can select a similar cohort of patients based on a common state, based on a common event or based on a pattern of an event, as we will see in the case study 2. The strategy that is shown in Figure 7.8 should be followed after selecting the best model of the related case study.

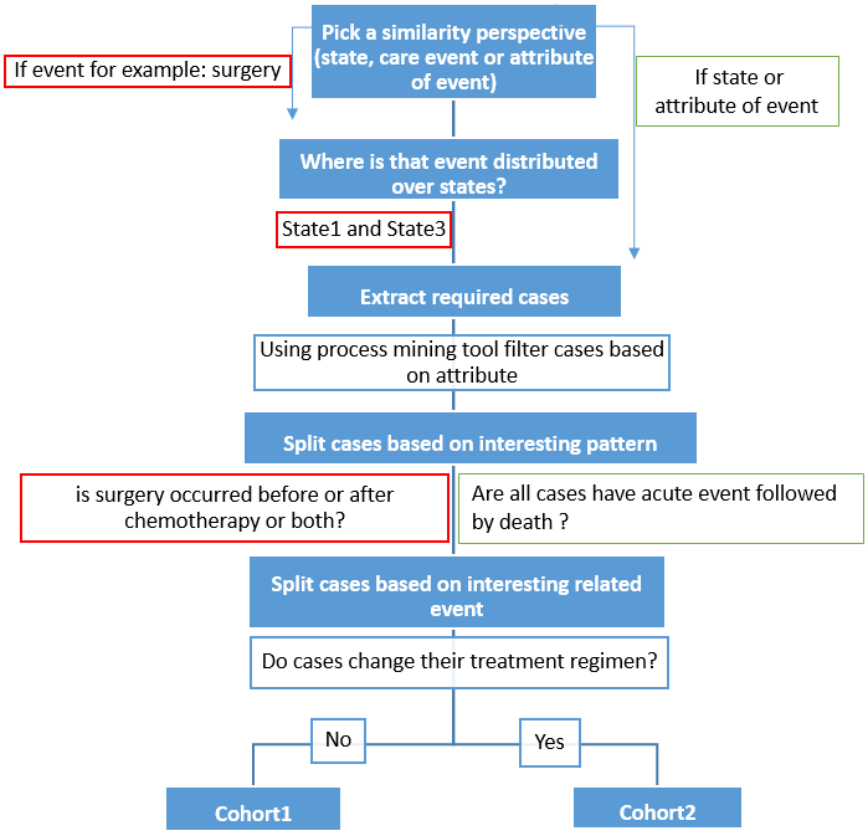


Figure 7.8: Strategy for patients selection in state based abstraction model

Once the perspective is chosen, the selection of the patients is narrowed down for a smaller cohorts of patients. Eventually, the process of the selected cohorts of patients are analysed using any available process mining algorithm.

7.3.1 Example of selection similar cohort of patients

In order to demonstrate the proposed strategy, case study 2 is used here as our running example, however, this strategy can be generally applied on any case study. Based on the best model that is illustrated in Figure 7.4, some significant events such as surgery and chemotherapy are distributed over multiple different states. Hence, we suggest that selecting patients based on a specific event that is related to a particular state can separate patients into possibly distinct groups.

The focus here is on analysing the variations of healthcare processes that are followed in different surgical therapies with considering the performance outcome. These treatments differ if a patient has a surgery or not and when the surgery has happened. For instance, patients have a surgery after chemotherapy(neoadjuvant), surgery before chemotherapy (adjuvant), a surgery that is proceeded and followed by a chemotherapy course and lastly, patients who have not

experienced surgical treatment in the healthcare process.

Group 1: This group represents the patients who have surgical treatment in the pattern of a surgery that is followed by a chemotherapy (adjuvant). The results show that, there are (n= 495, 50.4%) of patients who met this condition. Surgery event has happened at most once for all patients. There are 331 variants and the most followed process is: diagnosis → surgery → chemotherapy → radiotherapy → visiting outpatient clinic,
where it has been followed by 21 patients (4.5%). Other processes are mostly varied on the radiotherapy and blood or microbiology test events. By considering different contexts of the surgery event that are detected using the state abstracted model in Figure 7.4, we found that surgery can be occurred in two different states. There are 333 patients who have surgery in state 1 which might mean a surgery has happened in the past where it is recorded after diagnosis event instantly (both events are recorded with the same day) whereas 162 patients have surgery in state 3 which means surgery has been operated inside the hospital after staying in a ward.
Patients who have a surgery in state 1 have been diagnosed by cancer and the surgery event is recorded with the same time of diagnosis. The majority of patients (n= 235) were admitted multiple times to the hospital before they have been diagnosed. The patients had the surgery then after an average time of 2 months of the primary treatment, the patients had their chemotherapy. Also, 23 of them have radiotherapy in the healthcare process.
On the other hand, patients who have surgery in state 3 which means the surgery is operated inside the hospital ward. The majority of patients (n=114) started by diagnosis in state 1 then after 40 days have been admitted to the hospital electively and do the surgery. The chemotherapy sessions started after 46 days of the surgery.

Group 2: This group is for patients who have surgical treatment in the pattern of chemotherapy followed by surgery (neoadjuvant). We found that, there are (n= 322, 32.82%) of patients who met this condition. Surgery event has happened at most once for all patients.
There are 247 variants of the process and the most followed variant is: diagnosis → chemotherapy → radiotherapy → visiting outpatient clinic → surgery
Where it has been followed by 16 patients (4.8%). Other processes mostly varied on radiotherapy and blood or microbiology test events. By examining different contexts of surgery event, we found that surgery can be occurred in three different states. There are 199 patients who have surgery in state 1 which means surgery has happened in the past, 116 patients have surgery in state 3 which means surgery has been operated inside the hospital and only 7 patient have surgery in state 4 which has happened in the hospital after admission elective directly but ward stay event was not recorded before surgery event.
Patients who have surgery in state 1 have started from state 1 and have been diagnosed in the clinic. Then after 75 days they get chemotherapy. After the chemotherapy by average 22

months the patients had the surgery.
Patients who have surgery in state 3 have been diagnosed in the outpatient clinics (start from state 1). After 62 days they started chemotherapy and then do the surgery after 18 months as average. All cases have previous multiple elective and emergency admissions. Patients who have surgery in state 4 have a very different context where the surgery was recorded after admission elective directly. This sequence of events might be related to data quality issue since the patents need to be set in a ward before doing the surgery. The patients have been diagnosed and after 78 days have been given chemotherapy. Then after an average of 21 months, the patients have a surgery.

Group 3: Patients here have a surgery and they have the pattern of chemotherapy followed by surgery then another chemotherapy is given. The result shows that, there are (n= 156, 15.9%) who have met this selection.
There are 144 variants of the process and the most followed variant is: diagnosis → chemotherapy → surgery → chemotherapy → radiotherapy → visiting outpatient clinic → diagnosis
Where it has been followed by 8 patients (5.12%). Other processes are mostly varied on the radiotherapy and blood or microbiology test and if a patient has repeated diagnosis. Analysing the states related to surgery event, we found that the surgery can be occurred in three different states. There are 83 patients have surgery in state 1, 70 cases have surgery in state 3 and only 3 patients have surgery in state 4.
Patients who have surgery in state 1 are diagnosed in state 1 then after 61 days of diagnosis, chemotherapy is given. Then after 85 days patients had the surgery that is followed by a chemotherapy course after an average of 27 days.
Patients who have surgery in state 3 they were diagnosed in state 1. Then after 69 days chemotherapy is taken. After 78 days of chemotherapy, patients had the surgery which is followed by a chemotherapy treatment after 29 days as an average. Patients who have surgery in state 4 has a different context where surgery has occurred after admission emergency directly. 3 of patients in this states are diagnosed in state 1 then after 54 days of diagnosis, the chemotherapy is taken. Then after 20 days, the patients had the surgery which is followed by a chemotherapy course after an average of 11 days.

Group 4: Patients here do not have any surgery throughout their healthcare process. There are only (n= 8, 0.81%) who met this condition.
In this group, 6 of the patients took chemotherapy and radiotherapy, however, 2 of them have only chemotherapy. Totally, (50%, n=4) of them ended by death.
Although of the few number of patients in this cohort, there is no common variant of process among them.

Table 7.6 shows a summary of the main characteristics of groups of patients in case study 2.

Group 1 has the largest number of patients whereas group 4 has the smallest number of patients. Despite group 1 is the largest group, it has the lowest percentage of processes variation. In contrast to group 4 which has only 8 cases, but it is the most varied processes.
The main pattern of the surgical treatment is surgery followed by chemotherapy (adjuvant) in group 1, chemotherapy then surgery (neoadjuvant) in group 2 and chemotherapy followed by surgery then another chemotherapy course is given in group 3. Based on different contexts(states) for surgery event, the temporal average between surgery and chemotherapy is different in all groups. For group 1, the shortest time between surgery and chemotherapy is nearly a month and 2 months for the surgery that is operated inside the hospital(state3) and for history surgery (state1) respectively.
The longest time between surgery and chemotherapy is 5 months and this for the surgery that is operated after admission but no ward stay event is recorded(state4), which may be affected by data quality issue where this event is mistakenly not recorded. For group 2, the temporal average is roughly the same for the three different surgical contexts which is around 20 months. In group 3, the time between chemotherapy and surgery is longer than the time between surgery and chemotherapy where the time of the latter is less than a month. As overall, the adjuvant treatment of group 1 is the fastest pathway because it has the least time gap between surgery and chemotherapy than group 2 and group 3.

Table 7.6: Summary of the characteristics of groups of patients in case study 2

	group 1	**group 2**	**group 3**	**group 4**
number of patients	495	322	156	8
number of variants	331 (66%)	247 (76%)	144 (92%)	8 (100%)
surgical treatment pattern	surgery then chemotherapy	chemotherapy then surgery	chemotherapy, surgery then chemotherapy	no surgery
temporal average between surgery and chemotherapy in state 1	2 months	22 months	2.7 months then 27 days	-
temporal average between surgery and chemotherapy in state 3	1.4 months	18 months	2.5 months then 29 days	-
temporal average between surgery and chemotherapy in state 4	-	21 months	20 days then 11 days	-

The proposed strategy for patients selection has helped in identifying similar groups of patients based on common process characteristics. It should be noted that, although the discussed strategy is based on hidden state modelling, there are other techniques can be used as well for patients selection such as selecting patients based on their demographic data for instance, age or gender. However, the aim of the suggested strategy is to focus on events related to different contexts in the process.

7.4 Case study 3: Different regimens and treatment types of breast cancer patients with an acute event

In this case study, we would like to include further complexity of the healthcare processes. Our hypothesis, which is the same one that is adopted in case study 2, that is broaden the selection of patients and care events increases the healthcare process complexity. One way to do that is by extracting a larger group of patients which includes patients who have different chemotherapy regimens. For instance, extracting a mixture of breast cancer chemo-regimens with all treatment types which are adjuvant, neoadjuvant and palliative. In addition to expanding the care events that are used in case study 2 to include an acute event such as Neutropenia sepsis as well.

7.4.1 Extraction criteria

We have extracted the same inclusion criteria for previous case studies in addition to different chemotherapy regimens besides the 'EC90' that was used before. In the PPM data there are more than 600 combinations of regimen and treatment type that is used for breast cancer treatment. The tops three regimens which are; 'EC 90' , 'Tamoxifen breast' and 'Anastrozole' were selected with different treatment types such as adjuvant, neoadjuvant and palliative, see Figure 7.9.

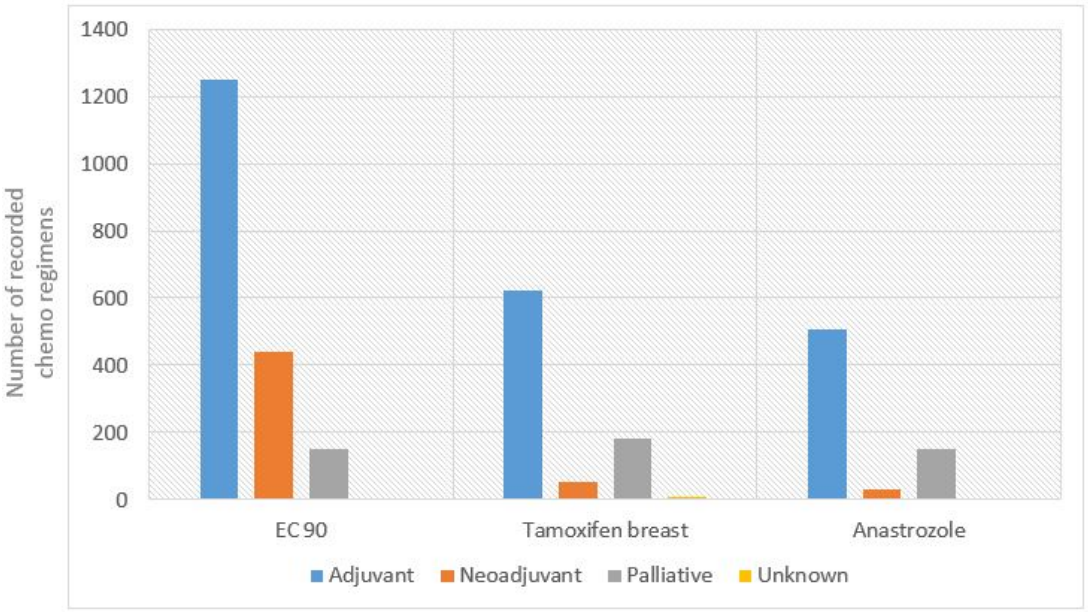


Figure 7.9: Top three regimens in the PPM data extract with different treatment types

Clearly, the regimen 'EC 90' is the most given chemotherapy regimen in PPM data extract for

breast cancer patients. Furthermore, adjuvant treatment type is the most frequent therapy in all extracted regimens.
In 'EC 90' neoadjuvant is the second treatment given, however, the other two regimens 'Tamoxifen breast' and 'Anastrozole' are mostly given for palliative therapy as the second treatment type after adjuvant therapy.
The extraction criteria selected 1520 patients with high variation percentage 99% of the process and high variable lengths processes that are ranging from a very short process with only 8 events to a very long process with 926 events as shown in Table 7.7.

Table 7.7: Log characteristics of different regimens and treatment types of breast cancer

total cases	distinct event	total events	variants	variation (%)	nulls	case length		
						min	avg	max
1520	15	115737	1505	99	1291786	8	76	926

The extracted event log has 15 event types. It should be noted that, in this case study we used all the tables that are available in the PPM data extract that was given to us except chemo-drugs table due to data quality issues. The reason for excluding this table is discussed in Chapter 6. Events names and frequency are displayed in Figure 7.10 below.

Activity	▲ Frequency
Discharge	26,736
ward stay	26,725
admission - Elective	23,892
chemotherapy session	17,420
regimen start	5,341
diagnosis	3,574
admission - Emergency	2,839
blood test	2,506
microbiology test	2,001
visiting outpatient clinic	1,518
surgery	1,495
radiotherapy	1,260
death	344
Neutropenia sepsis	81
admission - Other	5

Figure 7.10: Screenshot of case study 3 events and frequency

7.4.2 Models learning and decoding

Applying our method on this case study has resulted in populating 14 possible models with maximum 15 hidden states, which is the upper bound based on the number of distinct events, as shown in Table 7.8.

Table 7.8: Learning HMMs with different number of hidden states in case study 3

Model number	Number of states	Iterations =(MX)3000	Initial log	Final Log likelihood	BIC (Bayesian information criterion)
1	2	235	-309493.5	-207884	416129.4
2	3	226	-336767.1	-164145	328873
3	4	259	-321287.4	-147933.9	296695.5
4	5	426	-323283.5	-125076.7	251249.3
5	6	328	-308651.8	-125856.9	253101.3
6	7	230	-304885.1	-113322	228346.3
7	8	283	-326334.5	-110093.7	222227.8
8	9	470	-326085.1	-111439.6	225280.9
9	10	443	-312510.3	-107884.2	218555
10	11	1038	-311682.2	-107527.6	218249.7
11	12	985	-317009.6	-102026.9	207679.7
12	13	731	-321741.9	-104197.4	212475.4
13	**14**	**1054**	**-313520.2**	**-99997.62**	**204553.9**
14	15	2066	-319227.2	-104223.2	213506.4

The standard metric of selecting the best model of HMM, BIC, has selected a model with 14 states as the best model. However, this model suffers from the limitations that were discussed in Chapter 4, we believe that the general process model can be discovered using fewer number of hidden states. Therefore, all models are decoded using the Viterbi algorithm as a pre-step for models candidate optimisation. The results of our method are explained below.

7.4.3 Optimisation

The two different types of our proposed multi-objective function are performed to optimise models candidate space:

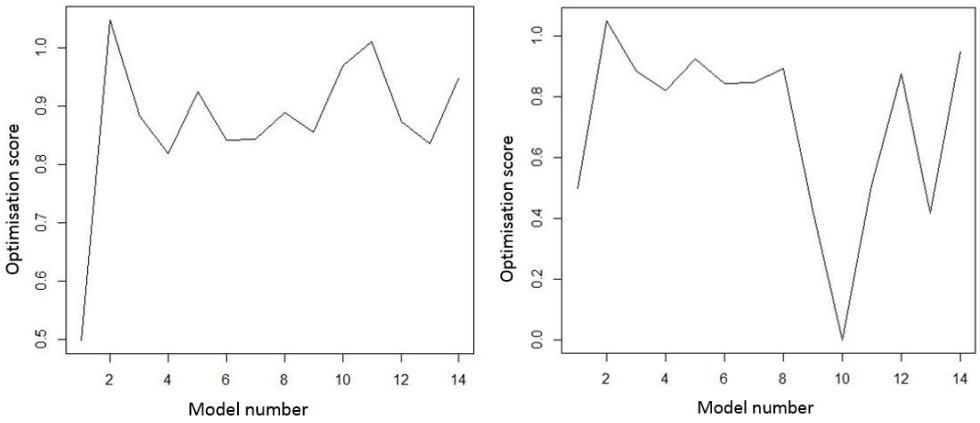
Strict and soft optimisation for models' candidate space

Using our method both types of optimisation have selected the same model as shown in Table 7.9. Model of 3 states is the best abstracted model based on our criteria. This model has the maximum score of the strict optimisation and soft optimisation which is 1.049.

Table 7.9: Criteria calculation in case study 3 (strict and soft optimisation)

Model	1	2	3	4	5	6	7	8	9	10	11	12	13	14
States	2s	**3s**	4s	5s	6s	7s	8s	9s	10s	11s	12s	13s	14s	15s
Linearity	0.685	**0.844**	0.743	0.624	0.623	0.577	0.582	0.593	0.595	0.638	0.636	0.563	0.574	0.638
Compactness	0.289	**0.097**	0.043	0.027	0.017	0.014	0.013	0.011	0.013	0.013	0.030	0.028	0.018	0.044
Cross sim.	0.727	**0.590**	0.581	0.415	0.312	0.305	0.313	0.291	0.327	0.301	0.247	0.238	0.303	0.307
Normalized importance	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0	0.5	1.0
Strict optimisation	0.499	1.049	0.883	0.819	0.925	0.842	0.844	0.890	0.428	0.000	0.505	0.874	0.418	0.948
Soft optimisation	0.499	**1.049**	0.883	0.819	0.925	0.842	0.844	0.890	0.856	0.969	1.011	0.874	0.836	0.948

Similarly to the second case study, due to the absence of unimportant states in most of the models, the optimisation scores have not changed. In contrast of the 11states model, with two unimportant states, where it has been penalised and its scores turned to zero, see Figure 7.11.



