Evaluation of Process Mining Techniques for Modeling In-Hospital Patient Care Pathways Using the MIMIC-III Dataset

Anita Klementiev

Master of Science

Cognitive Science

School of Informatics

University of Edinburgh

2020

Abstract

Process mining is a set of techniques to uncover, define and validate process flows from event logs. The main purpose of this project is to explore the usefulness of process mining in healthcare settings using electronic healthcare records. Insights gained are being shared with collaborators developing DataLoch, a resource for data-driven innovation in health and social care in South East Scotland. To achieve these insights, different process mining approaches are implemented using the publicly available MIMIC-III dataset and their usefulness is evaluated through a series of interviews with the DataLoch clinical team. Specifically, the techniques implemented are: the PM4Py Python library for large-scale data science applications of process mining; the user-friendly graphical interface of Disco; and a single-pathway approach to process analysis proposed in this report.

Acknowledgements

I would like to thank my two supervisors, Catherine Stables and Jacques Fleuriot, for their ceaseless support, crucial advice and earnest enthusiasm for the project. I would also like to thank the DataLoch team, in particular Atul Anand and Colan Mehaffey. Additionally, I am grateful for the support group I found in the Artificial Intelligence Modelling Lab (AIML). Finally, I would like to thank my family and friends for encouraging me and keeping me company on my MSc journey.

Table of Contents

Chapter 1

Introduction

1.1 Project Overview

Process mining [22] is a set of techniques to uncover, define and validate process flows from event logs. Process mining can be applied to any dataset which contains timestamped activities or events. A model which defines the ordering of events in a process can be extracted using a one of several process mining algorithms. Process models can be built for a wide range of processes in fields spanning business, finance, manufacturing and healthcare among others. Analysis of the resulting process models can give insights which can be used to identify workflow patterns, optimise use of resources and pinpoint inefficiency trends in the execution of the process.

Process mining has been applied to healthcare processes successfully on several occasions [4, 6, 19]. However, despite the accuracy of the models which recent process mining techniques are known to give, there are many challenges when it comes to adopting this technology in real-life healthcare environments. There are many different process mining algorithms and tools, and it is not always clear which should be used and for what purpose.

This project was conducted to help inform the potential implementation of process mining into Dataloch [7], a data repository that is being developed to facilitate data-driven innovation in Edinburgh and South East Scotland. DataLoch aims to create a comprehensive ecosystem of health data across the region by linking multiple data assets from primary, secondary and social care. Given the wealth of data assets which will make up DataLoch, as shown in Figure 1.1, I speculate that process mining techniques have the potential to provide abundant insights about a wide range of healthcare processes occurring in the region.

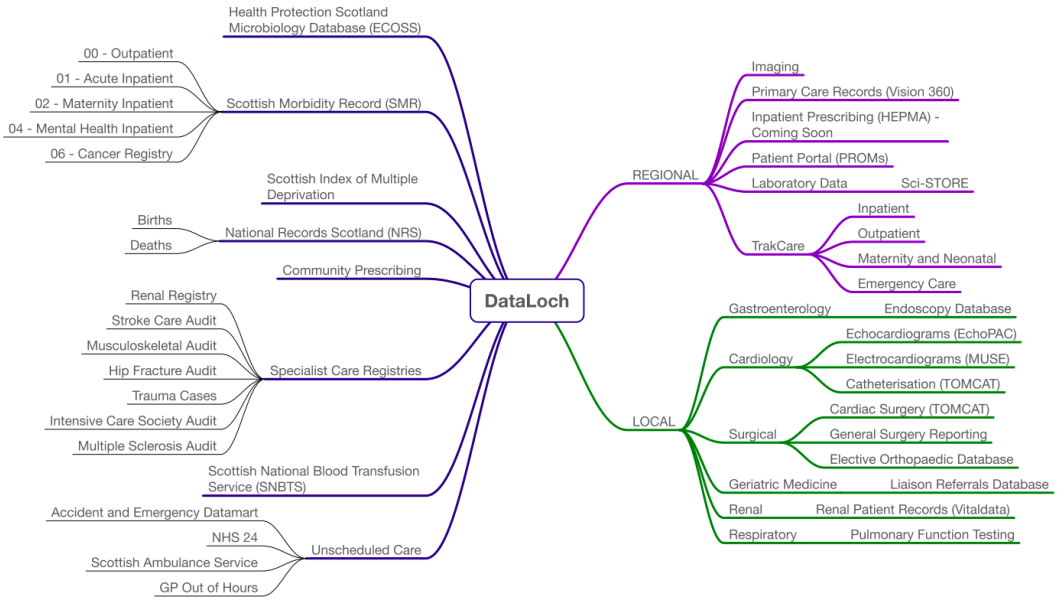


Figure 1.1: Examples of data which will be stored in the DataLoch repository for singular access.

In order to explore the possibilities of applications of process mining to DataLoch, I am using the open source Medical Information Mart for Intensive Care III (MIMIC III) database [10] in conjunction with a selection of tools and software. Process mining has only been applied to this dataset once before, in a case study by Kurniati et al. [4] modelling admission and discharge of patients in oncology. The tools and software which will be evaluated are:

- **PM4Py** [2]: a process mining Python library developed to help researchers mine processes from large datasets and easily control the parameters of the process mining algorithms. The two process mining algorithms which will be applied are:
  - **InductiveMiner** [12]: a popular algorithm known for modelling event log behaviour with near-perfect accuracy.
  - **HeuristicMiner** [23]: an algorithm often used when mining healthcare processes, known for representing the process behaviour more generally by using probabilistic methods.
- **Disco** [21]: licenced process mining software with a graphic user interface and multiple extensive filtering options.

- **Custom Generated Models of Individual Path Variants**: an approach implemented in this project aiming to help visualise the process mining outputs in a simplistic and easy-to-read manner.

The aim of this project is to use above approaches and the MIMIC III database to identify questions about patient pathways through intensive care which can reasonably be answered using process mining and evaluate the suitability of the different techniques for use by clinical and management staff in hospitals.

1.2 Research Questions

The main questions which I aim to answer in this report are are:

- What kinds of questions can process mining help answer with the use of DataLoch or a similar dataset?
- How successfully can similar questions be solved using the the MIMIC III database?
- What process mining tools are best suited to solving these questions?
- What representations of process models are best for answering these questions?
- What is the trade-off between maximising quantitative performance metrics of the process models and potential usability of the models?
- How useful do clinicians find the process models generated through process mining to be?

1.3 Contributions

The main contributions made by this body of work are:

- A PM4Py replication of the pathway models for cancer patients in MIMIC III created by Kurniati et al. [4], and a comparison of the original models to those generated by InductiveMiner and HeuristicMiner.
- Analysis of pathways of patients through the coronary care unit (CCU) in MIMIC III using PM4Py implementations of InductiveMiner and HeuristicMiner. In particular, the following patient groups are extracted and their pathways are explored.

  - Patients who stayed in CCU for less than a day.
  - Patients who stayed in CCU for over a week.
  - Patients who suffered a myocardial infarction.

- An overview of potential use-cases of process mining with data from hospital information systems compiled from conversations around different process models with the DataLoch clinical team.

- Evaluation of process models and tools such as Disco, with feedback from clinical staff on how useful they find the various approaches to process mining listed above.

- Isolation and statistical analysis of individual paths.

  - An implementation of PM4Py to extract patient paths and represent them in a readable way.
  - Graphical representations of characteristics of patients following different paths which aim to give insight to what patient attributes are correlated with certain pathways.
  - A prototype model of tool which could be used for such analysis.

1.4 Report Structure

In chapter 2, I will give an introduction to process mining and the process mining discovery algorithms used in this project. I will also examine published applications of process mining with healthcare data and the challenges that arise when using process mining techniques in this context. In chapter 3, I will describe the dataset, tools and metrics I used to build and evaluate the process models described above. In chapter 4, the resulting models will be presented. I will compare the models based on quanitiative performance metrics, and also outline remarks and observations from clinical staff at DataLoch when they were presented with these models. Finally, chapter 5 will address how well the models answered the questions they were built to solve as well as criticisms from the clincal team. I will present possible ways to move forward in order to ensure successful applications of process mining in this field.

Chapter 2

Background

2.1 Process Mining

Process mining [22] is a set of techniques for extracting knowledge from data generated and stored in information systems in order to analyse executed processes. Process mining can be used to model, monitor and improve operational processes in a wide range of fields including business, healthcare, manufacturing and finance. The information extracted from event logs using process mining can reflect the reality of the processes within an organisation. Although an organisation might have perceptions of how processes are ideally executed, process mining can uncover inefficiencies, bottlenecks and non-conformance issues which are not otherwise evident. Event logs contain timestamped activities (or events) associated with a case ID e.g. a patient, an order from a retail website or a helpline ticket. Process mining organises the event log into traces, which are time-ordered lists of activities corresponding to unique case IDs. Through various algorithms, process mining extracts probable transitions and dependencies between activities and uses them to build a model of the process. The three main applications of process mining are [19]:

- **Discovery:** the use of process mining algorithms to extract models which accurately model the process behaviour in the log.
- **Conformance:** the process of analysing how well behaviour in an event log conforms to guidelines or expectations using a process model.
- **Extension:** the use of a combination of both discovery and conformance techniques to project log information onto a process model in order to improve it.