(a) Soft optimisation scores in case study 3 (b) Strict optimisation scores in case study 3

Figure 7.11: Multi-objective optimisations scores in case study 3

The percentage of state importance over all models is presented in Figure 7.10. We can see that, three models only have one single unimportant state and only one model has two unimportant states based on 50% threshold.

Table 7.10: State coverage and importance in case study 3

Model	State coverage percentage															# of unimportant states of two thresholds	
																50%	30%
2s	100	100														0	0
3s	100	100	100													0	0
4s	100	98	100	100												0	0
5s	99	100	98	100	100											0	0
6s	89	89	100	96	100	100										0	0
7s	100	100	79	100	89	99	79									0	0
8s	99	100	76	89	100	96	97	89								0	0
9s	85	79	99	91	98	96	93	99	98							0	0
10s	80	99	74	99	44	66	99	99	88	79						1	1
11s	100	97	86	79	79	99	41	15	88	99	99					2	2
12s	99	99	78	63	14	78	100	99	88	89	97	79				1	1
13s	100	99	99	79	79	79	60	100	64	76	99	99	99			0	0
14s	100	99	99	98	79	73	79	79	98	99	79	68	27	63		1	1
15s	10	99	98	99	99	98	92	83	78	85	79	88	85	99	95	0	0

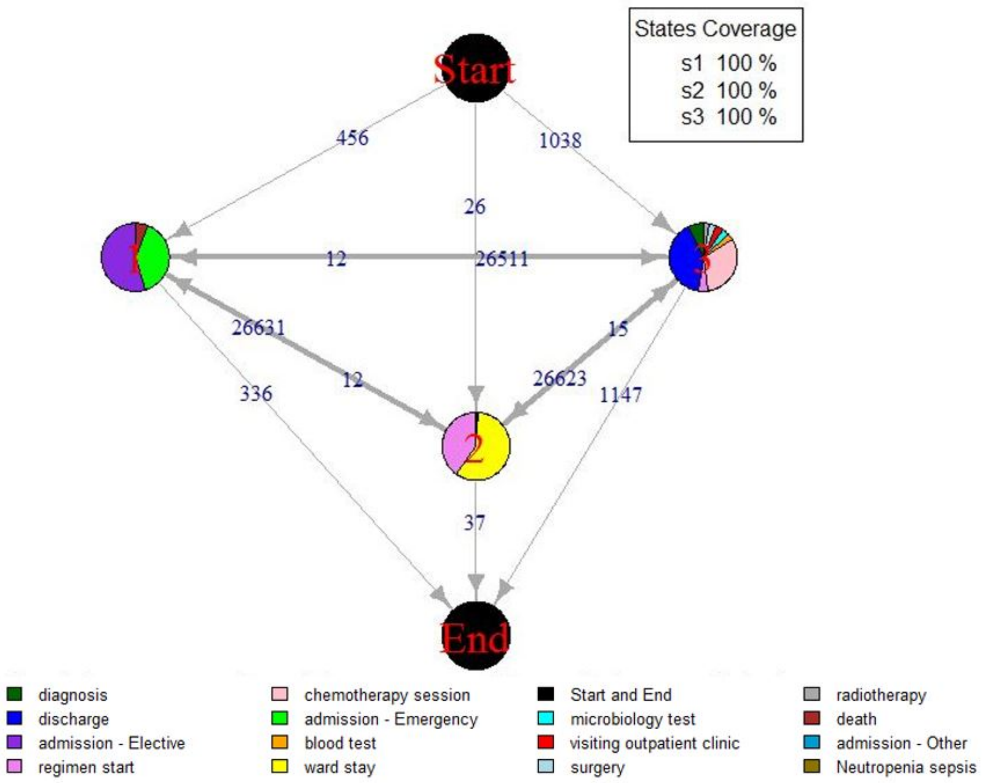


Figure 7.12: The best model of case study 3 selected by both strict and soft optimisation

Despite the high variety processes in this case study(variation = 99%), this model provides the general pattern that is followed for all different regimens patients. The majority of the patients (68%, n=1038) have started their process in state 3 where (77%, n=808) of patients have firstly diagnosed with cancer. (23%, n=230) of patients may have blood test or visiting outpatients clinic.

Then patients are allowed to be admitted to the hospital to get their treatment through state 1. After admission, patients move to a ward and start their chemotherapy regimen as shown in state 2. Inside the hospital ward patients can have several medical intervention based on their need in state 3 and lastly they will be discharged.

It could be clearly seen that, state 3 here represents multiple event types and this may not be the best modelling of the processes. Also, state 3 in this model is considered as a composite state, based on our classification of state type, because 80% of the state is occupied by more than two events as shown in Figure 7.13

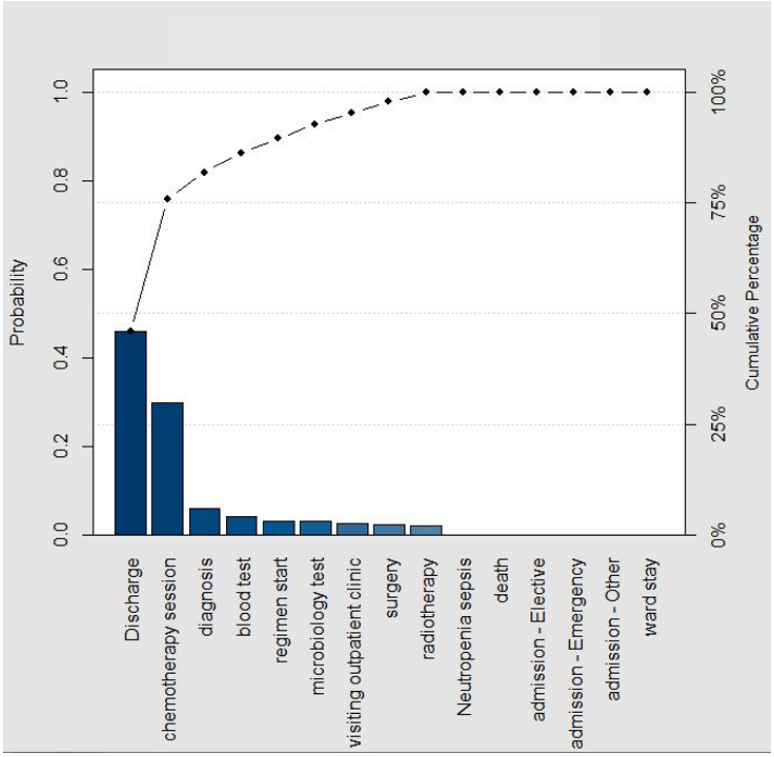


Figure 7.13: State 3 is a complex state in this model (since it is of the type composite with high process variations)

We believe that doing a further modelling for events in state 3 may help in getting better understanding of the process. In order to decide if a state needs a hierarchical modelling, the state type metric is not enough but the variation percentage inside the state should be measured as well. Hence, we found that the variation of processes in state 3 = 97%, consequently, a hierarchical modelling of process in state 3 is recommended.

7.4.4 Hierarchical modelling for complex state in case study 3

In Hierarchical Hidden Markov model (HHMM), every state can be considered as a single HMM and has its own probabilistic parameters [125]. The structure of Hierarchical Hidden Markov model (HHMM) consists of a root state, internal state, which is sometimes called abstract state, and production state, which is the leaf of the HHMM. In this research, we aim to apply a hierarchical modelling for any complex state that can be found in HMM. The structure of HHMM here is the same with the standard structure where we introduce three types of internal hidden states which are simple, composite and complex state besides the production state. The aim of the HHMM is to provide better process modelling that can be represented in different levels.

In case study 3, all events that are observed in state 3 are extracted as a sub log using ProM process mining tool. This sub log has the following characteristics as reported in Table 7.11.

Table 7.11: Sub-log characteristics of different regimens and treatment types of breast cancer

total cases	distinct event	total events	variants	variation (%)	nulls	case length		
						min	avg	max
1520	14	58535	1501	97%	668025	4	39	478

There are 14 distinct events included in state 3. This state covers 100% of cases and has high variable length processes.

Models learning and decoding

Applying our method to this event log has resulted in populating 13 possible models with 14 hidden states, which is the upper bound based on the number of distinct events, as shown in Table 7.12. BIC has selected a model with 14 states as the best model.

Table 7.12: Learning HMMs with different number of hidden states in the sub-log of case study 3

Model number	Number of states	Iterations =(MX)3000	Initial log	Final Log likelihood	BIC (Bayesian information criterion)
1	2	284	-153285.5	-92919.94	186158.2
2	3	417	-168773.1	-85869.19	172254.3
3	4	667	-161079.2	-85250.69	171236.9
4	5	536	-156003.6	-79684.09	160345.2
5	6	580	-153330.6	-77769.38	156779.2
6	7	1124	-155356.7	-77008.11	155542.1
7	8	1407	-150470.9	-75831.88	153497.0
8	9	905	-168378.5	-73586.61	149335.8
9	10	1567	-153328.5	-73737.72	149989.3
10	11	942	-159939.6	-73522.12	149931.3
11	12	987	-161585.5	-71124.90	145532.0
12	13	857	-157687.5	-70136.24	143971.9
13	**14**	**2864**	**-149930.9**	**-69405.84**	**142950.2**

Based on our method all models are decoded using the Viterbi algorithm to provide the input of the next optimisation stage. The results of our method are explained below.

Optimisation

The results of applying our proposed multi-objective functions to optimise models candidate space is discussed here.

1- Strict optimisation for models' candidate space

Selecting the best model with considering state importance is done using the strict type of our proposed multi-objective function (Equation 5.6). In this case, we would like to see the process model of most patients where each state should have no less than 50% of cases.

Table 7.13: Criteria calculation in the sub-log of case study 3 (strict optimisation)

Model	1	2	3	4	5	6	7	8	9	10	11	12	13
States	2s	3s	4s	**5s**	6s	7s	8s	9s	10s	11s	12s	13s	14s
Linearity	0.622	0.726	0.627	**0.769**	0.759	0.696	0.757	0.722	0.682	0.683	0.713	0.736	0.693
Compactness	0.610	0.588	0.417	**0.355**	0.302	0.225	0.202	0.175	0.180	0.212	0.245	0.251	0.239
Cross sim.	0.839	0.773	0.665	**0.551**	0.486	0.561	0.483	0.461	0.428	0.426	0.506	0.465	0.416
Normalized importance	1	1	1	0.857	0.714	0.857	0.714	0.571	0.429	0.571	0.571	0	0.429
Strict optimisation	0.101	0.386	0.381	**0.694**	0.629	0.616	0.664	0.512	0.363	0.476	0.455	0.000	0.364

Optimising the candidate space of 13 models using the strict optimisation has selected model of 5 states as the best model with the maximum value of 0.694, see Table 7.13.

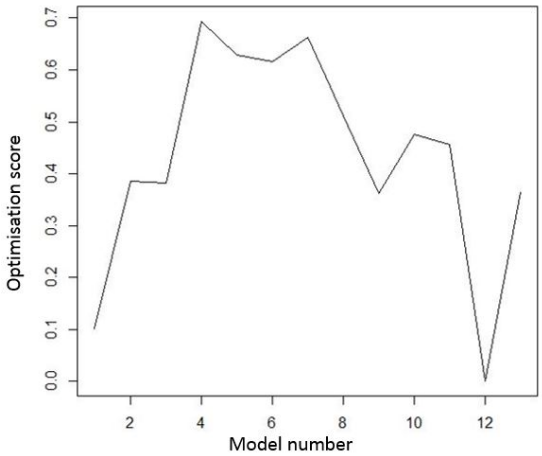


Figure 7.14: Strict optimisation scores in the sub log of case study 3

As can be seen in Figure 7.14, model of 5 states is the highest whereas the 12th model, which has 13 states, is the worst model since it has 7 unimportant states. State importance threshold that is used in our experiments is 50%, however, different thresholds can be tested as presented in Figure 7.14.

Table 7.14: State coverage and importance in case study 3 sub log

Model	State coverage percentage														# of unimportant states of two thresholds	
															50%	30%
2s	100	100													0	0
3s	72	72	100												0	0
4s	95	99	87	88											0	0
5s	92	75	21	95	88										1	1
6s	95	72	41	29	93	72									2	1
7s	54	97	71	79	71	31	95								1	0
8s	75	52	70	71	42	93	56	41							2	0
9s	18	65	88	93	68	20	44	65	69						3	2
10s	94	30	64	30	46	52	68	25	68	93					4	1
11s	84	30	56	66	93	72	85	77	28	64	40				3	1
12s	72	82	96	50	84	74	66	68	6	6	60	7			3	3
13s	95	38	81	68	68	7	7	18	71	39	49	27	73		7	4
14s	99	57	85	75	66	76	93	41	67	26	26	56	76	15	4	3

The percentage of state importance over all models is presented in Figure 7.14. In this case study, only three models consist of important states and these models have a few number of states. Other models have a range of unimportant states. These models have from 1 to 4 unimportant states in addition to one model, which is the worst, has 7 unimportant states based on 50% threshold.

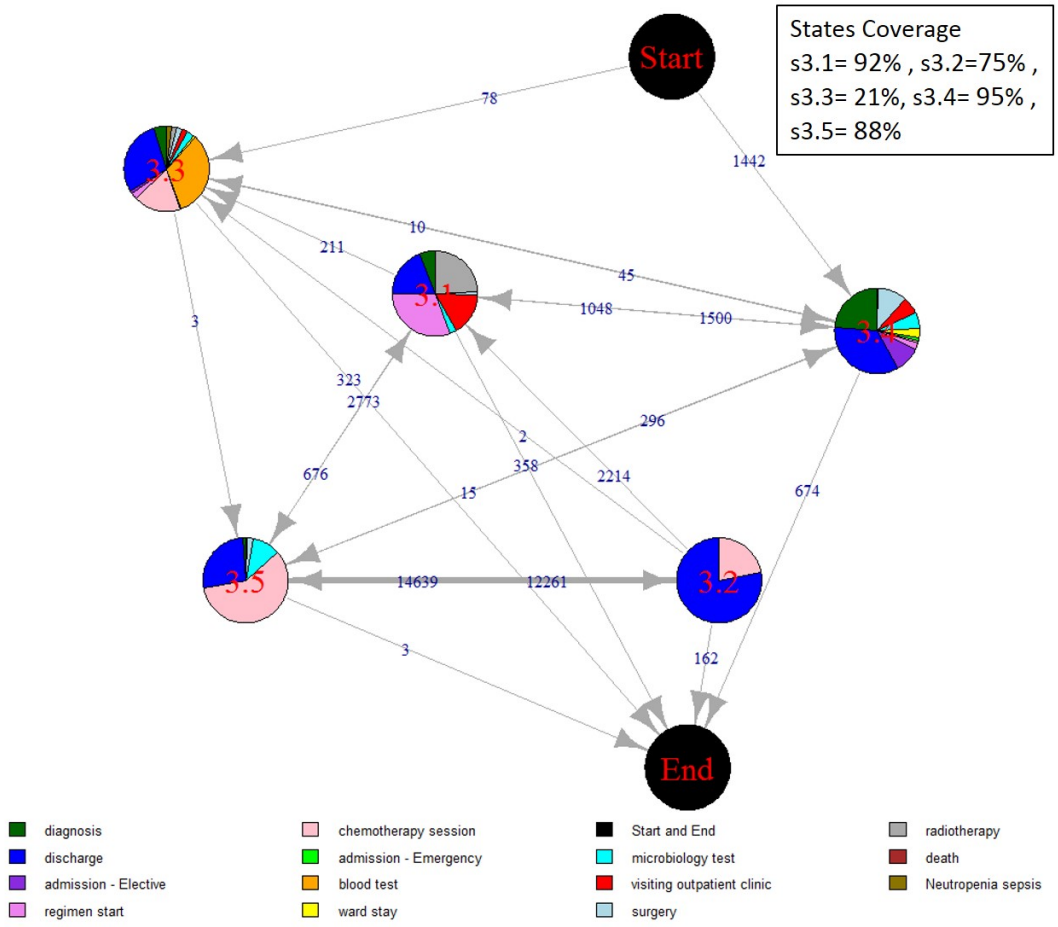


Figure 7.15: The best model of the sub-log of case study 3 selected by the strict optimisation

Strict optimisation has penalised the models with a high number of unimportant states. At the same time it aims to balance between other criteria. Although the selected model has one single unimportant state, it has a balance of other criteria; linearity, compactness and cross state similarity.

This model represents different kinds of medical interventions that may happen in patient visits. It should be noted that, each state in this model has the discharge event which indicates the end of the current visit. This implies a non-observed transition to level 1 model.

The majority of the patients start with diagnosis in state 3.4 then move to state 3.1 where the chemo-regimen start or to do a radiotherapy. After starting chemotherapy regimen, patients move to state 3.5 for the chemotherapy sessions then they can be discharged.

78 of patients may start with state 3.3 for a blood test in order to avoid an acute event such as Neutropenia, which has occurred 81 times for 36 patients.

2- Soft optimisation for models' candidate space

Selecting the best model with a flexibility toward state importance can be done using our soft optimisation in (Equation 5.5). Our method has optimised the space of candidate models for this sub log and then the criteria are calculated as displayed in Table 7.15. The best model is a model of 8 states where it has the maximum score of the optimisation function which is 0.929.

Table 7.15: Criteria calculation in the sub-log of case study 3 (soft optimisation)

Criteria	2s	3s	4s	5s	6s	7s	**8s**	9s	10s	11s	12s	13s	14s
Linearity	0.622	0.726	0.627	0.769	0.759	0.696	**0.757**	0.722	0.682	0.683	0.713	0.736	0.693
Compactness	0.610	0.588	0.417	0.355	0.302	0.225	**0.202**	0.175	0.180	0.212	0.245	0.251	0.239
Cross sim.	0.839	0.773	0.665	0.551	0.486	0.561	**0.483**	0.461	0.428	0.426	0.506	0.465	0.416
Soft optimisation	0.101	0.386	0.381	0.810	0.880	0.719	**0.929**	0.896	0.847	0.833	0.797	0.882	0.850

The worst model is the first model which is the model of 2 state where it has 0.101 score as can be seen in Figure 7.16.