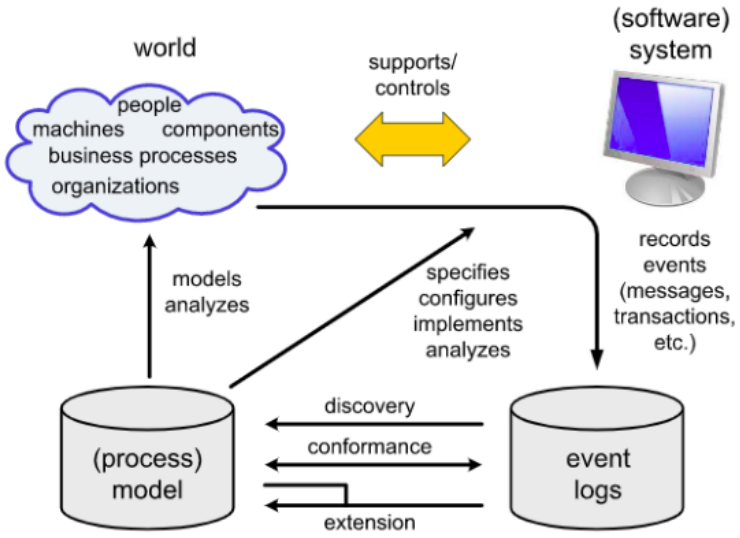


Figure 2.1: The process mining framework [6]. Information about real-world behaviours is stored in an information system. This information is used to build an event log, an organised information structure made up of cases and sequences of events corresponding to each case. Process mining uses discovery and conformance algorithms to create, check and extend process models.

2.2 Process mining algorithms

The process models which will be built in this body of work will be in the form of Petri nets, also known as Place/Transition nets [14]. Petri nets represent a set of states (drawn as circles) and transitions between states (drawn as rectangles), which represent events and activities in process mining. A Petri net consists of

- a set of places P
- a set of transitions T, where $P \cap T = \emptyset$
- the *flow relation*, a set of arcs $F \subset (P \times T) \cup (T \times P)$

The two process discovery algorithms investigated in this report are InductiveMiner and HeuristicsMiner. InductiveMiner was chosen as it is currently the most used process mining algorithm and can handle invisible tasks and generates a sound model. HeuristicsMiner is chosen because it is the most commonly used algorithm identified in the 2016 literature review of process mining in healthcare conducted by Rojas et al. [6]. Heuristics miner generates robust models and is known for dealing well with noisy data. The HeuristicMiner does not guarantee a sound model.

InductiveMiner The InductiveMiner [12] is one of the most common process discovery algorithms used today, particularly in mining business processes. The InductiveMiner builds *process trees* which can then be converted to a Petri net representation. There are six kinds of *nodes* in an InductiveMiner:

- **Sequence:** The child activities of must be executed one-by-one in order.
- **Exclusive choice:** Exactly one of the activities stemming from an exclusive choice node must be executed.
- **Interleaved:** All activities stemming from an interleaved node must be executed with no overlap.
- **Parallelism:** All activities stemming from a concurrent node mus be executed, and they are allowed to overlap.
- **Or choice:** One or more children of an or node must be executed and can be executed concurrently.
- **Loop:** The first child of a loop node must be executed. If the second activity stemming from the node occurs, the first child is executed again.

The InductiveMiner algorithm builds but these nodes by recursively splitting event logs into sub-logs until the model is complete, guaranteeing a sound model which very accurately represents event log behaviour. Figure 2.2 shows how the nodes of a process tree can be translated to a Petri net. It is evident that the InductiveMiner makes extensive use of hidden transitions in a Petri net, which is how it accounts for skipped and looped tasks.

HeuristicMiner The HeuristicMiner [23] builds a Petri net directly and aims to represent the general behaviour of an event log, not necessarily capturing details and exceptions. HeuristicMiner discovers the following transitions:

- $a > b$: Task a directly precedes task b.
- $a \rightarrow b$: A sequential transition from task a to task b.
- $a \mid\mid b$: Tasks a and b occur in parallel.
- $a \# b$: Tasks a and b never follow each other directly and do not occur in parallel.

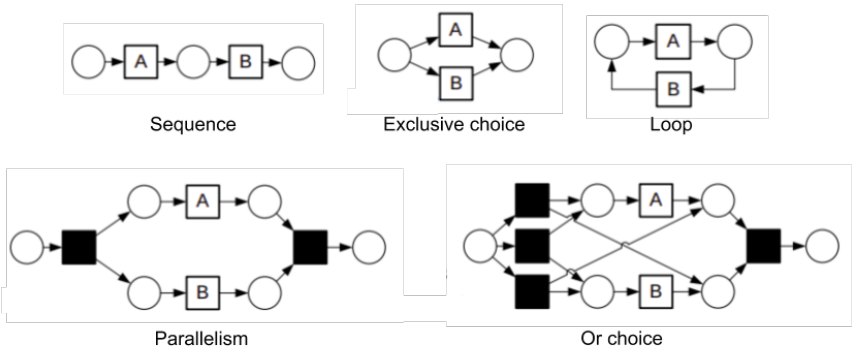


Figure 2.2: Representations of sequence, exclusive choice, parallel, or choice and loop splits in a process tree as Petri nets.

- $a >> b$: This represents a loop of length 1.
- $a >>> b$: This represents a loop of length 2.

The feature of HeuriticMiner which sets it apart from other discovery algorithms is its consideration of frequency. The dependency relation $a \Rightarrow_W b$ is a frequency based metric indicating the certainty with which we can say there is a direct dependency relation between tasks a and b using the following equation.

$$a \Rightarrow_W b = \frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \tag{2.1}$$

The Heuristics Miner builds a table of transition probabilities using dependency relation metrics and searches for the highest probable paths to generate a Petri-net which most accurately reflects the behaviour of processes in the event log at a high level.

Other common process mining algorithms used in the healthcare field are FuzzyMiner [8], which generates several process models in different levels of detail, and Trace Clustering [20], which clusters similar processes to build multiple models for one event log.

2.3 Applications of Process Mining in Healthcare

2.3.1 Healthcare processes

There are two kinds of process types which can be examined through process mining in the context of healthcare [17]. These are:

- **Medical treatment processes:** These processes describe observation, reasoning and action pertaining to a patient's treatment. Medical treatment process are directly linked to individual patients and depend on *cases-specific* decisions. Medical processes are also known as clinical processes.

- **Organisational processes:** These processes coordinate treatment collaboratively between people and units. These process are not tailored to specific patients. These processes can represent transfer of information between professionals, for example, the process of scheduling a patient. Organisational processes are also known as administrative processes.

Healthcare processes have characteristics that set them apart from other business processes where process mining has been applied. This often causes process models discovered from healthcare information systems to be less straightforward than those modelling less variable processes in other fields such as sales. Healthcare processes are [6, 17]:

- **Dynamic**: Several variables pertaining to healthcare processes change over time. New techniques and drugs are continually introduced and medical knowledge on diseases and treatments is rapidly expanding.

- **Complex**: Large volumes of data are involved in each case, and this data informs decisions about treatment for each individual patient. Outcomes are usually hard to predict, and obstacles commonly arise.

- **Multi-disciplinary**: Healthcare processes are highly collaborative between professionals and units. Repeated exchange of information between parties amplifies the complexity of the data.

- **Ad-hoc**: Deviation from guidelines are more of a norm than an exception. In each case, treatment is highly dependant on qualities of the patient, such as past complications. Additionally, patients have autonomy and nearly always also have control in the process.

- **Non-repetitive**: On account of the above characteristics, care pathways are not shared by patients as often as cases in other kinds of processes. The more fine-grained the description of the process (e.g. including procedures, lab test events, consultations), the less repetitive the process behaviour.

- **Non-deterministic**: Order of events in healthcare processes is often non-deterministic. Different approaches to treatment can be tried, and the order of these is dependent on clinical judgement of the patient's condition.

2.3.2 Data types

For the purpose of process mining, Mans et al. [16] divide healthcare data types into four categories based on their source: administrative systems, clinical support systems, healthcare logistic systems and medical devices. The type of data used determines what sort of process is extracted during process modelling. Figure 2.3 plots the four kinds of healthcare data on a spectrum composed of 'abstraction' and 'accuracy' metrics.

Administrative systems, which deal with management and billing, operate on a high level of abstraction because the refer to groups of tasks, rather than individual tasks. Medical devices operate on a low level of abstraction because they refer to subcomponent parts of tasks. Clinical support systems and healthcare logistic systems refer to individual tasks in a process, such as taking a measurement (clinical) or making an appointment (logistic), and are therefore on a mid-level of abstraction. The tasks in these two systems are typically entered manually.

Accuracy refers to a combination of the following three components: *granularity* of the timestamp, *correctness* of the data point given the granularity and *directness of registration*, referring to whether the data point is input manually or automatically. Administrative systems are classed as 'low-accuracy' because the data is usually entered into the system by hand and has low granularity (e.g. only referring to the date of the task, omitting time). Clinical system data is typically also entered manually, but has higher granularity because the minute/second at which the task was executed is significant in clinical processes. On account of this, clinical systems are said to have 'medium-accuracy.' Logistics systems have medium to high accuracy as well because they have high directness and correctness, but can vary in granularity. Medical devices have the highest accuracy because the data is imputed to the system automatically and accurate to the millisecond.