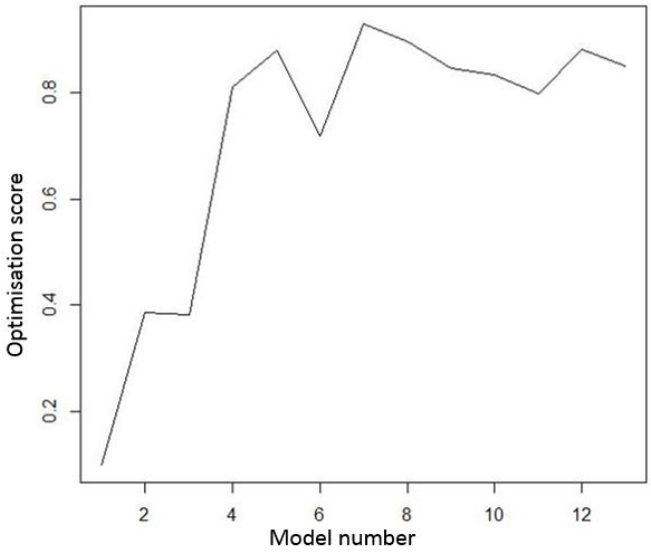


Figure 7.16: Soft optimisation scores in the sub log of case study 3

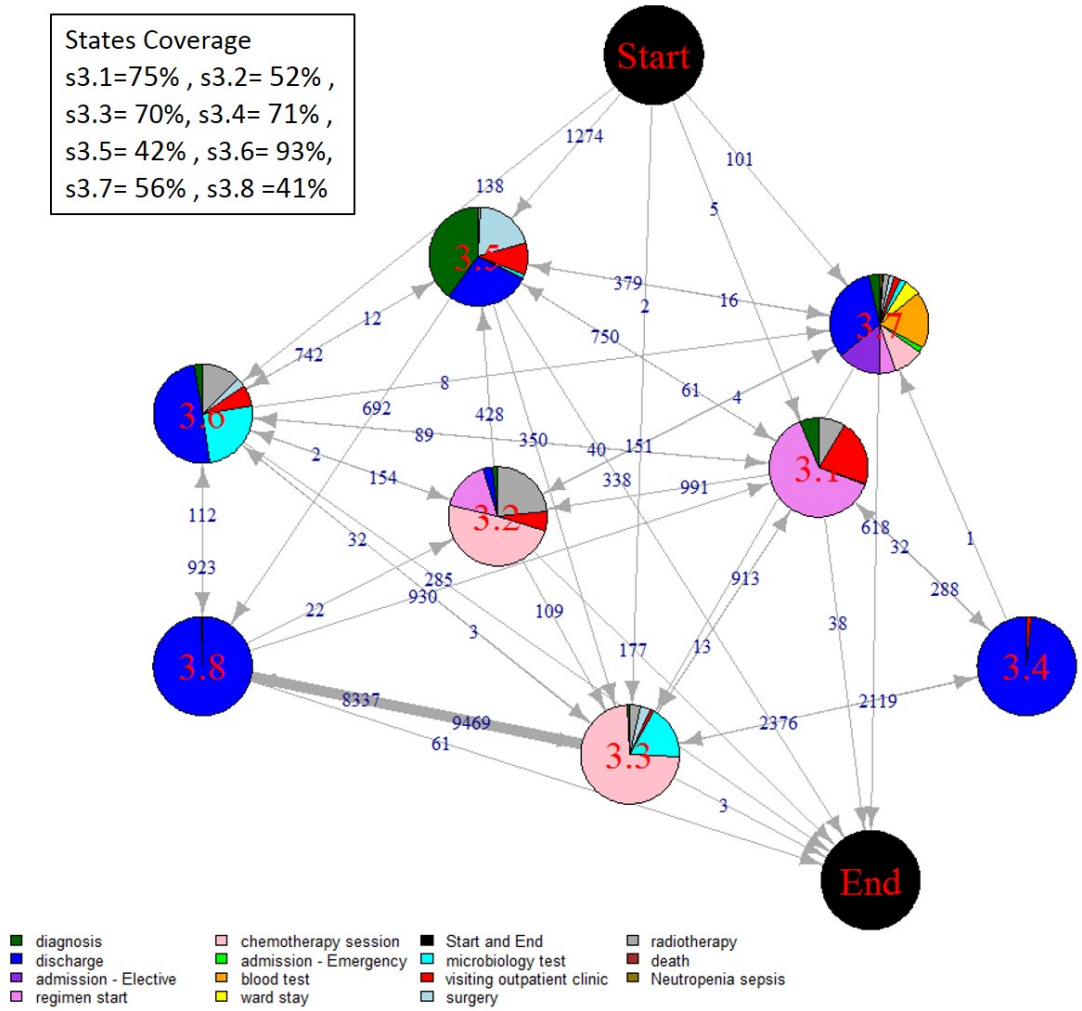


Figure 7.17: The best model of the sub-log of case study 3 selected by the soft optimisation

Soft optimisation selects the best model that provides a balance between all criteria without considering state importance. In this case study, model of 8 states is the best model. It represents the same general process of previous model, that is selected by strict optimisation, however it splits the discharge event into two distinct states 3.4 and 3.8. Also, this model distributes start process into fine-grained states such as state 3.6 which is derived from state 3.4 in the model selected by strict optimisation previously.

Hierarchical visualization of process model:
Focusing on the discovering of the mainstream process model has lead us to use the model of the strict optimisation, which is 5states model, for the hierarchical visualization for our process

model. By examining the distinct events and process of each state we can relabel the states initially based on the main events that contribute in forming the states, see Figure 7.18.

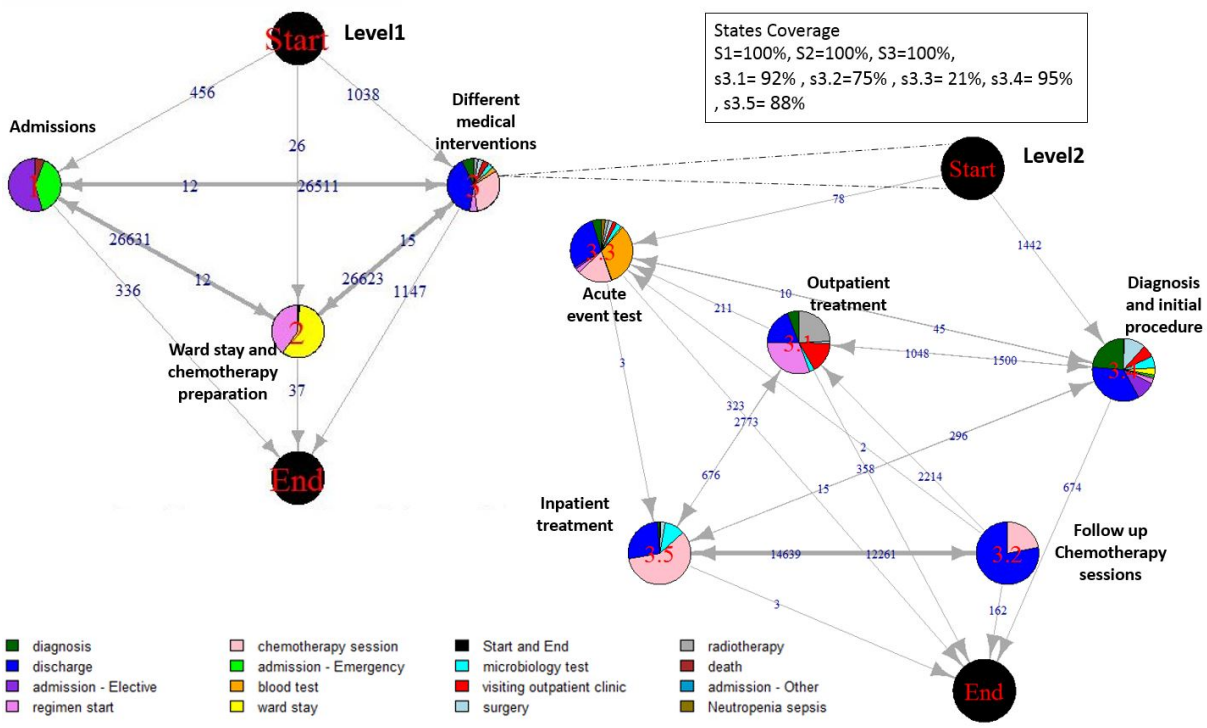


Figure 7.18: A two level abstracted model in case study 3 showing initially labelled states

7.4.5 Discussion

In this case study, we could discover the general pattern of care as presented in the model in Figure 7.18. This might help in identifying different outcomes. Bad outcome in this case study can be a pathway that may have death event or multiple change of chemotherapy regimens. Change of chemotherapy regimens in cancer treatment means the exposing to at least two different regimens whether the reason for the change is the change of the regimen components such as regimen drug or change of the doses. Multiple changes in chemotherapy regimens may imply not responsive treatment where the oncologist needs to adjust treatment plan [135]. In this discussion, the change of regimen in a patient treatment is identified if the regimen start event is observed at least twice in the process.

1- Death event:

Total death in this case study is 344 cases. As in case study 2, death event here is mostly happened after discharge however, 127 cases of death happened inside hospital ward. Interestingly, those cases have been admitted as emergency after getting a chemotherapy session. All death

events are observed in state 1.

2- Multiple change in chemotherapy regimens:

In this case study the majority of cases (74%, n=1137) patients have changed their chemo-regimen. As mentioned above, change in regimen does not necessarily mean moving to other types of chemotherapy regimens for instance, changing from EC90 to Anastrozole. It includes any change that may happen into the regimen either change of regimen drug or dose. Chemo-regimens might change in state 2, state 3.1 or state 3.3.

There might be a kind of correlation between the exposing to different regimens and death. Numbers in the following table reports that the majority of death, more than 90% of cases,(313/344*100) have happened for patients with multiple change of chemotherapy regimens.

On the other hand, one single case has been changing its regimen at most 28 times, but this patient has survived.

Table 7.16: Bad outcomes in different regimens of cancer therapy

Regimen type	patients	death	change regimen	average of regimen change	death in regimen changed patients	temporal pattern of changing regimen
EC90	772	143	590 (76%)	3	127 (21%)	86 days
Anastrozole	263	55	110 (41%)	2	43 (39%)	17 weeks
Tamoxifen breast	221	86	155 (70%)	4	83 (96%)	22 weeks
Mixed	264	60	264 (100%)	3	60 (22%)	14 weeks

7.5 Models evaluation for case study 2 and case study 3

This section provides an evaluation of the model selected in case study 2 and case study 3. The evaluation includes; model selection validation, evaluation using process model quality metrics and evaluation using domain expert.

7.5.1 Model selection validation for case study 2

In order to avoid repetition, the methods of identifying validation metrics, that were plotted in case study 1, are not presented in case study 2 and case study 3. However, the results are reported here. In case study 2, the proposed optimisation methods both strict and soft have selected the same model that has less states, 8 states, compared with BIC that has selected a model of 14 states as the best model. Figure 7.19 shows the best model for our optimisation with maximum score whereas the best model using BIC has the minimum value. A summary of the validation issues is discussed through three aspects which are the number of highly connected states, similar states and non-significant states.

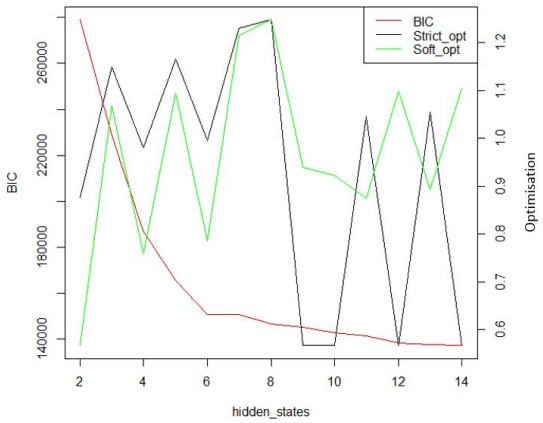


Figure 7.19: Best models of different metrics in case study 2

1- Connected components:

The strong connected component detection shows the possible connectivity between states of the 14states model, which is selected by BIC. The number of components in this model is 3 where each component(cluster) has at least 3 states. Likewise the model that is selected using our method, where there are 3 possible clusters. However, the number of states inside each cluster is 2 states only.

2- Similar state:

The model selected by BIC has three same-type similar states as follows:

1- Production states (Discharge) are shown in state 2 and 13.

2- Production state (Ward stay) are state 3, 5, 7 and 10.

3- Production states (Admission-Elective) are shown in state 11 and 14.

The model selected by our method has one same-type similar states which is:

1- Simple state (Regimen start) which are state 4 and state 7.

3- Unimportant states:

Adopting state importance percentage that is reported in Table 7.4, we use the threshold of 50% to determine state importance. The result showed that our model represents the significant states, however, BIC model has one unimportant state.

A summary of model selection validation metrics are presented in Table 7.17. Although the model that is selected by BIC and our methods have the same number of connected components, the number of inner states in the components is less in our model. The model that is selected using our method has a lower number of similar states and all the presented states are important.

Table 7.17: Validation metrics of case study 2

Issues	**Strict and soft optimisation**		**BIC**	
	found	count	found	count
strong connected components	yes	3	yes	3
similar states	yes	1	yes	3
unimportant states	no	-	yes	1

7.5.2 Model selection validation for case study 3

In case study 3, the proposed optimisation methods both of them have selected less number of states, the 3 state model, compared with BIC that has selected 14 state model as the best model. Figure 7.20 shows the best model for our optimisation with maximum score whereas the best model using BIC has the minimum value.

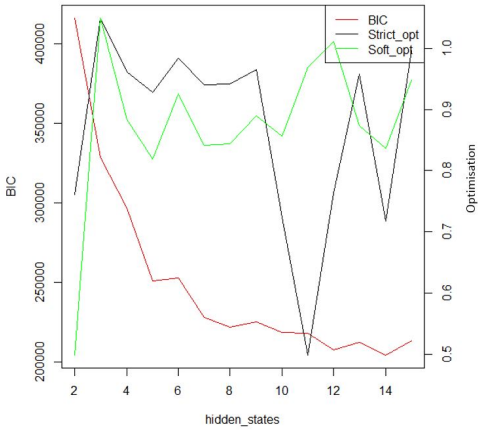


Figure 7.20: Best models of different metrics in case study 3

1- Connected components:

The number of connected states is high in the model selected by BIC, which is the model of 14 states. There are 3 possible higher abstraction that can be detected in this model.

In contrast to our model, the number of connected states is very few where there is only 1 possible cluster of states in this models that is selected by both soft and strict optimisation.

2- Similar state:

The model selected using BIC has three same-type similar states which are:

1- Production states (Ward stay) are shown in state 1, 4 and 11.

2- Production states (Discharge) are state 3 and 7.

3- Simple state (Regimen start) are state 2 and 6.

The model selected by our method has no states of similar states all states are constructed from different event types.

3-Unimportant state:

Using state importance percentage that is discussed Table 7.10, all states in our model were

significant whereas BIC model had a non-significant state. A summary of model selection validation metrics are reported in Table 7.18 where the model that is selected by our method is better than the model that is selected by BIC in all three issues.

Table 7.18: Validation metrics of case study 3

Issues	**Strict and soft optimisation**		**BIC**	
	found	count	found	count
strong connected components	yes	1	yes	3
similar states	no	-	yes	3
unimportant states	no	-	yes	1

7.5.3 Model selection validation for hierarchical case study 3

The proposed optimisation method both strict and soft have selected fewer number of states comparing with BIC. The strict optimisation selected the 5states model and the soft optimisation picked the 8states model. Unlike BIC that has selected a model of 14 states as the best model. Figure 7.21 shows the best model for our optimisation with maximum score of optimisation whereas the best model using BIC has the minimum value.

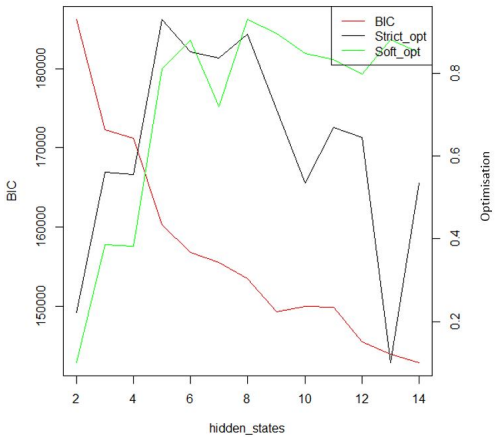


Figure 7.21: Best models of different metrics in the sub log of case study 3

1- Connected components:

The existence of highly connected states is abundantly observed in model of 14 states that is selected by BIC. There are 5 possible clusters can be detected in this model where each cluster has at least two states.

In contrast to our method, the number of connected states fewer where there are 2 and 1 cluster of states in models selected by soft and strict optimisation respectively.

2- Similar state:

The model selected by BIC has two same-type similar states which are:

1- Production states (Discharge) are shown in state 1, 5 and 13.

2- Simple states (Chemotherapy session) are shown in state 2, 9 and 10.
The model selected by the soft optimisation has one same-type similar states which is:
1- Production states (Discharge) are state 4 and state 5.
The model selected by the strict optimisation has no similar states where all states are constructed from different event types.

3- Unimportant states:

The state importance percentage that is presented in Table 7.14, showed that there are 4, 2 and 1 number of unimportant states that were found in BIC, soft and strict models respectively.
A summary of model selection validation metrics are presented in Table 7.19. The model that is selected by BIC has the highest number of connected components, similar states and unimportant states. In contrast to the model that is selected by our method, in particular the strict optimisation, has no similar states, only one unimportant state and one possible of higher abstraction.

Table 7.19: Validation metrics of the sub-log of case study 3

Issues	**Strict optimisation**		**Soft optimisation**		**BIC**	
	found	count	found	count	found	count
strong connected components	yes	1	yes	2	yes	5
similar states	no	-	yes	1	yes	2
unimportant states	yes	1	yes	2	yes	4

7.5.4 Models evaluation based on process mining metrics

As we did in case study 1, different process mining metrics are used to evaluate the abstracted models against models built using the original logs. Table 7.20 shows process evaluation metrics of the extracted event logs. These metrics concern about complexity, accuracy and performance. Table 7.21 shows the results of the discovered process models using IM and SM after using the strict and soft optimisation models. The three complexity metrics that are used for evaluation, have improved in the abstracted models.
On the other hand, the accuracy metrics show an overall increase in all case studies. However, the model of case study 3 that is generated by the IM has a less precision after abstraction. We believe this might be affected by the already known limitation of the IM, that is discussed in Chapter 2, where this algorithm tends to generate a flower under-fitting model with large event log. However, applying hierarchical abstraction for this case study has improved the precision as well. Case study 2 and the sub-log of case study 3 show improvement in all accuracy metrics including the generalization. In case study 3, the generalization remains the same before and after abstraction. Moreover, the performance of building the models has required less time after abstraction. Also, the process variation percentage has decreased in the abstracted models for all case studies.

Table 7.20: Process evaluation of case study 2 and case study 3 before abstraction

Logs	Variation	Discovery algorithm	Complexity			Accuracy				Execution time
			size	CFC	struct	fitness	precision	f-measure	generalization	
Case study 2	99%	IM	34	25	1	1	0.261	0.414	0.99	3081 ms
		SM	58	46	0.5	0.921	0.543	0.683	0.92	415 ms
Case study 3	99%	IM	30	22	1	1	0.31	0.48	0.99	1862 ms
		SM	53	41	0.81	0.89	0.54	0.67	0.89	436 ms
Sub-log of cs3	97%	IM	30	23	1	1	0.238	0.386	1	1049 ms
		SM	44	33	0.901	0.838	0.492	0.620	0.838	245 ms

Table 7.21: Process evaluation of case study 2 and case study 3 after abstraction using our optimisation

Logs	Variation	Abstraction	Discovery algorithm	Complexity			Accuracy				Execution time
				size	CFC	struct	fitness	precision	f-measure	generalization	
Case study 2	79%	Both	IM	30	22	1	1	0.458	0.628	0.99	2680 ms
			SM	35	26	1	0.965	0.677	0.796	0.97	378 ms
Case study 3	19%	Both	IM	17	11	1	1	0.26	0.41	0.99	1135 ms
			SM	13	8	1	0.98	0.72	0.83	0.98	422 ms
Sub-log of cs3	52%	Strict	IM	24	17	1	1	0.460	0.578	1	591 ms
			SM	16	10	1	0.866	0.796	0.826	0.860	118 ms
Sub-log of cs3	64%	Soft	IM	25	18	1	1	0.401	0.572	0.99	602 ms
			SM	23	16	1	0.449	0.913	0.602	0.731	238 ms

7.5.5 Models evaluation based on domain experts

Case study 2 and case study 3 were also demonstrated to the domain experts. After explaining the case study 1 in the meeting with the experts, the models of case study 2 and case study 3 were discussed in a brief way. The experts have confirmed that models were comprehensible and the processes were realistic based on their knowledge of the breast cancer healthcare process in the PPM. The domain experts have commented on the model in case study 2 that was selected by both strict and soft optimisation. The model showed that the radiotherapy (in state 1 and state 8) can be given without admission to the hospital. This was confirmed by the expert where they said radiotherapy is mostly taken in outpatients clinics where no admission is needed. Also, the domain experts have emphasized the usefulness of the hierarchical representation of the complex state in case study 3 in order to provide more understandable process models.