2.3.3 Frequently posed questions

Mans et al. [16] identify four overarching questions frequently posed in studies of process mining with healthcare data:

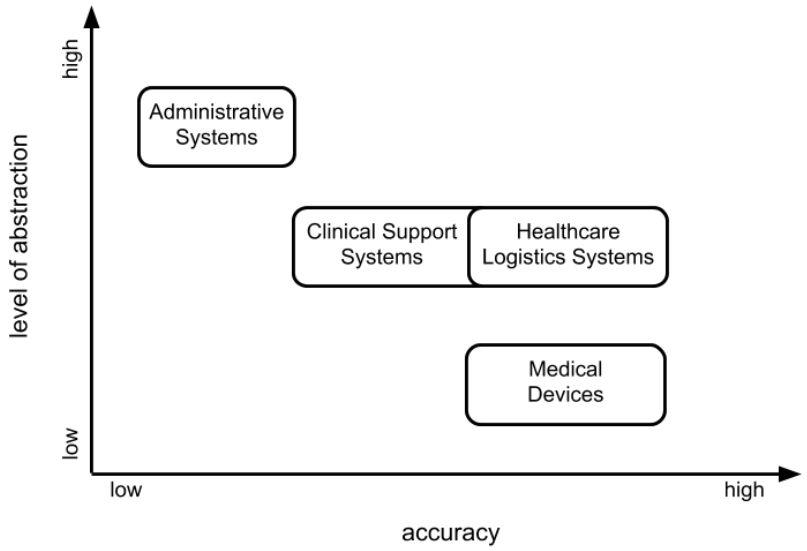


Figure 2.3: Spectrum of healthcare information system data types [16]. "Accuracy" measures are composed of granularity, directness and correctness dimensions.

1. What are the most followed paths and what exceptional paths are followed?
2. Are there differences in care paths followed by different patient groups?
3. Do we comply with internal and external guidelines?
4. Where are the bottlenecks in the process?

Administrative system data is commonly used to answer any of the above questions using process modelling. However, since the granularity for the timestamps in these systems is typically low, several complications arise. If the finest granularity of event timestamps is the date of the activity, the discovery algorithm might conclude that events occurring on the same day occur in parallel when in reality they have an ordered relationship. Therefore, most event logs used for process mining use data from a combination of the four types of systems.

Rojas et al. [18] identify four more abstract question categories for healthcare process mining:

1. What are the activities in a process?
2. What are the causes different executions of processes?

3. What are the circumstances in which an activity will take place? How can this information be used to predict how a process will play out?

4. What are the ways in which a process can be improved?

All four of the above question can be answered on any level of granularity, from broad to specific processes. The datatypes used are therefore specific to the subordinate question.

2.3.4 Challenges in process mining in healthcare

There are several challenges which arise when applying process mining to healthcare which are less common in other applications of process mining. Firstly, there is an absence of good visualisation of models and results. As seen in Figure 2.2, algorithms like InductiveMiner, although their output models represent event stream behaviour very accurately, use complex Petri net representations made up of several components to represent just one transition. When several of these transitions are combined to show a process, the model quickly becomes too complex to easily read.

Most applications of process mining in healthcare in literature [6] are specific to a singular problem or question being answered. The studies also use data and data models which are specific to a singular institution and the systems used for storing the information. This results in the techniques being hard to scale for implementation across medical centres.

Finally, applications of process mining in this field highly rely on process mining experts. Process mining is difficult to apply without knowledge of data types, tools and algorithms needed to execute it. Most hospitals do not have the incentive to spend resources on building process mining tools specifically for their organisation.

The implementations of process mining built in this project will address these challenges and propose ways in which to make process mining techniques more saleable regardless of hospital or database. In process mining, it is important to choose data types and discovery algorithms carefully in order to create a model of a process which is not only accurate, but also practical and usable. The following chapters will describe three different software approaches to process mining using healthcare data and explore the insights they provide about pathways of patients who stayed in the coronary care unit of a hospital. The evaluation of these implementations will focus on determining the optimal combination of process mining software, data types, discov-

ery algorithms and visualisation techniques discussed in this chapter in order to answer questions about the patients visiting coronary care and their pathways.

Chapter 3

Methodology

3.1 Dataset

3.1.1 MIMIC III

Medical Information Mart for Intensive Care III [10] is an open source database of information stored on MetaVision and CareVue systems at the Beth Israel Deaconess Medical Center in Boston, MA (USA). It contains de-identified data on over forty thousand patients. The data ranges from administrative, such as admission data, insurance type, to medical device data including vital-signs monitoring and blood test results. In this project, the version used is MIMIC-III v1.4, from September, 2016.

Of the 26 tables in MIMIC III, only 16 contain timestamped data. Kurniati et al. explore 16 of these tables thoroughly in their application of process mining using this data [4]. They settle on using data from the tables `ADMISSIONS`, containing information about patient admission, discharge and death, and `ICU_STAYS`, containing information about individual intensive care unit (ICU) stays. They used the `DIAGNOSES` table to help filter out cancer patients by ICD9 code [1]. In this project, I also consider the tables:

- `LABEVENTS`: laboratory measurements, including out patient data.
- `MICROBIOLOGYEVENTS`: microbiology information, including cultures acquired and associated sensitivities.
- `PROCEDUREEVENTS_MV`: procedures logged in the MetaVision ICU database.
- `TRANSFERS`: physical locations for patients throughout their hospital stay.

Process models were created with each of the above tables, some resulting in large models which were built on large sets of data, e.g. the LABEEVENTS table which contans over 27 million timestamped data points. The models which will be discussed in detail in the Results and Analysis chapter (chapter 4) are all based on data from only ADMISSIONS, ICU_STAYS and TRANSFERS, a collection of 382,465 timestamped event data points.

Given that MIMIC III is specifically an ICU database, the data has some limitations. The timestamped data contains few events that happen outside of intensive care. This results in gaps in the knowledge about patient pathways if a patient was moved outside of ICU. In the TRANSFERS table, if a patient was moved to or from a non-ICU ward, the WARD_ID column is null. For the process models created for this report, these activites are given the label "non-ICU ward."

3.1.2 Data Preprocessing

The data point used as the case ID in all processes mined in this project is 'HADM_ID', the unique hospital stay identifier. In order to filter patients by diagnosis, such as cancer or myocardial infraction, their subjects were isolated by filtering the DIAGNOSES table by the desired ICD9 code or range of codes. Then the activity tables were filtered to contain only SUBJECT_IDs which appeared in the filtered DIAGNOSES table.

Filtering on variants was done after the event logs were formed but before process discovery algorithms were applied. These filtered groups include

- Patients who followed one of the top 5 most commonly appearing paths;
- Patients who stayed in a specific care unit during their stay, e.g. CCU;
- Patients whose pathway contained one event node eventually followed by some other specified activity, e.g. patients who stayed in CCU who were eventually transferred to a surgical intensive care unit (SICU); and
- Patients who stayed in a care unit for a certain amount of time, e.g. patients who stayed in CCU over a week.

3.2 Applications of process mining to the MIMIC III dataset

Given the limitations of the MIMIC III dataset, we identified a number of questions which could potentially be answered using this dataset in conjunction with process

mining, and fit into the overarching frequently posed questions described in subsection 2.3.3. These questions were agreed on from an early meeting with DataLoch clinical staff for an initial gauge of process models created with MIMIC III.

- Are there any patients currently admitted which are on an unusual path and is it problematic?
- Who are patients that spend an unusually long time in CCU (or any individual intensive care unit) and what are their pathways?
- Who are patients that spend a short time in a care unit ($<$1 day) and what are their pathways?
- What differentiates pathways followed by patients with a certain acute condition, for example myocardial infarction, from all other patient pathways.
- What differentiates pathways followed by patients with a certain chronic condition, for example chronic ischaemic heart disease, from all other patient pathways.
- What are different paths through a certain surgery and the death rate for patients following different paths?
- What are characteristics of patients whose condition got worse causing them to move from a non-ICU ward to CCU?

Although all of the above questions can be solved with process mining, nearly all require additional computations/statistical analysis to answer fully.

3.3 Tools

The following three approaches to process mining will be applied to the MIMIC III dataset and their performance will be compared in chapter 4.

PM4Py Although plug-in open source tools such as ProM are often used in process mining research, the primary tool used for these experiments was the Process Mining for Python (PM4Py) [2] library. With PM4Py, it is easy to fine-tune discovery algorithms and their parameters, while toolkits like ProM offer limited limited support for algorithmic customisation. PM4Py's specific implementation of the InductiveMiner

Directly Follows algorithm (referred to in PM4Py as IMDFc) was used for process discovery, as was its HeuristicMiner package. PM4Py was also used to generate Petri net visualisations of these process models and for calculating fitness, simplicity, precision and generalisation. Finally, the PM4Py `graphs_visualiser` module was used for graphing distribution of numeric values.

Disco Disco is an alternative licensed tool used for process mining, with several case examples of use with in healthcare. It is created by the company Fluxicon. The tool has a friendly visual interface and multiple extensive filtering options. Disco uses its own specially designed discovery algorithm.