7.6 Discussion

Based on our experiments, the size of the event log, that is based on patients and events selection, has a strong influence on the proposed method for abstraction. On one hand, model with complex state type is found only in the largest size of our case studies, case study 3. Therefore, a hierarchical modelling is applied for that complex state in order to get better insights about the process as shown previously.

On the other hand, investigating the relation between the size of event log and the percentage

of processes variation has lead us to an interesting healthcare finding. From the experiments of our case studies we found that, breast cancer patients in Leeds cancer centre follow the same general process of healthcare regardless of the chemotherapy drug that is used. For more explanation, the scope of patients selection in case study 1 was focused on patients who had one single chemotherapy regimen (EC-90 drug) that was given as (adjuvant) treatment type. While in case study 2, we aimed to get more complex processes by including different treatment types such as adjuvant, neoadjuvant and palliative breast cancer treatments. This has indeed increased the complexity of the healthcare processes, due to the increase of the number of events and number of patients. Also, the processes variation, that is calculated in formula (3.1), has increased likewise. However, including different chemotherapy regimens such as Tamoxifen breast and Anastrozole, as we did in case study 3, did not increase processes variation but the complexity has increased.

From healthcare point of view, this may reveal an unknown property for healthcare process variation where it suggested that, the variation keeps increasing until it reaches a point that will never increase after it. This property is illustrated in Figure 7.22. The figure shows the relation between the number of cases(patients), number of events and the percentage of process variation. The variation has increased in case study 2 and reached 99% compared to case study 1 which was 86%, however in case study 3 the percentage of variation did not change. Considering a different formula for calculating process variation percentage has resulted in the same conclusion. We have calculated the percentage of process variation by considering the number of process variants that represented 80% of cases only, to get more realistic conception of the process variability. The percentage of process variation in the three case studies were; 66%, 79% and 79% for case study1, 2 and 3 respectively.

This may also be an indication to process standardisation between different cancer treatment regimens. All regimens follow the same healthcare general processes but the process differ according to the treatment type of cancer therapy whether adjuvant, neoadjuvant or palliative. It should be noted that, there might be some differences of the processes of different chemoregimens at low level details. However, these details are not captured in the abstract models primarily due to the event selection scope in the extraction stage.

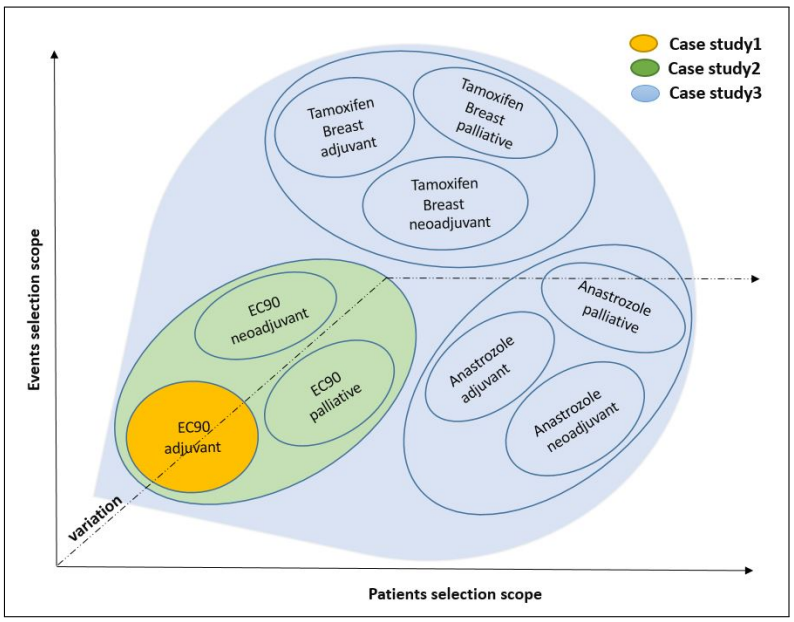


Figure 7.22: A conceptual illustration of the discovered property of process variations in the three case studies

From process mining point of view, the unvarying process variation percentage in case study 3 suggests reconsidering our hypothesis of increasing the complexity, which was based on the selection of patients and care events. In our experiments we found that, expanding the scope of care events extraction does not necessarily increase the process variations. This is mainly depend on the nature of the included events. For example, the included event in case study 3 was the acute event 'neutropenia sepsis' which was observed only 81 times for 36 patients, which is a very small number of the total sample of 1520. Hence, due to the limited occurring of this event in terms of the number of neutropenic patients, the process variations have not affected. Paying more attention to the affect of care events with regard to process variation can offer a new way for healthcare events extraction. In other words, in order to control processes variability, a gradual incrementing extraction of healthcare events is advised which can help in early identification of care events that cause process variability.

Furthermore, the size of the event log impacts significantly on the computational processing time of our method. The time needed for the stages of learning and decoding increases by the the increase of event log size. Figure 7.23 shows the the required time for learning HMMs and decoding of five event logs of different number of cases. It can be seen that, the event log with 296 cases has the least time of processing which is 30 hours due to its relatively small sample size. In contrast, event log with number of cases = 1520 required around 120 hours of processing. The approximate processing times depends on machine specifications where we have used a desktop computer with the following specifications; Intel Core I7 processor with 16 GB memory. Thus, using high performance machine would reduce the time of our abstraction

method.

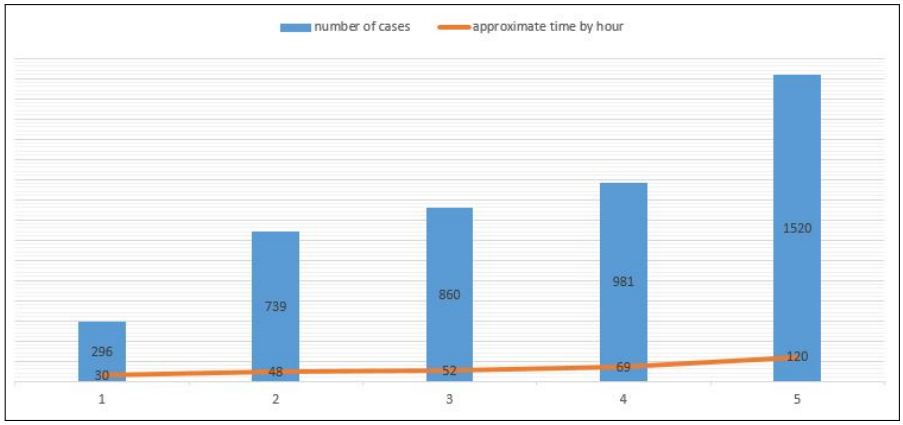


Figure 7.23: Computational processing time required for learning and decoding stages for event logs of different sample sizes

7.7 Conclusion

The two experiments in this chapter aimed to provide further complexity and test the capability of our proposed method of finding the main care pattern and reduce complexity. We applied our method on both case study 2 and case study 3. The candidate HMM models, for each case study, were optimised using the strict and soft optimisation methods. The models that were selected have successfully discover the main care pattern based on domain expert evaluation. Also, the resulted models have reduced the complexity by measuring different process models complexity metrics. In addition to that, the abstracted models have improved the models accuracy as well.

Finding a good or bad process outcome in our discussion is dependable on the presence or absence of an interesting event, such as interesting sequence of events, for instance, regimen start that is followed by another regimen start which means the change of the treatment plan. Another possible process outcome may be suggested is the use of the temporal pattern of an event as a constraint for identifying good or bad outcome. For example, if cancer reoccurring event is occurred after 1 year of the first diagnosis.

Chapter 8

Conclusion

8.1 Overview

This is the final chapter of the thesis and outlines the main challenges and contributions of the research. It contains a summary of each chapter followed by a discussion and concludes by explaining the limitations of this research and suggestions for possible future work. In summary, this research has successfully achieved the aims and objectives of this thesis. Using our proposed method we have reduced the complexity of healthcare process models and discovered the general process models of different complex healthcare processes. This was achieved without involving the domain experts in the abstraction stage. The large amount of process variation initially perceived when using process mining for healthcare processes may lead one to assume there is no general pattern of care. However the abstracted models presented within this thesis have demonstrated that the main care pattern is often hidden and can be discovered within the complexities of the unprocessed data. The method has been applied successfully to discover the healthcare process of breast cancer patients in Leeds Cancer Centre. The results of the process model evaluation have shown that as the complexity of the process models decreases the true care patterns are more easily revealed. Finally this research has been evaluated by domain experts who have confirmed the correctness and improved understandability of the discovered process models.

8.2 Summary of the challenges addressed in this thesis

This thesis has addressed five different challenges concerned with applying process mining to healthcare data.

The first and most important challenge is the complexity of healthcare process models as has been discussed in Chapter 1 and 2. Healthcare processes are highly complex due to the different choices that can be taken in order to meet the variety of patients needs. For this challenge, we

aimed to answer two simple questions:
1. 'Does a main care pattern exist for a group of patients with a similar diagnosis?'
2. 'If it is thought a main pattern of care does exist, can it be found using existing process mining methods?'
Analysing the different reasons for such complexity apart from the medical reasons, has helped us in knowing what kind of components may contribute in simplifying the complexity of healthcare process models such as the number of events and the links between these events.

The second challenge is related to the involvement of domain experts during the abstraction stage. In the process mining literature we have explored the methods that are suggested to cope with complex and highly unstructured processes. We established that all of these methods have relied on the concept of process abstraction, however they are all heavily dependent on the inclusion of domain experts during the abstraction stage. The involving of domain experts is costly in terms of time, money and the arrangement efforts that are needed for organizing regular meetings for all people who participate in the process discovery research. In order to meet this challenge, we have used an abstraction method that is based on machine learning technique in particular Hidden Markov Model (HMM) and the algorithms Expectation-Maximization (EM) and the Viterbi decoder. Our initial work was published in [3] to show the potential of using HMM in process modelling.

The third challenge is the extraction of care events from electronic health records (EHR). As mentioned in Chapter 3, most current electronic healthcare systems are not process-aware and care events are not recorded in a specific event log, instead they are often distributed over different database tables. In this thesis, two different EHR datasets were used; MIMIC-III and PPM. The MIMIC-III dataset is from an American intensive care hospital, while the PPM is from a British secondary care healthcare system, thus, represent different healthcare contexts. For both of these datasets a number of steps have been taken to create a large event log. The event logs contain all care events available in the system and required further next steps for extracting a specific group of patients.

The fourth challenge is knowing which model is best for complex unstructured processes. Defining the best model is subjective to different points of view and could have a clear definition if it is provided by domain experts. The best choice may be the most precise model or the model with the best replay fitness. However, metrics such as fitness and precision have sometimes be worthless when dealing with complex processes due to the discovery of either spaghetti-like model, that results in over-fitting and poor understandability, or flower model, that results in a highly imprecise under-fitting model.
As we aimed not to include domain expert in the process discovery phase, we adopted cluster validation metrics for our abstraction approach. The aim was to find states in the HMM

compact enough where the processes inside a state are similar. Also, it is preferred that the processes between states are highly dissimilar. Another criteria is the linearity of the model, which ensured a clear flow between states(block of events). We suggested a criteria which helps to select the best abstracted model based on state importance. State importance discovers the model that has total significant states that have a high coverage with regard to the number of related cases.

The fifth and final challenge has highlighted the role of model visualization. Using abstraction for addressing the complexity has imposed a further challenge regarding model visualization. Current techniques for process model visualization use the name of the event type for node labelling, however using abstraction, the single node may be a discovered state with a number of event types. Considering the correct labelling for these abstracted states requires the use of domain knowledge. We have therefore developed a new transparent (events can be seen through the pie plot states) abstracted model which is inspired by the current visualization of HMM in (SeqHMM) R package to be more appropriate for visualizing the abstracted process model without the need for node labelling.

8.3 Summary of the contributions of this research

This research has resulted in seven contributions to knowledge. The first contribution was using a new medical database, MIMIC-III, for process mining research. The MIMIC-III database has been publicly available since 2016, however, according to [23] there are 134 publications mostly describing data mining and machine learning approaches for medical research. Therefore, to the best of our knowledge, our published work in [20] is the first research that uses MIMIC-III for process mining purposes. This healthcare resource has provided the first healthcare event log written using the English language. The only healthcare event logs available on-line have been extracted from a Dutch Academic Hospital with the events recorded using the Dutch. It has been provided by the Business Process Management (BPM)[1] for process mining annual challenges. To ease the work for other process mining researchers, the steps to create an event log from MIMIC-III and extraction of the log for a specific group of patients is explained in our published work [20].

For our second contribution we developed a novel pre-processing method targeting the events that have a periodic presence in the process. The aim was to improve capturing the temporal pattern of such events and reduce the complexity of the process. For example, the event of measuring blood pressure should happen every 45 minutes on average in an intensive care unit. Therefore if these events reoccur in less than 45 minutes they are hidden. This method has

[1] https://www.win.tue.nl/bpi/doku.php?id=2011:challenge

improved the visualization of temporal patterns with repeating events and has decreased the variation of this kind of events as discussed in Chapter 3.

The third contribution is the development of an improved approach to using Hidden Markov model for process abstraction. The motivation for this was to better manage the limitations of the Bayesian Information Criteria (BIC) metric in model selection of high dimensional sparse data. We have demonstrated three practical issues that can be found in models selected by BIC as best models. These issues are the existence of higher level of abstraction between states that are strongly connected to one another, the issue of similar states and the issue of unimportant states based on our definition where these kind of states do not have high case coverage. The improved approach required the use of a combination of new criteria that may help in selecting the best desirable process model. We have investigated different experimental results for models trained with toy and real processes in order to analyse the characteristics of the more desirable process models. This resulted in four important criteria which are linearity, state compactness, cross state similarity and state importance. The properties of the proposed criteria were explored and helped in designing the multi-objective function that combined the criteria and in choosing the appropriate weights for each criteria. The multi-objective function is used for optimising the space of the candidates HMM. We proposed two types of optimisations; strict optimisation and soft optimisation. The former tried to select a model with balance of all criteria and aimed to select the model that provided the general flow of the healthcare processes whereas the latter is relaxed regarding the state importance where it can select a model with states that may have low case coverage.

The fourth contribution is the proposed strategy for selecting similar patients based on the state abstraction model. Detecting similarities between patients based on their healthcare processes is not achievable with complex spaghetti-like models, therefore, this strategy for finding similar groups of patients is one of the main contributions of this thesis. The suggested strategy is applicable after finding the best abstracted model that is resulted from the optimisation. In this top-down strategy we can explore different perspectives for process similarity which are; similarity based on common state, common event or common event with a particular attribute, such as the event of chemo-regimens where the chemo-regimen label is EC90 or Tamoxifen breast. In Chapter 7, case study 2 was used to demonstrate an example of cohort selection strategy based on common event, surgery, that was observed in different states.

The fifth contribution is the implementation of a new visualization tool for state abstraction model that is inspired from the package 'SeqHMM' within the R platform. This tool aims to support different enhancements which include; providing clear start and end nodes for the process, using frequency of transition rather than probability and providing better model layout instead of representing all states in one single row (either left-to-right or right-to-left layout).

For the sixth contribution we have identified three new types of hidden states which are simple, composite and complex. There are two possible types of hidden states that have been discussed in literature [125]. These states are production and abstract node. Our new types of hidden states can be used as a classification for the abstract states which may help in providing an indication for the need of further hierarchical modelling. These types take into consideration the percentage of process variation inside a state in addition to how many event types occupy 80% of the state.

The seventh and final contribution provides the implementation of our method which includes two types of soft and strict optimisations, criteria calculations and the visualization method as an open source package in the R language. The package documentation is supplemented in appendix D.

8.4 Chapters summary and overall discussion

In Chapter 1, we explained the motivations of this research by outlining the main challenges of using process mining in healthcare. Research problem statement was discussed in addition to the proposed research method. The main contributions of this research are described besides the work that were published throughout this research.

In Chapter 2, we have explained the nature of healthcare processes and the implications for process mining in healthcare. A discussion of applying process mining in healthcare is now provided. The complexity, which is the most challenging part of modelling healthcare processes is discussed in detail with the causes of such complexity and how it can be measured. We have adopted the complexity definition from [1] and included the fourth component of complexity which is the type of interrelatedness. From a process mining view, the type of connection between events can be represented as process constructors, which can increase the complexity of the process model.
The chapter explored a background of general process mining algorithms and algorithms that aim to address complexity. Methods that tackle complexity can be categorised as a supervised method, abstraction with domain expert, or a pattern based method. The available methods have helped in identifying the second challenge in our research which is the involvement of domain experts in the abstraction. The algorithms that are designed to cope with complexity mostly have four properties as suggested in [56] which are aggregation, abstraction, emphasise and customisation. We have taken these properties into account during the development of our method. Chapter 2 has provided a background of HMM and its related algorithms which are the EM and the Viterbi and a brief review and discussion of the use of HMM in process mining. This has shown that clustering based methods mainly aimed to split groups of patients into

different clusters before process discovery to solve the complexity, hence, cannot discover the general process model for the whole extracted log.

In Chapter 3, we investigated the use of a new healthcare resource, MIMIC-III, for modelling patient processes. Before this thesis the only healthcare data that was available on-line was the BPI challenge data which was recorded in Dutch, not English. MIMIC-III provided a tertiary care data source from a hospital and for inpatients who required special expertise and equipment. The steps for producing the event log for a specific cohort are explained in detail. This chapter demonstrates the third challenge of our thesis which is the extraction of care processes from non-process aware systems. We have explored different approaches for event log pre-processing which include aggregation and temporal methods. The aggregation approach dramatically reduced the number of events. The temporal approach related to the interval aspect of an event was used as an ad-hoc approach for MIMIC-III since it had a high number of repeated events such as chart event in the intensive care units. Although the number of events significantly reduced after the pre-processing steps in this chapter, the variation of the process was still high. Therefore, some process mining techniques such as fuzzy miner and local process mining were tested and they presented a number of limitations that prevent generating understandable models.

In Chapter 4, we described how we applied the method of HMMs for state abstraction modelling. Several model selection metrics are discussed for example, AIC, BIC, ICL and cross validation likelihood which showed BIC was the best metric for selecting the most right model, however, some concerns were raised in using BIC with high dimensional data. In order to investigate these concerns, we have conducted some empirical experiments using toy data and colorectal cancer real data that was extracted from MIMIC-III. The data used were varied in size and sparsity. Three main issues were empirically detected on models selected by BIC as best models. These issues are; the potential for higher abstraction, the existence of multiple similar states and unimportant states. The issue of similar states lead us to identify the new classification of the hidden states types that are mentioned in the contributions. Interestingly, using the BIC metric has selected a good model with a small event log and this helped us to characterise what a good model can be for larger scale processes.