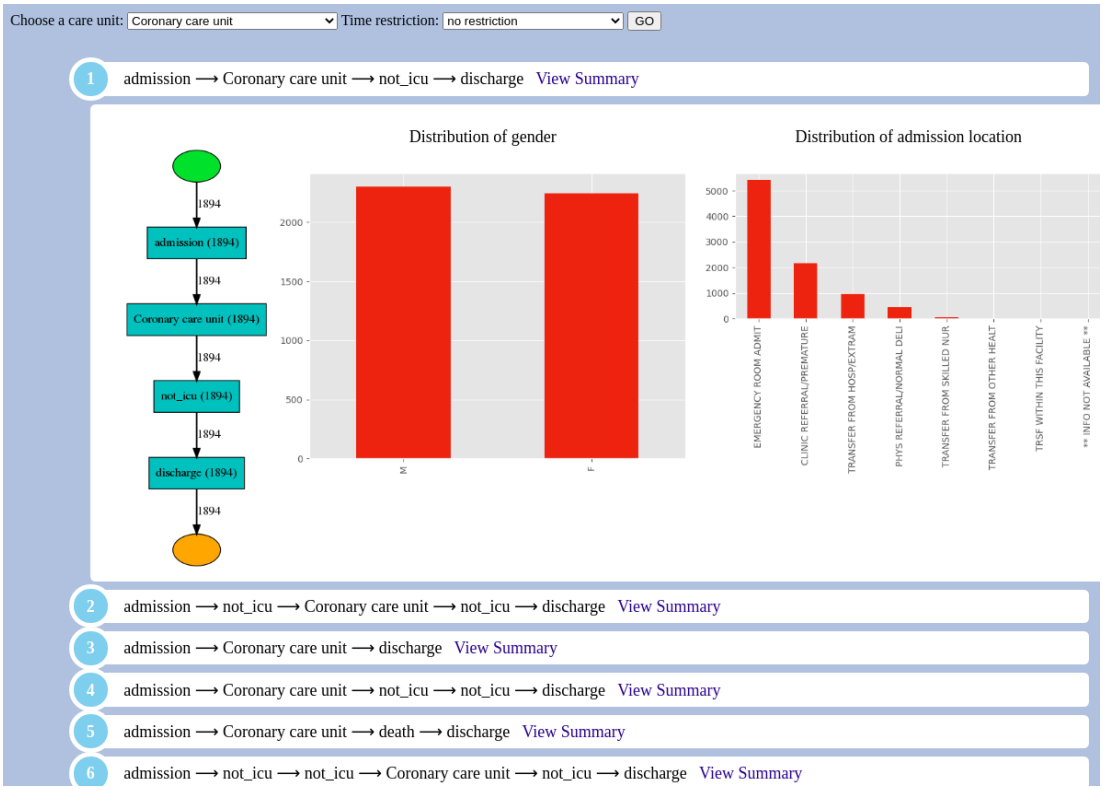


Figure 3.1: Screenshot for prototype pathway analysis tool. Using the dropdown menus at the top, the user can filter by care unit and length of stay in that care unit. The system returns the top pathways for each pathway and shows distributions of various characteristics among patients who followed that path. In this example, bar charts for gender and locations from which the patients were admitted are shown.

Extraction of individual process variants for isolated analysis The final process mining technique evaluated was a method for examining individual variants of pathways. A prototype user interface was created for this using the Python Flask Framework. The user can filter the top ten pathways for patients travelling through a certain care unit. These pathways are represented as singular flowcharts and various statistics about the patient groups for each pathway are calculated such as top diagnoses and mean and standard deviation of patient age. A screenshot of this tool is shown in Figure 3.1.

3.4 Evaluation metrics

Simplicity, fitness, precision and generalisation are the four main evaluation metrics in process mining [9]. All four will be calculated for models created in this project,

Simplicity Simplicity quantifies the complexity of a model. In this report, simplicity will be evaluated using the inverse arc degree simplicity metric [5]. This metric is chosen because it is designed for use with Petri nets, as opposed to other simplicity metrics designed for process trees [9]. To calculate the degree, we inversely examine the ratio of places in the Petri net to the number of possible transitions in and out of those nodes. A model with fewer transitions per activity results in a higher simplicity metric. Inverse arc simplicity ranges from 0 to 1 and is calculated using the following formula:

$$\text{Simplicity} = \frac{1}{1 + \frac{|\text{\# places}|}{|\text{\# transitions}|}}$$

Replay fitness Replay fitness quantifies how well a model can reproduce the behaviour exhibited in the event log. An alignment-based technique is used to evaluate the fitness of models in this report. The formula attempts to align traces to the model, assigning penalty costs for inserted and deleted (skipped) activities. These costs are divided by the minimal costs for treating all events as inserted activities (a scenario where the event log does not match the model) to normalise the resulting value. The equation is written as:

$$\text{Fitness} = 1 - \frac{\sum_{a \in A_s} k^s(a) + \sum_{e \in E_i} k^i(e)}{\sum_{e \in E_c} k^i(e)}$$

In the above equation, A_s represents the set of skipped activities, while the set $E_i \subseteq E_c$ represents inserted events. k^i and k^s represent cost of insertion and skipping, respec-

tively. In this report, we will examine the number of pathways that completely fit the model as well as the mean fitness value for individual pathways.

Precision Precision determines how much behaviour not observed in the event log (unacceptable behaviour) is allowed by the model. Higher precision models are less likely to underfit. The equation used is take from Muñoz-Gama et al. [13].

$$\text{Precision} = 1 - \frac{\sum_{e \in EE} e}{\sum_{t \in AT} t}$$

In this formula, AT stands for allowed transitions and EE stands for escaping edges, i.e. transitions allowed by the Petri net but not reflected in the log.