In Chapter 5, we introduced the idea of multi-objective optimisation. The question of what is the best model was not easy to answer, however, we suggest four possible criteria which may help when selecting the best state abstracted healthcare model. These criteria were linearity, state compactness, cross state similarity and state importance. We have demonstrated the rationale for selecting these criteria and how they can be calculated. A simple calculation using the transition matrix is used to compute model linearity. For the state compactness calculation, a context-aware score is used where the weight of events transposing was reduced to half

because transposition between healthcare events is expected based on our data observation. In cross state similarity we adopted the general process similarity metric that has used the common node and common edges for measuring the similarity. Regarding the final criteria, the importance threshold can be set by the user, however we selected 50% as state importance percentage aiming at discovering the general model where states represent at least half of the patients. It should be noted that, the idea of this multi-objective function is built upon the idea of general clustering validation in addition to the linearity of the model and considering case coverage that is represented by state significance.
Criteria properties and the relations between the criteria have helped in weighting the parameters of the multi-objective function. In this chapter, criteria exploring and weights tuning have been achieved using toy data and a real event log data extracted from MIMIC-III. In order to provide some flexibility in model selection, two types of optimisation are suggested which are soft and strict. Soft optimisation used unconstrained criteria, which are linearity, compactness and cross state similarity, while strict optimisation included the constrained criteria as well which is state importance. Lastly, a robust method is developed to include the suggested optimisation as a previous step before model selection.

In Chapter 6, we moved to a different healthcare data source, PPM, with the aim of testing our method through a number of case studies then evaluating the models with a domain expert. Our PPM dataset contained de-identified electronic health records for cancer patients in Leeds cancer centre in the UK. We extracted the first case study which contained event data of chemotherapy cycles of breast cancer patients. There were some challenges in preparing the PPM data particularly for process mining. Our abstraction method successfully discovered the general process model for this case study. In this chapter we discussed how the definition of a good and bad outcome is depend on the selection made by the process analyst and the concept of quality that might be important for stakeholders. For example, process model outcomes can rely on the presence or absence of an interesting event or pattern. An interesting finding is that our approach has differentiated between the different contexts of the main steps in the healthcare process. For instance, the HMM that was selected using soft optimisation identified two different states for chemotherapy cycles and this was confirmed by the domain expert in the evaluation session. Models were validated against the issues that were key drivers for our optimisation and evaluated also using process mining quality metrics that evaluate a models complexity, such as size, and model accuracy, such as fitness and precision.

In Chapter 7, we extracted two further case studies of patients with breast cancer. The complexity of the models varied depending on the number of patients, number of events and sparsity. The aim of extracting these case studies was to test the proposed method of discovering the general process model with more complex processes. The proposed method was applied on both case studies, however, in case study 3 we needed to apply hierarchical modelling for complex

states. The results of the extraction have also shown that the changing of treatment types for breast cancer that happened in case study 2 increased both the complexity and the process variation in comparison to case study 1. However, the changing of chemo-regimens that happened in case study 3 has not increased the variation in comparison to case study 2, though this has increased the complexity of the model. In order to have more details about a specific cohort of patients, we proposed a top-down strategy for selecting and analysing similar cohorts based on the similarity of their processes. At the last stage, models were evaluated and the results showed that models' complexity had improved in both case studies based on different metrics such as size, control-flow-complexity (CFC) and structuredness. Furthermore, models' accuracy metrics based on fitness, precision and generalization had improved in both case studies. Also, the models were evaluated with domain experts where they have agreed on the correctness of the resulted process model and they emphasised the clear understandability of the healthcare process models for all cases of breast cancer patients.

8.5 Limitations

Although this research has successfully improved model understandability and reduced healthcare model complexity, there are some limitations that should be stated:

- A methodological limitation related to the use of hidden Markov models which is subjected to Markov assumption. The assumption implies that the transition to the next state depends only on the current state. This makes the model not able to capture the long dependency between events.
- When discovering the abstract process models, discovering the main pattern of care required several steps of data transformation and selection of a number of events, especially when solving the batch events. Therefore, the selection of events may have impacted our abstract model.
- We have successfully developed a method that targeted complexity. However, referring to the properties, that are discussed in section 2.6, of the methods that should be able to address the process complexity, our method needs to support more flexible customization. This means the developed method in this research supported two types of optimisations, but there might be other aspects which can be explored based on user preference. For instance, the tool can support different options of visualizing the best model based on single criteria such as the model with the best linearity or best cross state similarity.
- Long computational processing time that is required for the stages of model learning and decoding.
- Our method is not intended to discover business process structures such as parallel, choice and exclusive-or. We do not consider this a problem as these constructors rarely exist in

healthcare processes due to the impact of variety and flexibility.

8.6 Future work

The methodology developed in this thesis could be extended and applied in the future in the following ways:

- Integrating further functionality of HMMs. There is a potential of harnessing the probability nature of hidden Markov model to be used for operational support in healthcare processes such as prediction and recommendation of the next state of care process.

- Adopting an interactive abstracting approach for modelling hierarchical patients pathways. In case study 3, we have applied hierarchical modelling for complex state by manually extraction of the events. However, the approach for automating the hierarchical modelling can be done by extracting events related to the required state. Then, provides these events as a sub-log input for the subsequent stages of learning, decoding, optimising, selecting and visualizing. Figure 8.1 shows our vision for the extension proposed tool.

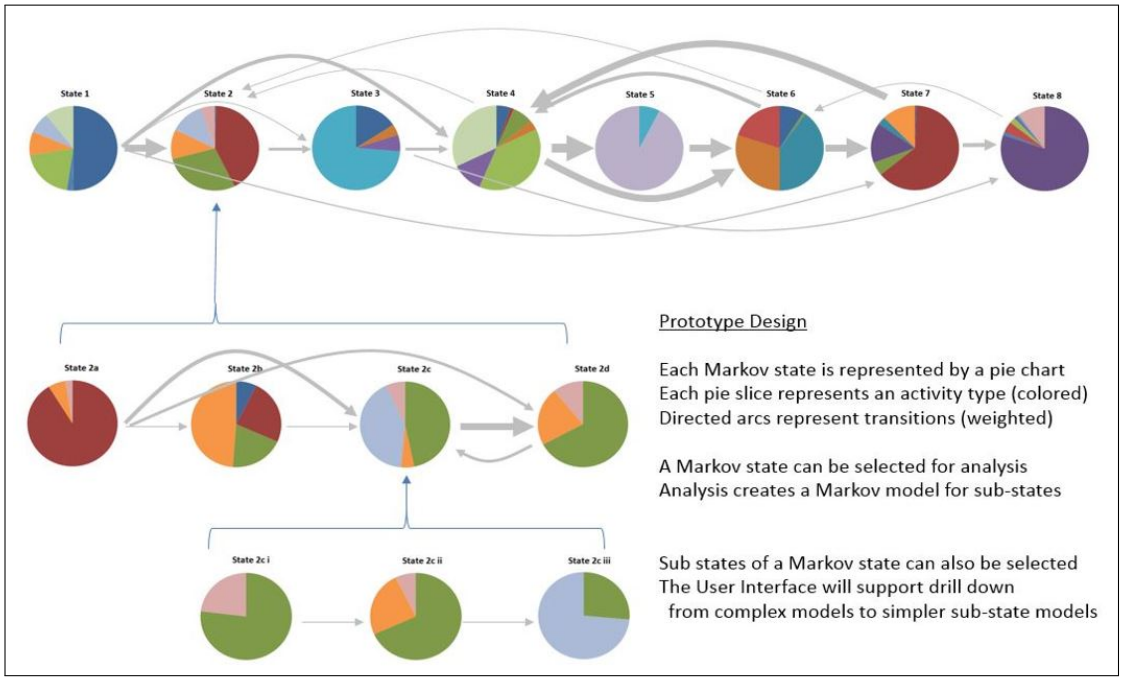


Figure 8.1: Our vision of an interactive hierarchical state modelling for mining patients pathway. This figure is presented in the conference where our work [3] was published.

The automation of our approach might be extended to include patients selecting from the abstracted HMM instead of selecting patients using another process mining tool.

- Using high-order hidden Markov models to capture long dependencies between events and not only first order-hidden Markov model.
- Broadening the scope of the proposed optimisation and its applicability in other domains. Our optimisation method could be used alone as a metric for selecting the best model of hidden Markov model in sparse data in non-healthcare domains.
- Using state types, such as simple or composite types, and the similarity score for merging similar states as another method to cope with the limitations of BIC metric in sparse data.
- Involving process temporal constraints on state transition matrix. For example; if there is a persistent state which means patients may take longer time in this state we could make use of time constraint on state change. However this may require domain knowledge to set the appropriate related time for each state.
- Investigate the applicability of using other algorithms that are designed for handling different sequences such as genetic sequences alignment algorithm and examining the strength and weakness of using such cross disciplinary algorithms.

In conclusion, this thesis has shown that our method of unsupervised abstraction of care events has successfully discovered the general process models of complex healthcare processes represented in three real case studies of breast cancer patients from Leeds Cancer Centre. Our developed abstraction method supported an automatic abstraction for start-to-end process models and discovered the general care of pattern for a complex large event log with the ability of handling process variations. Moreover, the generated abstracted process models could be evaluated and assessed within the available process mining frameworks besides the capability of distinguishing care events that occurred in different contexts of the process.

Appendices

A ICD9 code for Diabetes

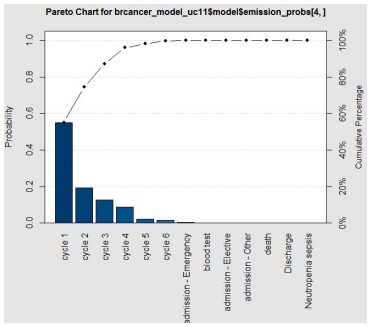
Type 2 Diabetes Mellitus: Commonly Used ICD-9 Codes according to [139].
V58.67 Long term, current insulin use
250.0 Diabetes mellitus without mention of complication
250.00 Diabetes mellitus without complication type 2 or unspecified type not stated as uncontrolled
250.02 Diabetes mellitus without complication type 2 or unspecified type uncontrolled
250.1 Diabetes with ketoacidosis
250.10 Diabetes mellitus with ketoacidosis type 2 or unspecified type not stated as uncontrolled
250.12 Diabetes mellitus with ketoacidosis type 2 or unspecified type uncontrolled
250. 4 Diabetes with renal manifestations
250.40 Diabetes mellitus with renal manifestations type 2 or unspecified type not stated as uncontrolled
250.42 Diabetes mellitus with renal manifestations type 2 or unspecified type uncontrolled
250.5 Diabetes with ophthalmic manifestations
250.50 Diabetes mellitus with ophthalmic manifestations type 2 or unspecified type not stated as uncontrolled
250.52 Diabetes mellitus with ophthalmic manifestations type 2 or unspecified type uncontrolled
250.6 Diabetes with neurological manifestations
250.60 Diabetes mellitus with neurological manifestations type 2 or unspecified type not stated as uncontrolled
250.62 Diabetes mellitus with neurological manifestations type 2 or unspecified type uncontrolled
250.7 Diabetes with peripheral circulatory disorders
250.70 Diabetes mellitus with peripheral circulatory disorders type 2 or unspecified type not stated as uncontrolled
250.72 Diabetes mellitus with peripheral circulatory disorders type 2 or unspecified type uncontrolled
250.9 Diabetes with unspecified complication
250.90 Diabetes mellitus with unspecified complication type 2 or unspecified type not stated as uncontrolled
250.92 Diabetes mellitus with unspecified complication type 2 or unspecified type uncontrolled

B ICD9 code for Colorectal caner

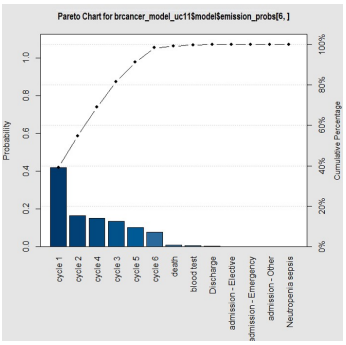
Colorectal cancer mostly used ICD9 code according to [139].
V10.05 Personal history of malignant neoplasm of large intestine (high risk screening code)
V10.06 Personal history of malignant neoplasm of rectum, rectosigmoid junction, and anus (high risk screening code)
V12.72 Personal history of adenomatous colonic polyps (high risk screening code)
V16.0 Family history of malignant neoplasm of gastrointestinal tract (first degree relative-sibling, parent, child) (high risk screening code)
V18.51 Family history, adenomatous colonic polyps (high risk screening code)
V76.41 Special screening for malignant neoplasms of rectum
V76.51 Special screening for malignant neoplasm of colon
V84.09 Genetic susceptibility to other malignant neoplasm (not covered by all payers)
153.0-154.9 Malignant neoplasm of colon, rectum, rectosigmoid junction and anus
209.11-209.17 Malignant carcinoid tumours of the appendix, large intestine, and rectum
209.50-209.57 Benign carcinoid tumours of the appendix, large intestine, and rectum
211.3 Benign neoplasm of colon
211.4 Benign neoplasm of rectum and anal canal
569.0 Anal and rectal polyp

C Checking the presence of multiple similar states

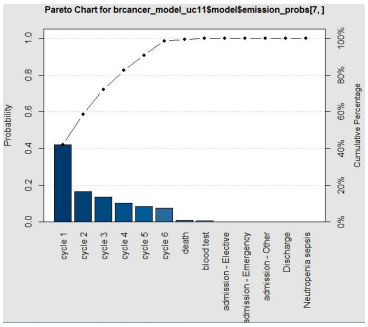
The following figures are for simple and composite states types of the best models selected in the three case studies that were demonstrated in chapter 6 and chapter 7.



(a) simple state (state 4)

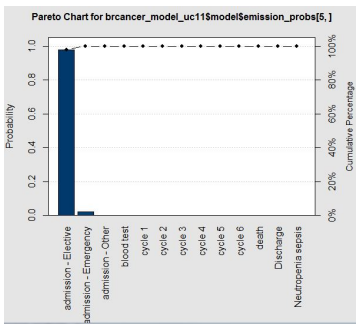


(b) composite state (state 6)

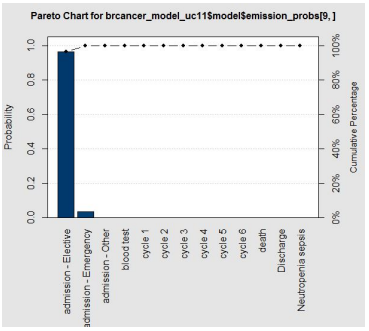


(c) composite state (state 7)

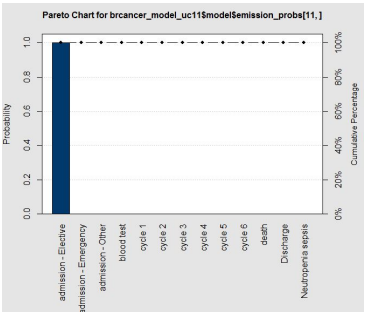
1: states has same main events (chemotherapy cycles)



(d) simple state (state 5)



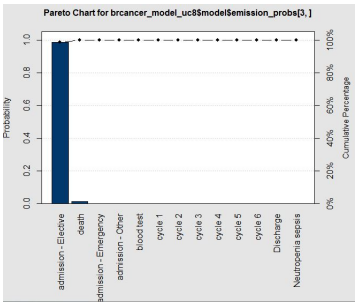
(e) simple state (state 9)



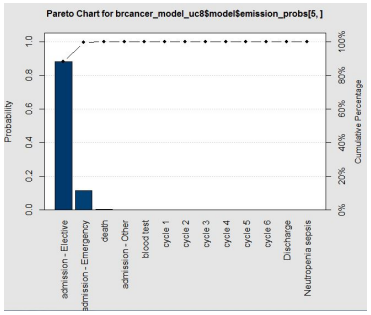
(f) simple state (state 11)

2: states has same main events (Admissions)

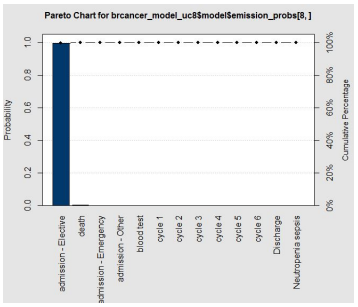
Pareto chart of suspicious similar states of 12states model that is selected by BIC in case study 1



(a) simple state (state 3)

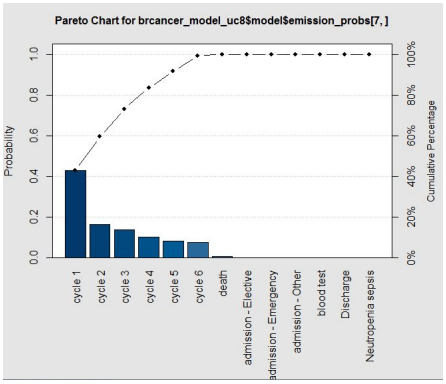


(b) simple state (state 5)

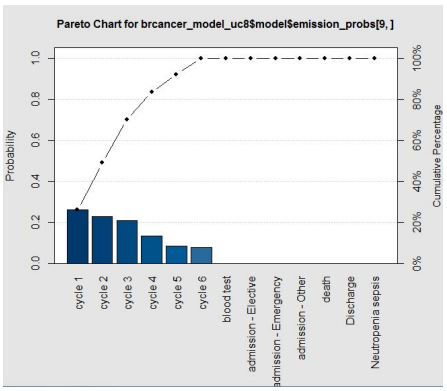


(c) simple state (state 8)

1: states has same main events (Admission-Elective)



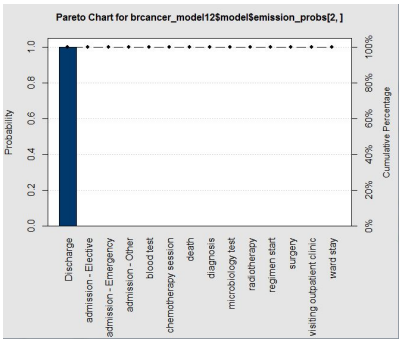
(d) composite state (state 7)



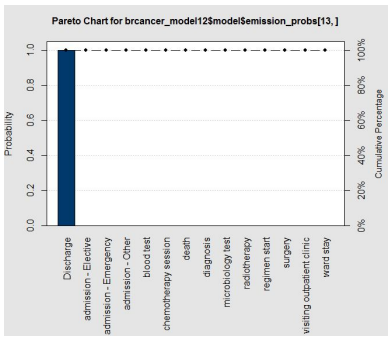
(e) composite state (state 9)

2: states has same main events (chemotherapy cycles)

Pareto chart of suspicious similar states of 9states model that is selected by soft optimisation in case study 1

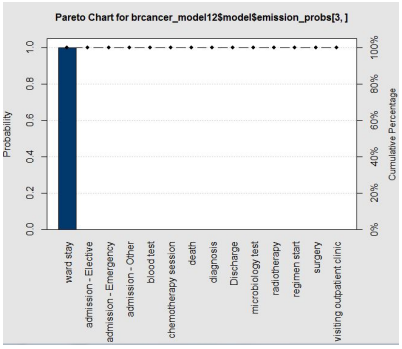


(a) simple state (state 2)

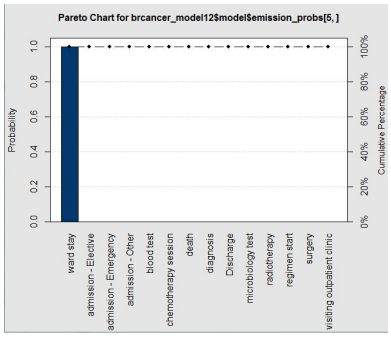


(b) simple state (state 13)

1: states has same main events (Discharge)

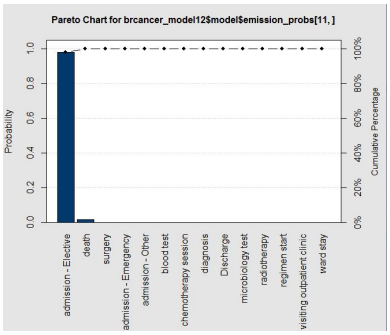


(c) simple state (state 3)

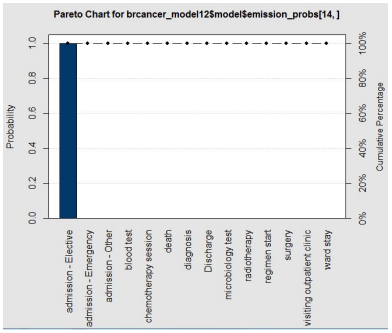


(d) simple state (state 5)

2: states has same main events (Ward stay)



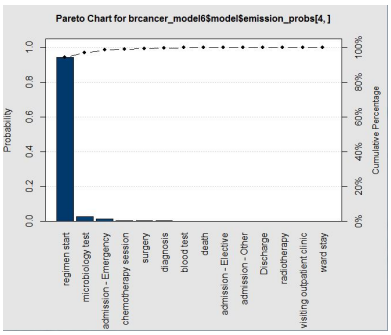
(e) simple state (state 11)



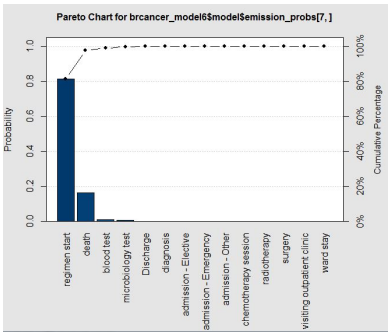
(f) simple state (state 14)

3: states has same main events (Admission-Elective)

Pareto chart of suspicious similar states of 14states model that is selected by BIC in case study 2



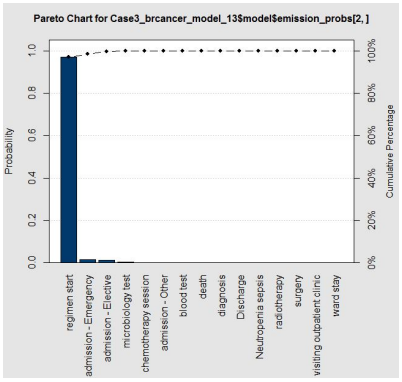
(a) simple state (state 4)



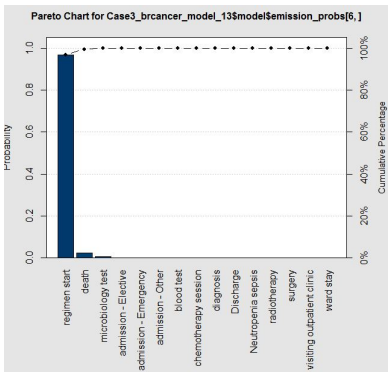
(b) simple state (state 7)

2: states has same main events (Regimen start)

Pareto chart of suspicious similar states of 8states model that is selected by strict and soft optimisation in case study 2

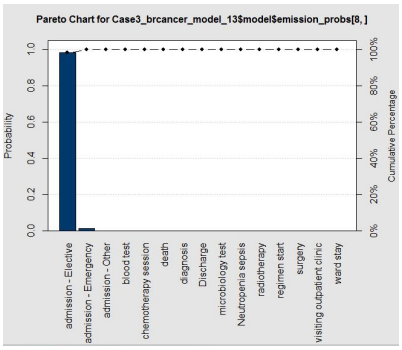


(a) simple state (state 2)

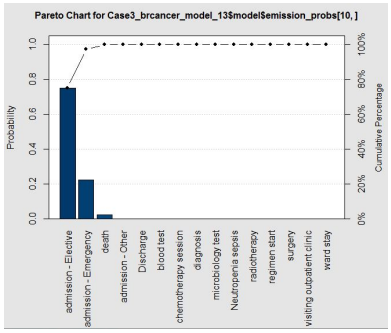


(b) simple state (state 6)

1: states has same main events (Regimen start)

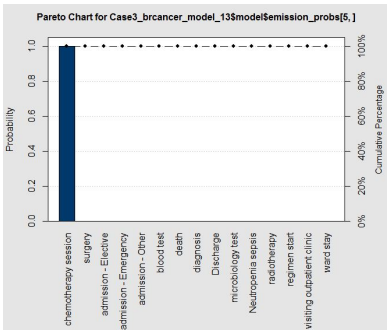


(c) simple state (state 8)

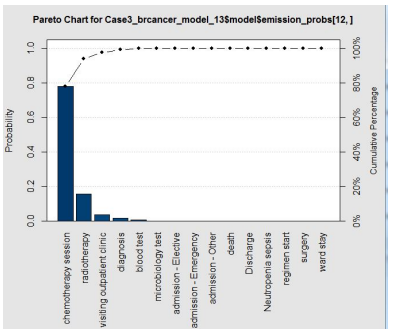


(d) simple state (state 10)

2: states has same main events (Admissions)



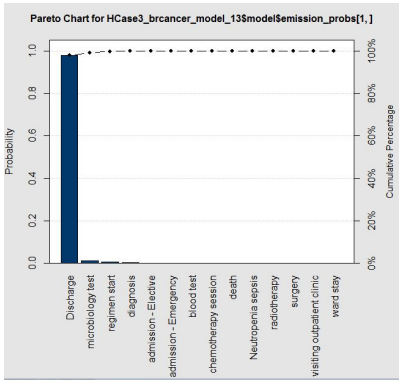
(e) simple state (state 5)



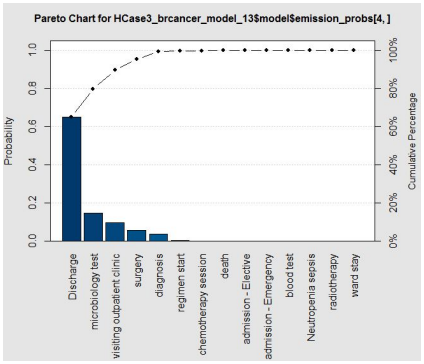
(f) simple state (state 12)

3: states has same main events (chemotherapy session)

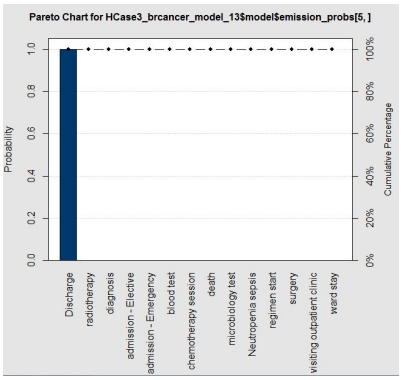
Pareto chart of suspicious similar states of 14states model that is selected by BIC in case study 3



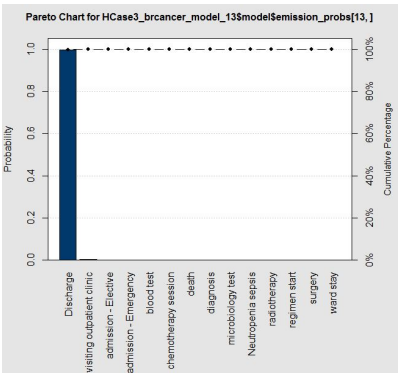
(a) simple state (state 1)



(b) simple state (state 4)

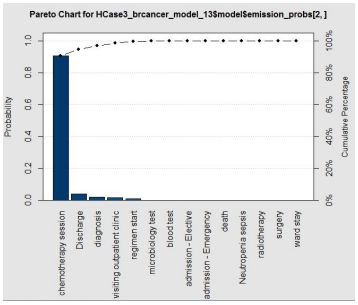


(c) simple state (state 5)

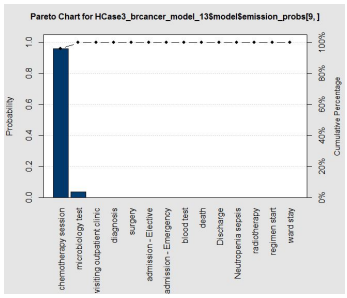


(d) simple state (state 13)

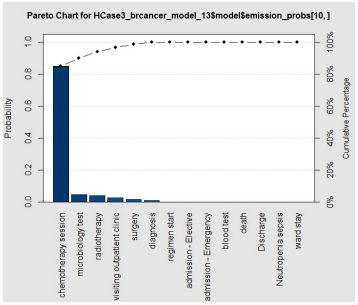
1: states has same main events (Discharge)



(e) simple state (state 2)



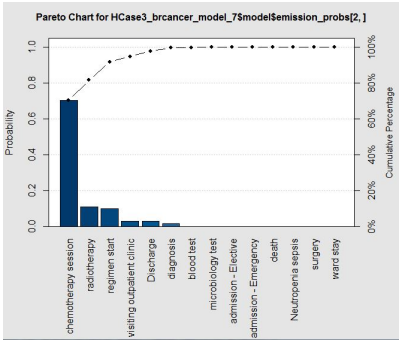
(f) simple state (state 9)



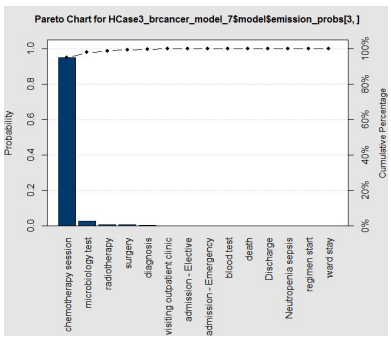
(g) simple state (state 10)

2: states has same main events (chemotherapy session)

Pareto chart of suspicious similar states of 14states model in hierarchical case study 3



(a) simple state (state 2)



(b) simple state (state 3)

1: states has same main events (chemotherapy session)

Pareto chart of suspicious similar states of 8states model in hierarchical case study 3

D Package 'AbstractHMM'

Package 'AbstractHMM'

June , 2019

Title Selecting the best Hidden Markov Model for abstracting complex process models.
Version 1.0
Description This package is designed to select the best Hidden Markov Model for abstracting complex process models. This version of the package is compatible with the structure of HMM that is trained using 'seqhmm' package in R. The package provides four different criteria that can be used for selecting the best number of hidden states. It is used as an alternative for information criteria metrics such as BIC and AIC. Models should be trained in unsupervised before using this package. Two types of optimisation are supported which are strict and soft optimisation. The strict optimisation takes state importance into consideration and selects a model that has a high state coverage in every state. Soft optimisation allows more flexibility towards model with low coverage states.
Depends R (>3.2.0)
Imports SeqHMM, igraph, devtools, stringdist, hashr, dplyr, ldply, ngram, bupa
License GPL (>2)
Encoding UTF-8
Author Amirah Alharbi
Repository CRAN
Package Functions:
1- linearity()
2- state_compactness()
3- cross_state_sim()
4- state_importance()
5- st_sf_optimization()
6- abstract_process_model()

linearity Compute to what extent a model is linear

Description

This function is designed to know what the main direction of Hidden Markov model is. The direction can be deduced from the probability of transition matrix. If the sum of the transitions of upper triangular part is higher than the sum of the transitions of lower triangular part then the model is mostly linear with left-to-right direction and vice versa.

Usage

linearity(HMM)

Arguments

This function takes a hidden Markov model as input and compute the linearity.

Value

linearity score

Dependent package

require(gdata)

state_compactness Calculate the compactness of a state

Description

This function calculates the compactness of states which will be used later as one criteria for the optimisation and selecting the best model. The model should be learned first using 'Seqhmm' package and decoded using the Viterbi algorithm. compactness is measured using an optimal string alignment (OSA) between processes inside a state.

Usage

state_compactness(HMM, viterbi_decoder = NULL, log)

Arguments

This function takes three inputs; HMM after learning using 'seqhmm', the Viterbi decoder is optional parameter if it is not available this function will decode the log with the viterbi decoder and the event log.

Value

compact_score which is the average compactness score of model's states.

See Also
fit_model(), hidden_paths()

Dependent package
require(stringdist) , require(hashr)

---

cross_state_sim Calculate the similarity score between model states

---

Description
This function calculates the similarity of the process between states. The criteria is computed based on the similarity of common nodes and common edges between processes.

Usage
cross_state_sim(HMM, decoder = NULL, log)

Arguments
This function takes three inputs; HMM after learning using 'seqhmm', the Viterbi decoder is optional parameter if it is not available this function will decode the log with the viterbi decoder and the event log.

Value
cross_sim_score which is the average of cross state similarity for a model.

See Also
fit_model(), hidden_paths()

Dependent package
require(plyr) , require(ngram)

---

state_importance check if state is important or not based on user threshold

---

Description
This function checks if a state is important or not based on the state coverage percentage which is defined by the user. It counts how many states that are not important in a model.

Usage
state_importance(HMM, decoder = NULL, log , total_cases, importance_threshold)

Arguments

This function takes five inputs; HMM after learning using 'seqhmm', the Viterbi decoder is optional parameter if it is not available this function will decode the log with the viterbi decoder, the event log, the total number of cases that have been learnt and percentage threshold.

Value

importance_weight which shows the number of unimportant states to be used later in the strict optimisation.

Examples

state_importance(HMM1, decoder1 , eventLog , 100, 50)

---

st_sf_optimization Strict and Soft optimisation for candidates of HMMs

---

Description

This function optimize the space of HMMs candidates in order to select the best abstract process model. The optimisation is based on four criteria, linearity, state compactness, cross state similarity and state importance.

Usage

st_sf_optimization(linearity, compact_score, cross_sim_score, importance_weight = NULL)

Arguments

This function takes four input parameters. If the parameter importance_weight is Null then the soft optimisation is applied using this formula;
$f(1) = max\{2(l_i) - \frac{1}{2}(sc_i) - css_i : i = 1..n\}$

otherwise the strict optimisation is computed using the following formula;
$f(2) = max\{1 - si_i[2(l_i) - \frac{1}{2}(sc_i) - css_i] : i = 1..n\}$

Value

optimization_result which is a vector of two elements. The first element is the optimisation score and the second element is the index of the model.

---

abstract_process_model Visualization of the abstract process model

---

Description

This function provides a visualization of the best process model that is selected after the optimisation. The visualization includes two steps. The first step aims to know the structure of

the abstract model, for instance how many node a model has. The second step aims to fill the nodes with the relevant events and assigns different colour for each event.

Usage
abstract_process_model(graph, log)

Arguments
This function takes two inputs which are graph for the abstract model and the event log that is enriched with state number for every event.

Value
igraph object for the abstract process model.

Example
log = read_xes(eventlog)
precedence = log %>% precedence_matrix()
edgelist= as.matrix(precedence[,1:2])
g = graph_from_edgelist(edgelist, directed = TRUE)
abstract_process_model(g, log)

Dependent package
require(qdap), require(dplyr), require(bupaR)

E MIMIC-III event log example

Case ID	Activity	Complete Timestamp	Variant	subject_id	icu_id	cgid	item_id	cost	11states	10states
100088	Admission	(31/05/2176 15:20:00)	Variant 1	22872	0	0	0	0	10	7
100088	Transfers	(31/05/2176 15:50:00)	Variant 1	22872	0	0	0	0.53	9	2
100088	Transfers	(31/05/2176 16:22:00)	Variant 1	22872	0	0	0	0.01	9	2
100088	Transfers	(31/05/2176 16:23:00)	Variant 1	22872	0	0	0	0	9	2
100088	Transfers	(31/05/2176 16:31:00)	Variant 1	22872	0	0	0	150.97	9	2
100088	NoteEvents	(31/05/2176 18:17:00)	Variant 1	22872	0	0	0	0	11	3
100088	Labevent	(31/05/2176 19:20:00)	Variant 1	22872	0	0	51222	abnormal	11	3
100088	Labevent	(01/06/2176 05:00:00)	Variant 1	22872	0	0	50861	0	11	3
100088	NoteEvents	(01/06/2176 09:57:00)	Variant 1	22872	0	0	0	0	11	3
100088	Labevent	(02/06/2176 05:00:00)	Variant 1	22872	0	0	50971	0	11	3
100088	Labevent	(03/06/2176 04:25:00)	Variant 1	22872	0	0	51006	abnormal	11	3
100088	Labevent	(04/06/2176 04:45:00)	Variant 1	22872	0	0	51249	0	11	3
100088	Labevent	(05/06/2176 04:30:00)	Variant 1	22872	0	0	50861	0	11	3
100088	NoteEvents	(05/06/2176 10:21:00)	Variant 1	22872	0	0	0	0	11	3
100088	Labevent	(05/06/2176 22:46:00)	Variant 1	22872	0	0	51498	0	11	3
100088	Labevent	(06/06/2176 09:51:00)	Variant 1	22872	0	0	50822	abnormal	11	3
100088	Labevent	(06/06/2176 11:47:00)	Variant 1	22872	0	0	50822	0	11	3
100088	Labevent	(06/06/2176 13:14:00)	Variant 1	22872	0	0	50802	0	11	3
100088	Labevent	(06/06/2176 15:21:00)	Variant 1	22872	0	0	50983	0	11	3
100088	NoteEvents	(06/06/2176 19:29:00)	Variant 1	22872	0	0	0	0	11	3
100088	Labevent	(06/06/2176 20:53:00)	Variant 1	22872	0	0	51221	abnormal	11	3
100088	Chart	(06/06/2176 21:16:00)	Variant 1	22872	238680	15331	1535	0	6	3
100088	Labevent	(06/06/2176 21:16:00)	Variant 1	22872	0	0	50893	0	6	3

100088	Labevent	(06/06/2176 21:34:00)	Variant 1	22872	0	0	50808	0	6	3
100088	Chart	(06/06/2176 22:30:00)	Variant 1	22872	238680	19295	763	0	6	7
100088	Chart	(06/06/2176 22:33:00)	Variant 1	22872	238680	14077	69	0	6	2
100088	Labevent	(06/06/2176 22:51:00)	Variant 1	22872	0	0	50910	0	6	3
100088	Chart	(06/06/2176 22:51:00)	Variant 1	22872	238680	15331	785	0	6	7
100088	Transfers	(06/06/2176 23:29:00)	Variant 1	22872	238680	0	0	87.57	6	2
100088	Input	(07/06/2176 00:00:00)	Variant 1	22872	238680	19295	30013	0	7	10
100088	Output	(07/06/2176 01:00:00)	Variant 1	22872	0	19295	40055	0	7	8
100088	Output	(07/06/2176 02:00:00)	Variant 1	22872	0	19295	40055	0	7	10
100088	Chart	(07/06/2176 02:48:00)	Variant 1	22872	238680	15331	1542	0	8	9
100088	Labevent	(07/06/2176 02:48:00)	Variant 1	22872	0	0	51277	abnormal	5	5
100088	Output	(07/06/2176 03:00:00)	Variant 1	22872	0	19295	40055	0	7	8
100088	Output	(07/06/2176 04:00:00)	Variant 1	22872	0	19295	40055	0	7	10
100088	NoteEvents	(07/06/2176 04:40:00)	Variant 1	22872	0	19295	0	0	7	8
100088	Chart	(07/06/2176 05:00:00)	Variant 1	22872	238680	19295	455	0	8	9
100088	Output	(07/06/2176 05:00:00)	Variant 1	22872	0	19295	40055	0	5	5
100088	Output	(07/06/2176 06:00:00)	Variant 1	22872	0	19295	40614	0	7	10
100088	Chart	(07/06/2176 07:00:00)	Variant 1	22872	238680	14204	549	0	8	9
100088	Input	(07/06/2176 08:00:00)	Variant 1	22872	238680	14204	30021	0	5	5
100088	Output	(07/06/2176 09:00:00)	Variant 1	22872	0	14204	40055	0	7	10
100088	Chart	(07/06/2176 10:00:00)	Variant 1	22872	238680	14204	455	0	8	9
100088	Output	(07/06/2176 10:00:00)	Variant 1	22872	0	14204	40055	0	5	5
100088	Chart	(07/06/2176 11:00:00)	Variant 1	22872	238680	14204	674	0	8	9
100088	Output	(07/06/2176 11:00:00)	Variant 1	22872	0	14204	40055	0	5	5
100088	Output	(07/06/2176 12:00:00)	Variant 1	22872	0	14204	40055	0	7	10
100088	Output	(07/06/2176 13:00:00)	Variant 1	22872	0	14204	40055	0	7	8

100088	Output	(07/06/2176 14:00:00)	Variant 1	22872	0	14204	40055	0	7	10
100088	Output	(07/06/2176 15:00:00)	Variant 1	22872	0	14204	40055	0	7	8
100088	Output	(07/06/2176 16:00:00)	Variant 1	22872	0	14204	40055	0	7	10
100088	Chart	(07/06/2176 17:00:00)	Variant 1	22872	238680	14204	211	0	8	9
100088	Input	(07/06/2176 17:00:00)	Variant 1	22872	238680	14204	30013	0	5	5
100088	Output	(07/06/2176 18:00:00)	Variant 1	22872	0	14204	40055	0	7	10
100088	NoteEvents	(07/06/2176 18:27:00)	Variant 1	22872	0	14204	0	0	7	8
100088	Chart	(07/06/2176 19:00:00)	Variant 1	22872	238680	19295	455	0	8	9
100088	Input	(07/06/2176 19:00:00)	Variant 1	22872	238680	19295	30096	0	5	5
100088	Chart	(07/06/2176 20:00:00)	Variant 1	22872	238680	19295	455	0	8	9
100088	Output	(07/06/2176 20:00:00)	Variant 1	22872	0	19295	40055	0	5	5
100088	Output	(07/06/2176 21:00:00)	Variant 1	22872	0	19295	40055	0	7	10
100088	Chart	(07/06/2176 22:00:00)	Variant 1	22872	238680	19295	455	0	8	9
100088	Output	(07/06/2176 22:00:00)	Variant 1	22872	0	19295	40055	0	5	5
100088	Output	(07/06/2176 23:00:00)	Variant 1	22872	0	19295	40055	0	7	10
100088	Chart	(08/06/2176 00:00:00)	Variant 1	22872	238680	19295	455	0	8	9
100088	Output	(08/06/2176 00:00:00)	Variant 1	22872	0	19295	40055	0	5	5
100088	Output	(08/06/2176 01:00:00)	Variant 1	22872	0	19295	40055	0	7	10
100088	Chart	(08/06/2176 02:00:00)	Variant 1	22872	238680	19295	455	0	8	9
100088	Output	(08/06/2176 02:00:00)	Variant 1	22872	0	19295	40055	0	5	5
100088	Output	(08/06/2176 03:00:00)	Variant 1	22872	0	19295	40055	0	7	10
100088	Chart	(08/06/2176 03:19:00)	Variant 1	22872	238680	15331	1525	0	8	9
100088	Labevent	(08/06/2176 03:19:00)	Variant 1	22872	0	0	50912	0	5	5
100088	Chart	(08/06/2176 04:00:00)	Variant 1	22872	238680	19295	455	0	8	9
100088	Output	(08/06/2176 04:00:00)	Variant 1	22872	0	19295	40055	0	5	5
100088	Output	(08/06/2176 05:00:00)	Variant 1	22872	0	19295	40055	0	7	10

100088	NoteEvents	(08/06/2176 05:01:00)	Variant 1	22872	0	19295	0	0	7	8
100088	Output	(08/06/2176 06:00:00)	Variant 1	22872	0	21167	40055	0	7	10
100088	Input	(08/06/2176 07:00:00)	Variant 1	22872	238680	17746	30073	0	7	8
100088	Chart	(08/06/2176 08:00:00)	Variant 1	22872	238680	17746	211	0	8	9
100088	Output	(08/06/2176 08:00:00)	Variant 1	22872	0	17746	40052	0	5	5
100088	Chart	(08/06/2176 09:00:00)	Variant 1	22872	238680	17746	455	0	8	6
100088	Output	(08/06/2176 09:00:00)	Variant 1	22872	0	17746	40055	0	5	10
100088	Output	(08/06/2176 10:00:00)	Variant 1	22872	0	17746	40055	0	7	8
100088	Output	(08/06/2176 11:00:00)	Variant 1	22872	0	17746	40055	0	7	10
100088	Chart	(08/06/2176 12:00:00)	Variant 1	22872	238680	17746	8548	0	8	9
100088	Output	(08/06/2176 12:00:00)	Variant 1	22872	0	17746	40055	0	5	5
100088	Output	(08/06/2176 13:00:00)	Variant 1	22872	0	17746	40055	0	7	10
100088	Chart	(08/06/2176 14:00:00)	Variant 1	22872	238680	17746	455	0	8	9
100088	Output	(08/06/2176 14:00:00)	Variant 1	22872	0	17746	40055	0	5	5
100088	Output	(08/06/2176 15:00:00)	Variant 1	22872	0	17746	40055	0	7	10
100088	Output	(08/06/2176 16:00:00)	Variant 1	22872	0	17746	40055	0	7	8
100088	Output	(08/06/2176 17:00:00)	Variant 1	22872	0	17746	40055	0	7	10
100088	Output	(08/06/2176 18:00:00)	Variant 1	22872	0	17746	40055	0	7	8
100088	Output	(08/06/2176 19:00:00)	Variant 1	22872	0	17746	40055	0	7	10
100088	NoteEvents	(08/06/2176 19:06:00)	Variant 1	22872	0	17746	0	0	7	8
100088	Output	(08/06/2176 20:00:00)	Variant 1	22872	0	17770	40055	0	7	10
100088	Input	(08/06/2176 21:00:00)	Variant 1	22872	238680	17770	30021	0	7	8
100088	Chart	(08/06/2176 22:00:00)	Variant 1	22872	238680	17770	706	0	8	9
100088	Output	(08/06/2176 22:00:00)	Variant 1	22872	0	21570	40614	0	5	5

F PPM event log example

Case ID	Activity	Complete Timestamp	Variant	4states	8states
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(06/06/2076 12:00:00)	Variant 34	4	2
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(27/01/2080 12:00:00)	Variant 34	4	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(11/02/2080 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(17/03/2080 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(23/04/2080 12:00:00)	Variant 34	1	2
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(23/04/2080 12:00:00)	Variant 34	2	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(28/07/2080 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(06/10/2080 12:00:00)	Variant 34	4	2
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(31/08/2081 12:00:00)	Variant 34	4	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(01/02/2082 12:00:00)	Variant 34	4	3
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(26/04/2082 12:00:00)	Variant 34	1	6
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(26/04/2082 12:00:00)	Variant 34	2	2
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(13/03/2083 12:00:00)	Variant 34	4	4
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(13/04/2083 12:00:00)	Variant 34	1	5
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(13/04/2083 12:00:00)	Variant 34	2	3
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(18/07/2083 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(20/07/2083 12:00:00)	Variant 34	1	2
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(24/07/2083 12:00:00)	Variant 34	2	5
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(31/08/2083 12:00:00)	Variant 34	1	3
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(03/09/2083 12:00:00)	Variant 34	2	6
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(03/09/2083 12:00:00)	Variant 34	4	2
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(28/09/2083 12:00:00)	Variant 34	4	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(10/10/2083 12:00:00)	Variant 34	4	3

00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(11/10/2083 12:00:00)	Variant 34	1	6
00036B8E-55A1-4FDF-B795-F51C392EE641	cycle 1	(11/10/2083 12:00:00)	Variant 34	3	2
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(11/10/2083 12:00:00)	Variant 34	2	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(29/10/2083 12:00:00)	Variant 34	4	3
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(01/11/2083 12:00:00)	Variant 34	1	6
00036B8E-55A1-4FDF-B795-F51C392EE641	cycle 2	(01/11/2083 12:00:00)	Variant 34	3	2
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(01/11/2083 12:00:00)	Variant 34	2	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(21/11/2083 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(22/11/2083 12:00:00)	Variant 34	1	2
00036B8E-55A1-4FDF-B795-F51C392EE641	cycle 3	(22/11/2083 12:00:00)	Variant 34	3	5
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(22/11/2083 12:00:00)	Variant 34	2	3
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(22/11/2083 12:00:00)	Variant 34	4	3
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(08/12/2083 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(09/12/2083 12:00:00)	Variant 34	1	2
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(10/12/2083 12:00:00)	Variant 34	2	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(10/12/2083 12:00:00)	Variant 34	4	3
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(19/12/2083 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(20/12/2083 12:00:00)	Variant 34	1	2
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(20/12/2083 12:00:00)	Variant 34	2	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(02/01/2084 12:00:00)	Variant 34	4	3
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(09/01/2084 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(10/01/2084 12:00:00)	Variant 34	1	7
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(10/01/2084 12:00:00)	Variant 34	2	8
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(28/01/2084 12:00:00)	Variant 34	4	2
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(31/01/2084 12:00:00)	Variant 34	1	5
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(31/01/2084 12:00:00)	Variant 34	2	6

00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Emergency	(07/02/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(07/02/2084 12:00:00)	Variant 34	4	2
00036B8E-55A1-4FDF-B795-F51C392EE641	Neutropenia sepsis	(07/02/2084 12:00:00)	Variant 34	4	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(11/02/2084 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	Neutropenia sepsis	(11/02/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(12/02/2084 12:00:00)	Variant 34	4	2
00036B8E-55A1-4FDF-B795-F51C392EE641	Neutropenia sepsis	(12/02/2084 12:00:00)	Variant 34	4	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(13/02/2084 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(15/02/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(16/02/2084 12:00:00)	Variant 34	4	2
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Emergency	(19/02/2084 12:00:00)	Variant 34	4	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(19/02/2084 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(21/02/2084 12:00:00)	Variant 34	4	2
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(22/02/2084 12:00:00)	Variant 34	4	5
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(23/02/2084 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Emergency	(03/03/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(03/03/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(04/03/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(05/03/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(06/03/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(07/03/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Emergency	(10/04/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(10/04/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(12/04/2084 12:00:00)	Variant 34	4	2
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(09/06/2084 12:00:00)	Variant 34	4	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(15/07/2084 12:00:00)	Variant 34	4	1

00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(25/07/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(31/07/2084 12:00:00)	Variant 34	1	1
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(31/07/2084 12:00:00)	Variant 34	2	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(20/08/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(03/09/2084 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(06/01/2085 12:00:00)	Variant 34	1	1
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(06/01/2085 12:00:00)	Variant 34	2	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(10/03/2085 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(17/03/2085 12:00:00)	Variant 34	1	1
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(17/03/2085 12:00:00)	Variant 34	2	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(26/03/2085 12:00:00)	Variant 34	4	7
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(15/04/2085 12:00:00)	Variant 34	4	8
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(22/04/2085 12:00:00)	Variant 34	1	1
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(22/04/2085 12:00:00)	Variant 34	2	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(22/09/2085 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(23/04/2086 12:00:00)	Variant 34	4	2
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(14/05/2086 12:00:00)	Variant 34	1	5
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(14/05/2086 12:00:00)	Variant 34	2	6
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(31/05/2086 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	admission - Elective	(09/07/2086 12:00:00)	Variant 34	1	2
00036B8E-55A1-4FDF-B795-F51C392EE641	Discharge	(09/07/2086 12:00:00)	Variant 34	2	5
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(15/09/2086 12:00:00)	Variant 34	4	6
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(27/10/2086 12:00:00)	Variant 34	4	1
00036B8E-55A1-4FDF-B795-F51C392EE641	blood test	(27/04/2087 12:00:00)	Variant 34	4	2

Bibliography

[1] Thomas G Kannampallil, Guido F Schauer, Trevor Cohen, and Vimla L Patel. Considering complexity in healthcare systems. *Journal of Biomedical Informatics*, 44(6):943–947, 2011.

[2] Niek Tax, Xixi Lu, Natalia Sidorova, Dirk Fahland, and Wil MP van der Aalst. The imprecisions of precision measures in process mining. *Information Processing Letters*, 135:1–8, 2018.

[3] Amirah Alharbi, Andy Bulpitt, and Owen Johnson. Towards unsupervised detection of process models in healthcare. In *Medical Informatics Europe*, pages 381–385, 2018.

[4] Owen Johnson, Thamer Ba Dhafari, Angelina Kurniati, Frank Fox, and Eric Rojas. The clearpath method for care pathway process mining and simulation. In *International Conference on Business Process Management*, pages 239–250. Springer, 2018.

[5] Ronny S Mans, Wil MP Van der Aalst, and Rob JB Vanwersch. *Process mining in healthcare: Evaluating and exploiting operational healthcare processes*. Springer, 2015.

[6] JD Van der Bij and JMH Vissers. Monitoring health-care processes: a framework for performance indicators. *International Journal of Health Care Quality Assurance*, 12(5):214–221, 1999.

[7] Anthony Culyer, Christopher McCabe, Andrew Briggs, Karl Claxton, Martin Buxton, Ron Akehurst, Mark Sculpher, and John Brazier. Searching for a threshold, not setting one: the role of the national institute for health and clinical excellence. *Journal of Health Services Research & Policy*, 12(1):56–58, 2007.

[8] Ronny S Mans, Wil MP van der Aalst, Rob JB Vanwersch, and Arnold J Moleman. Process mining in healthcare: Data challenges when answering frequently posed questions. In *Process Support and Knowledge Representation in Health Care*, pages 140–153. Springer, 2012.

[9] Wil Van Der Aalst. *Process mining: discovery, conformance and enhancement of business processes*, volume 2. Springer, 2011.

[10] Laden Aldin and Sergio de Cesare. A literature review on business process modelling: new frontiers of reusability. *Enterprise Information Systems*, 5(3):359–383, 2011.

[11] Jonathan E Cook and Alexander L Wolf. *Process discovery and validation through event-data analysis*. PhD thesis, Citeseer, 1996.

[12] Rakesh Agrawal, Dimitrios Gunopulos, and Frank Leymann. Mining process models from workflow logs. In *International Conference on Extending Database Technology*, pages 467–483. Springer, 1998.

[13] Wil Van der Aalst, Ton Weijters, and Laura Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge & Data Engineering*, (9):1128–1142, 2004.

[14] Wil van der Aalst, Arya Adriansyah, Ana Karla Alves de Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos Buijs, Andrea Burattin, Josep Carmona, Malu Castellanos, Jan Claes, Jonathan Cook, Nicola Costantini, Francisco Curbera, Ernesto Damiani, Massimiliano de Leoni, Pavlos Delias, Boudewijn F. van Dongen, Marlon Dumas, Schahram Dustdar, Dirk Fahland, Diogo R. Ferreira, Walid Gaaloul, Frank van Geffen, Sukriti Goel, Christian Günther, Antonella Guzzo, Paul Harmon, Arthur ter Hofstede, John Hoogland, Jon Espen Ingvaldsen, Koki Kato, Rudolf Kuhn, Akhil Kumar, Marcello La Rosa, Fabrizio Maggi, Donato Malerba, Ronny S. Mans, Alberto Manuel, Martin McCreesh, Paola Mello, Jan Mendling, Marco Montali, Hamid R. Motahari-Nezhad, Michael zur Muehlen, Jorge Munoz-Gama, Luigi Pontieri, Joel Ribeiro, Anne Rozinat, Hugo Seguel Pérez, Ricardo Seguel Pérez, Marcos Sepúlveda, Jim Sinur, Pnina Soffer, Minseok Song, Alessandro Sperduti, Giovanni Stilo, Casper Stoel, Keith Swenson, Maurizio Talamo, Wei Tan, Chris Turner, Jan Vanthienen, George Varvaressos, Eric Verbeek, Marc Verdonk, Roberto Vigo, Jianmin Wang, Barbara Weber, Matthias Weidlich, Ton Weijters, Lijie Wen, Michael Westergaard, and Moe Wynn. Process mining manifesto. In Florian Daniel, Kamel Barkaoui, and Schahram Dustdar, editors, *Business Process Management Workshops*, pages 169–194, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[15] Orna Muller and Bruria Haberman. Supporting abstraction processes in problem solving through pattern-oriented instruction. *Computer Science Education*, 18(3):187–212, 2008.

[16] Karl Baker, Elaine Dunwoodie, Richard Jones, Alex Newsham, Owen Johnson, Christopher Price, Jane Wolstenholme, Jose Leal, Patrick McGinley, Chris Twelves, and Geoff Hall. Process mining routinely collected electronic health records to define real-life clinical pathways during chemotherapy. *International Journal of Medical Informatics*, 103:32–41, 2017.

[17] Payam Homayounfar. Process mining challenges in hospital information systems. In *Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1135–1140. IEEE, 2012.

[18] Wei Yang and Qiang Su. Process mining for clinical pathway: Literature review and future directions. In *11th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–5. IEEE, 2014.

[19] Eric Rojas, Jorge Munoz-Gama, Marcos Sepúlveda, and Daniel Capurro. Process mining in healthcare: A literature review. *Journal of Biomedical Informatics*, 61:224–236, 2016.

[20] Amirah Alharbi, Andy Bulpitt, and Owen Johnson. Improving pattern detection in healthcare process mining using an interval-based event selection method. In *International Conference on Business Process Management*, pages 88–105. Springer, 2017.

[21] Amirah Alharbi, Andy Bulpitt, Owen Johnson, and Eric Rojas. Multi-objective optimisation method for abstracting complex processes. *IEEE Journal of Biomedical and Health Informatics (Manuscript in preparation)*, 2019.

[22] Alistair Johnson, Tom Pollard, Lu Shen, Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016.

[23] Tom Pollard and Alistair Johnson. The mimic-iii clinical database [online]. `http://dx.doi.org/10.13026/C2XW26` [Accessed: 2016].

[24] Owen Johnson and Sofela Emmanuel Abiodun. Understanding what success in health information systems looks like: the patient pathway management (ppm) system at leeds. In *UK Academy for Information Systems Conference Proceedings*, 2011.

[25] Satu Helske and Jouni Helske. Mixture hidden markov models for sequence data: the seqhmm package in r. *Journal of Statistical Software*, 2017.

[26] AJ Weijters and JT Ribeiro. Flexible heuristics miner (fhm). In *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 310–317. IEEE, 2011.

[27] Wil MP van der Aalst. Process mining: discovering and improving spaghetti and lasagna processes. In *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 1–7. IEEE, 2011.

[28] Ronny Mans, MH Schonenberg, Minseok Song, Wil MP van der Aalst, and Piet Bakker. Application of process mining in healthcare–a case study in a dutch hospital. In *International Joint Conference on Biomedical Engineering Systems and Technologies*, pages 425–438. Springer, 2008.

[29] Mahdi Ghasemi and Daniel Amyot. Process mining in healthcare: a systematised literature review. *International Journal of Electronic Healthcare (IJEH)*, 9(1):60–88, 2016.

[30] Tuğba Erdoğan and Ayça Tarhan. Process mining for healthcare process analytics. In *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, pages 125–130. IEEE, 2016.

[31] Angelina Kurniati, Owen Johnson, David Hogg, and Geoff Hall. Process mining in oncology: A literature review. In *6th International Conference on Information Communication and Management (ICICM)*, pages 291–297. IEEE, 2016.

[32] Guntur Kusuma, Marlous Hall, Chris Gale, and Owen Johnson. Process mining in cardiology: A literature review. *International Journal of Bioscience, Biochemistry and Bioinformatics*, 8(4):226–236, 2018.

[33] Nik Farid, Mark de Kamps, and Owen Johnson. Process mining in frail elderly care: A literature review. In *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies*. SciTePress, Science and Technology Publications, 2018.

[34] Richard Williams, Eric Rojas, Niels Peek, and Owen Johnson. Process mining in primary care: A literature review. *Studies in Health Technology and Informatics*, 247:376–380, 2018.

[35] Álvaro Rebuge and Diogo R Ferreira. Business process analysis in healthcare environments: A methodology based on process mining. *Information Systems*, 37(2):99–116, 2012.

[36] Gabriel Veiga and Diogo Ferreira. Understanding spaghetti models with sequence clustering for prom. In *International Conference on Business Process Management*, pages 92–103. Springer, 2009.

[37] Sander Leemans, Dirk Fahland, and Wil MP van der Aalst. Process and deviation exploration with inductive visual miner. *BPM (Demos)*, 1295:46, 2014.

[38] Loubna Bouarfa and Jenny Dankelman. Workflow mining and outlier detection from clinical activity logs. *Journal of Biomedical Informatics*, 45(6):1185–1190, 2012.

[39] BFA Hompes, JCAM Buijs, WMP Van der Aalst, PM Dixit, and J Buurman. Discovering deviating cases and process variants using trace clustering. In *Proceedings of the 27th Benelux Conference on Artificial Intelligence (BNAIC), November*, pages 5–6, 2015.

[40] Benoit Depaire, Jo Swinnen, Mieke Jans, and Koen Vanhoof. A process deviation analysis framework. In *International Conference on Business Process Management*, pages 701–706. Springer, 2012.

[41] Xiang Li, Jing Mei, Haifeng Liu, Yiqin Yu, Guotong Xie, Jianying Hu, and Fei Wang. Analysis of care pathway variation patterns in patient records. In *Medical Informatics Europe*, pages 692–696, 2015.

[42] Herbert A Simon. The architecture of complexity. In *Facets of Systems Science*, pages 457–476. Springer, 1991.

[43] Herbert A Simon. *The sciences of the artificial*. MIT press, 1996.

[44] Javier de San Pedro and Jordi Cortadella. Discovering duplicate tasks in transition systems for the simplification of process models. In *International Conference on Business Process Management*, pages 108–124. Springer, 2016.

[45] Paul Plsek and Trisha Greenhalgh. The challenge of complexity in health care: an introduction. *British Medical Journal*, 323(7314):625–628, 2001.

[46] Marlon Dumas, Marcello La Rosa, Jan Mendling, Raul Mäesalu, Hajo A Reijers, and Nataliia Semenenko. Understanding business process models: the costs and benefits of structuredness. In *International Conference on Advanced Information Systems Engineering*, pages 31–46. Springer, 2012.

[47] Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, Fabrizio M Maggi, Andrea Marrella, Massimo Mecella, and Allar Soo. Automated discovery of process models from event logs: Review and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 31(4):686–705, 2018.

[48] Arya Adriansyah, Jorge Munoz-Gama, Josep Carmona, Boudewijn F van Dongen, and Wil MP van der Aalst. Alignment based precision checking. In *International Conference on Business Process Management*, pages 137–149. Springer, 2012.

[49] Krzysztof Kluza. Measuring complexity of business process models integrated with rules. In *International Conference on Artificial Intelligence and Soft Computing*, pages 649–659. Springer, 2015.

[50] Jan Mendling. *Metrics for process models: empirical foundations of verification, error prediction, and guidelines for correctness*, volume 6. Springer Science & Business Media, 2008.

[51] Wil MP van der Aalst. On the representational bias in process mining. In *2011 IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 2–7. IEEE, 2011.

[52] Anton JMM Weijters and Wil MP Van der Aalst. Rediscovering workflow models from event-based data using little thumb. *Integrated Computer-Aided Engineering*, 10(2):151–162, 2003.

[53] Angelina Kurniati, Guntur Kusuma, and Gede Wisudiawan. Implementing heuristic miner for different types of event logs. *International Journal of Applied Engineering Research*, 11(8):5523–5529, 2016.

[54] Sander Leemans, Dirk Fahland, and Wil MP van der Aalst. Exploring processes and deviations. In *International Conference on Business Process Management*, pages 304–316. Springer, 2014.

[55] Sander Leemans, Dirk Fahland, and Wil MP van der Aalst. Discovering block-structured process models from event logs containing infrequent behaviour. In *International Conference on Business Process Management*, pages 66–78. Springer, 2013.

[56] Wil MP van der Aalst and Christian W Gunther. Finding structure in unstructured processes: The case for process mining. In *7th International Conference on Application of Concurrency to System Design (ACSD)*, pages 3–12. IEEE, 2007.

[57] Christian W Günther. *Process mining in flexible environments*. PhD thesis, Technische Universiteit Eindhoven, 2009.

[58] Christian W Günther and Wil MP Van Der Aalst. Fuzzy mining–adaptive process simplification based on multi-perspective metrics. In *International Conference on Business Process Management*, pages 328–343. Springer, 2007.

[59] Jiafei Li, RP Jagadeesh Chandra Bose, and Wil MP van der Aalst. Mining context-dependent and interactive business process maps using execution patterns. In *International Conference on Business Process Management*, pages 109–121. Springer, 2010.

[60] John V Guttag. *Introduction to computation and programming using Python*. MIT Press, 2013.

[61] Timothy Colburn and Gary Shute. Abstraction in computer science. *Minds and Machines*, 17(2):169–184, 2007.

[62] Alfred V Aho and Jeffrey D Ullman. *Foundations of computer science*. Computer Science Press, 1995.

[63] Thomas Baier, Jan Mendling, and Mathias Weske. Bridging abstraction layers in process mining. *Information Systems*, 46:123–139, 2014.

[64] Felix Mannhardt, Massimiliano De Leoni, Hajo A Reijers, Wil MP Van Der Aalst, and Pieter J Toussaint. From low-level events to activities-a pattern-based approach. In *International Conference on Business Process Management*, pages 125–141. Springer, 2016.

[65] Camilo Alvarez, Eric Rojas, Michael Arias, Jorge Munoz-Gama, Marcos Sepúlveda, Valeria Herskovic, and Daniel Capurro. Discovering role interaction models in the emergency room using process mining. *Journal of Biomedical Informatics*, 78:60–77, 2018.

[66] Carlos Fernandez-Llatas, Aroa Lizondo, Eduardo Monton, Jose-Miguel Benedi, and Vicente Traver. Process mining methodology for health process tracking using real-time indoor location systems. *Sensors*, 15(12):29821–29840, 2015.

[67] Jasmine Tehrani, Kecheng Liu, and Vaughan Michell. Ontology modeling for generation of clinical pathways. *Journal of Industrial Engineering and Management (JIEM)*, 5(2):442–456, 2012.

[68] RP Jagadeesh Chandra Bose and Wil MP Van der Aalst. Abstractions in process mining: A taxonomy of patterns. In *International Conference on Business Process Management*, pages 159–175. Springer, 2009.

[69] Maikel Leemans and Wil MP van der Aalst. Discovery of frequent episodes in event logs. In *International Symposium on Data-Driven Process Discovery and Analysis*, pages 1–31. Springer, 2014.

[70] Niek Tax, Natalia Sidorova, Reinder Haakma, and Wil MP van der Aalst. Mining local process models. *Journal of Innovation in Digital Ecosystems*, 3(2):183–196, 2016.

[71] Felix Mannhardt and Niek Tax. Unsupervised event abstraction using pattern abstraction and local process models. *Working Conference on Enabling Business Transformation by Business Process Modeling, Development, and Support*, pages 55–63, 2017.

[72] Minseok Song, Christian W Günther, and Wil MP Van der Aalst. Trace clustering in process mining. In *International Conference on Business Process Management*, pages 109–120. Springer, 2008.

[73] RP Jagadeesh Chandra Bose and Wil MP Van der Aalst. Context aware trace clustering: Towards improving process mining results. In *Proceedings of the International Conference on Data Mining Society for Industrial and Applied Mathematics (SIAM)*, pages 401–412. SIAM, 2009.

[74] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[75] James H Martin and Daniel Jurafsky. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Pearson/Prentice Hall Upper Saddle River, 2009.

[76] Dan Jurafsky and James Martin. *Speech and language processing*, volume 3. Pearson London, 2014.

[77] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[78] Luis Javier Rodríguez and Inés Torres. Comparative study of the baum-welch and viterbi training algorithms applied to read and spontaneous speech recognition. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 847–857. Springer, 2003.

[79] Edward ML Peters, Guido Dedene, and Jonas Poelmans. Empirical discovery of potential value leaks in processes by means of formal concept analysis. In *2013 IEEE 13th International Conference on Data Mining Workshops (ICDMW)*, pages 433–439. IEEE, 2013.

[80] Jonas Poelmans, Guido Dedene, Gerda Verheyden, Herman Van Der Mussele, Stijn Viaene, and Edward Peters. Combining business process and data discovery techniques for analyzing and improving integrated care pathways. In *Industrial Conference on Data Mining*, pages 505–517. Springer, 2010.

[81] Berny Carrera and Jae-Yoon Jung. Constructing probabilistic process models based on hidden markov models for resource allocation. In *International Conference on Business Process Management*, pages 477–488. Springer, 2014.

[82] Anne Rozinat, Manuela Veloso, and Wil MP Van Der Aalst. Using hidden markov models to evaluate the quality of discovered process models. *Extended Version. BPM Center Report BPM-08-10, BPMcenter. org*, 161:178–182, 2008.

[83] Haytham Elghazel, Veronique Deslandres, Kassem Kallel, and Alain Dussauchoy. Clinical pathway analysis using graph-based approach and markov models. In *2007 2nd International Conference on Digital Information Management*, volume 1, pages 279–284. IEEE, 2007.

[84] Gil Aires Da Silva and Diogo R Ferreira. Applying hidden markov models to process mining. *Sistemas e Tecnologias de Informação. AISTI/FEUP/UPF*, pages 207–210, 2009.

[85] Haytham Elghazel, Véronique Deslandres, Mohand-Said Hacid, Alain Dussauchoy, and Hamamache Kheddouci. A new clustering approach for symbolic data and its validation: Application to the healthcare data. In *International Symposium on Methodologies for Intelligent Systems*, pages 473–482. Springer, 2006.

[86] Ghazaleh Khodabandelou, Charlotte Hug, and Camille Salinesi. A novel approach to process mining: Intentional process models discovery. In *8th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–12. IEEE, 2014.

[87] Ghazaleh Khodabandelou, Charlotte Hug, Rebecca Deneckere, and Camille Salinesi. Supervised vs. unsupervised learning for intentional process model discovery. In *Enterprise, Business-Process and Information Systems Modeling*, pages 215–229. Springer, 2014.

[88] Jens Meier, Andreas Dietz, Andreas Boehm, and Thomas Neumuth. Predicting treatment process steps from events. *Journal of Biomedical Informatics*, 53:308–319, 2015.

[89] Xiang Li, Haifeng Liu, Shilei Zhang, Jing Mei, Guotong Xie, Yiqin Yu, Jing Li, and Geetika T Lakshmanan. Automatic variance analysis of multistage care pathways. *Studies in Health Technology and Informatics*, 205:715–719, 2014.

[90] Anne Rozinat, M Veloso, and Wil MP Van der Aalst. Evaluating the quality of discovered process models. In *2nd Intl. Workshop on the Induction of Process Models, Antwerp, Belgium*, pages 45–52. Citeseer, 2008.

[91] Joos CAM Buijs, Boudewijn F Van Dongen, and Wil MP van Der Aalst. On the role of fitness, precision, generalization and simplicity in process discovery. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 305–322. Springer, 2012.

[92] Arya Adriansyah, Boudewijn F van Dongen, and Wil MP van der Aalst. Conformance checking using cost-based fitness analysis. In *15th IEEE International Enterprise Distributed Object Computing Conference (EDOC)*, pages 55–64. IEEE, 2011.

[93] Wil Van der Aalst, Arya Adriansyah, and Boudewijn van Dongen. Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(2):182–192, 2012.

[94] Jan Mendling. *Detection and prediction of errors in EPC business process models.* PhD thesis, Wirtschaftsuniversität Wien Vienna, 2007.

[95] Angelina Kurniati, Eric Rojas, David Hogg, Geoff Hall, and Owen Johnson. The assessment of data quality issues for process mining in healthcare using medical information mart for intensive care iii, a freely available e-health record database. *Health Informatics Journal*, pages 1–16, 2018.

[96] Samantha Syke, Sarah Kingsbury, Philip Conaghan, Mar Rodriguez, Paul Baxter, and Owen Johnson. A process mining approach to discovering cardiovascular disease trajectories. [poster]. Presented in the 29th Annual of Medical Informatics Europe Conference, Gothenburg , Sweden, 2018.

[97] Andreas Wombacher et al. Start time and duration distribution estimation in semi-structured processes. In *Proceedings of the 28th annual ACM Symposium on Applied Computing*, pages 1403–1409. ACM, 2013.

[98] Borja Vázquez-Barreiros, Manuel Mucientes, and Manuel Lama. Mining duplicate tasks from discovered processes. In *ATAED@ Petri Nets/ACSD*, pages 78–82. Citeseer, 2015.

[99] Wil MP Van der Aalst, Vladimir Rubin, HMW Verbeek, Boudewijn F van Dongen, Ekkart Kindler, and Christian W Günther. Process mining: a two-step approach to balance between underfitting and overfitting. *Software & Systems Modeling*, 9(1):87, 2010.

[100] Seppe Vanden Broucke. *Advances in Process Mining: Artificial Negative Events and Other Techniques*. PhD thesis, KU LEUVEN Economy and Faculty of Business Sciences, 2014.

[101] Xixi Lu, Dirk Fahland, Frank JHM van den Biggelaar, and Wil MP van der Aalst. Handling duplicated tasks in process discovery by refining event labels. In *International Conference on Business Process Management*, pages 90–107. Springer, 2016.

[102] Luís Filipe Nascimento da Silva. Process mining: Application to a case study. Master's thesis, Economics Faculty University of Porto, 2014.

[103] Suriadi Suriadi, Robert Andrews, Arthur HM ter Hofstede, and Moe Thandar Wynn. Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. *Information Systems*, 64:132–150, 2017.

[104] Valery A Petrushin. Hidden markov models: Fundamentals and applications. In *Online Symposium for Electronics Engineer*, 2000.

[105] Satu Helske and Jouni Helske. Using the seqhmm package for mixture hidden markov models. 2015.

[106] Jennifer Pohle, Roland Langrock, Floris van Beest, and Niels Martin Schmidt. Selecting the number of states in hidden markov models-pitfalls, practical challenges and pragmatic solutions. *Journal of Agricultural Biological and Environmental Statistics, In press, , Obtained from https://arxiv.org/abs/1701.08673*, 2017.

[107] Silvia Bacci, Silvia Pandolfi, and Fulvia Pennoni. A comparison of some criteria for states selection in the latent markov model for longitudinal data. *Advances in Data Analysis and Classification*, 8(2):125–145, 2014.

[108] Michele Costa and Luca De Angelis. Model selection in hidden markov models: a simulation study. *Dipartimento di Scienze Statistiche*, pages 1689–1695, 2010.

[109] Cen Li and Gautam Biswas. Applying the hidden markov model methodology for unsupervised learning of temporal data. *International Journal of Knowledge Based Intelligent Engineering Systems*, 6(3):152–160, 2002.

[110] Tao Xiang and Shaogang Gong. Beyond tracking: Modelling activity and understanding behaviour. *International Journal of Computer Vision*, 67(1):21–51, 2006.

[111] Karen L Nylund, Tihomir Asparouhov, and Bengt O Muthén. Deciding on the number of classes in latent class analysis and growth mixture modeling: A monte carlo simulation study. *Structural equation modeling: A multidisciplinary Journal*, 14(4):535–569, 2007.

[112] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[113] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[114] Richard Williams. Scalar measures of fit: Pseudo r2 and information measures (aic & bic) (lecture notes). *Retrieved from http://www3.nd.edu/ rwilliam/stats3/L05.pdf*, 2005.

[115] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[116] Gilles Celeux and Jean-Baptiste Durand. Selecting hidden markov model state number with cross-validated likelihood. *Computational Statistics*, 23(4):541–564, 2008.

[117] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2000.

[118] Yongdai Kim, Sunghoon Kwon, and Hosik Choi. Consistent model selection criteria on high dimensions. *Journal of Machine Learning Research*, 13:1037–1057, 2012.

[119] Greg Hamerly and Charles Elkan. Learning the k in k-means. In *Advances in Neural Information Processing Systems*, pages 281–288, 2004.

[120] Maud Delattre, Marc Lavielle, and Marie-Anne Poursat. A note on bic in mixed-effects models. *Electronic Journal of Statistics*, 8(1):456–475, 2014.

[121] Jiahua Chen and Zehua Chen. Extended bic for small-n-large-p sparse glm. *Statistica Sinica*, 22(2):555–574, 2012.

[122] Florian Frommlet, Felix Ruhaltinger, Piotr Twaróg, and Małgorzata Bogdan. Modified versions of bayesian information criterion for genome-wide association studies. *Computational Statistics & Data Analysis*, 56(5):1038–1051, 2012.

[123] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[124] Mark Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):113–128, 2004.

[125] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

[126] Owen Johnson, Peter Hall, and Claire Hulme. Netimis: Dynamic simulation of health economics outcomes using big data. *PharmacoEconomics*, 34(2):107–114, 2016.

[127] Andreas Rogge-Solti, Ronny Mans, Wil MP van der Aalst, and Mathias Weske. *Repairing event logs using stochastic process models*, volume 78. Universitätsverlag Potsdam, 2013.

[128] Yaochu Jin and Bernhard Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):397–415, 2008.

[129] JCAM Buijs. *Flexible evolutionary algorithms for mining structured process models*. PhD thesis, Netherlands. Eindhoven: Technische Universiteit Eindhoven, 2014.

[130] Enrico Coiera. *Guide to health informatics*. CRC press, 2015.

[131] Sandeep Hosangadi. Distance measures for sequences. *arXiv preprint arXiv:1208.5713*, 2012.

[132] Michael Becker and Ralf Laue. A comparative survey of business process similarity measures. *Computers in Industry*, 63(2):148–167, 2012.

[133] Timothy Marler and Jasbir Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41(6):853–862, 2010.

[134] Antony Pittathankal and Tim Davidson. Care pathways for patients with breast cancer. *Trends in Urology, Gynaecology & Sexual Health*, 15(2):10–13, 2010.

[135] Breast cancer - types of treatment [online]. `https://www.cancer.net/cancer-types/breast-cancer/types-treatment` [Accessed: 18 April 2019].

[136] Gert Janssenswillen and Benoît Depaire. Bupar: business process analysis in r. In *CEUR Workshop Proceedings*, volume 1920, pages 2–6, 2017.

[137] Adriano Augusto, Raffaele Conforti, Marlon Dumas, and Marcello La Rosa. Split miner: Discovering accurate and simple business process models from event logs. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1–10. IEEE, 2017.

[138] Jorge Munoz-Gama. *Conformance Checking and Diagnosis in Process Mining: Comparing Observed and Modeled Processes*, volume 270. Springer, 2016.

[139] Icd9/icd9cm codes [online]. `http://icd9cm.chrisendres.com/` [Accessed: 30 June 2017].