Generalisation The final metric is generalisation, which measures the models ability to model acceptable behaviour it has not yet seen in the event log. Models with higher generalisation are less likely to overfit. Generalisation is calculated as follows by computing the average number of occurrences of transitions during replay[9]:

$$\text{Generalisation} = 1 - \frac{\sum_{t \in \text{transitions}} \sqrt{\frac{1}{\#\text{ occurrences of } t}}}{|\text{transitions}|}$$

Chapter 4

Results and Analysis

4.1 Introduction

In this section, I will discuss the models created using the various process mining techniques defined in chapter 3. In section 4.3, the only published application of process mining to the MIMIC III dataset will be replicated using PM4Py and the resulting models will be compared. In section 4.4, the scope will be narrowed to examine models of pathways of patients travelling through the coronary care unit (CCU). From the CCU patient event logs, we will closely examine hospital stay processes of two specific patient groups in two case studies described in section 4.5 and section 4.6. Overall comparative results are discussed in section 4.7.

Before addressing the process models created, I will discuss the different data points which were used as event nodes in the models in section 4.2. Quantitative analysis of models is done using the metrics outlined in section 3.4. The resulting models were also shown to the DataLoch clinical team, who gave insights on the usability of the process models and suggested further applications for the techniques presented to them.

4.2 Variants on Activity Events

Admission and ICU Stay Events

The simplest models built replicated the case study in process mining oncology Kurniati et al. [4] and use only events from `ADMISSIONS` and `ICUSTAYS` tables. This resulted in models that generalised well to all patients. A model built using the top 5

path variants using Inductive Miner and accurately represented the paths of 73% percent of patients who were diagnosed with cancer. However, these models do not give much helpful insight to the internal activities or locations in care pathways, they only represent activities such as emergency department (ED) registration, admission to ICU and discharge.

Transfers Events

By adding events from the `TRANSFERS` table, it is possible to model movement between wards. MIMIC III contains data of when patients are moved between 8 different types of ICU cost centres. This data is timestamped for both in-time and out-time of the patient's stay in a given care unit. This information facilitates calculation of wait time between wards. As MIMIC III is an intensive care database, there is no information on what ward a patient is in when they are transferred outside of an ICU cost centre. Since there is still a small number of possible activities when using care unit location data, this means that process models built with this log model still have relatively few path variants and still reflect some commonly repeated paths, although not as broadly as when using 0only `ADMISSIONS` and `ICUSTAYS` data.

Procedures Events

Timestamped procedure data come from the MetaVision system and covers a range of categories such as "Intubation/Extubation", "Ventilation", "Significant Events" and "Imaging." Using all events in this table as activities in the event log results in high variability in paths with nearly all path variants being unique.

4.3 Process Mining in Oncology

The first models I created for this project aimed to replicate the models generated by Kurniati et al. [4] in their paper "Process mining in oncology using the MIMIC-III dataset," currently the only instance of process mining using the MIMIC III. The original paper used the ProM software implementation of Inductive Visual Miner [11], a variation on InductiveMiner, in conjunction with ProM's BPMN Miner plugin [15] to generate the model shown in Figure 4.1. Models generated by InductiveMiner and HeuristicMiner using the PM4Py library are shown in Figure 4.2 and Figure 4.3 respectively.

As can be seen in Figure 4.1, the model generated has little indication of the order of activities. It starts with an optional emergency department (ED) registration event, which is followed by a large `AND` loop containing the rest of the possible activities. This model has a fitness score of 0.971, higher than that of the PM4Py InductiveMiner model and the PM4Py HeuristicMiner model. This model also has a precision score of 0.881. Kurniati et al. do not include generalisation and simplicity scores. The fitness and precision scores for this model are high because the large number of paths which the model accepts on account of the `AND` loop allowing highly variable sequences of events independent of order. However, this means that the model's generalisation score would probably be low, because it allows unacceptable path behaviour, e.g. discharge occurring before admission.

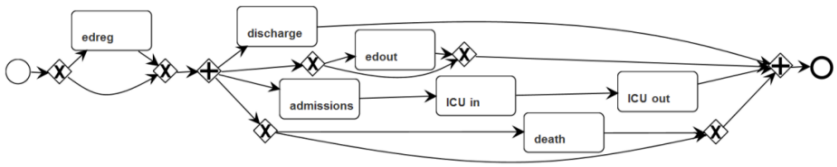


Figure 4.1: Process model for registration and discharge of patients in the study conducted by Kurniati et al. [4] using ProM's BPMN mine in conjunction with Visual InductiveMiner. Fitness: 0.971, Precision: 0.881.

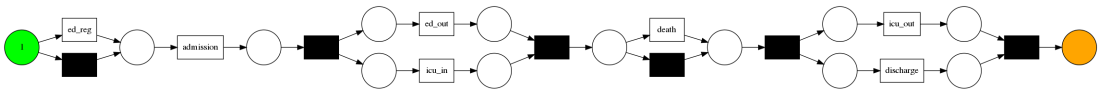


Figure 4.2: The model for the same group of cancer patients as in Figure 4.1 created using PM4Py's implementation of InductiveMiner. Fitness: 0.929, Precision: 0.795.

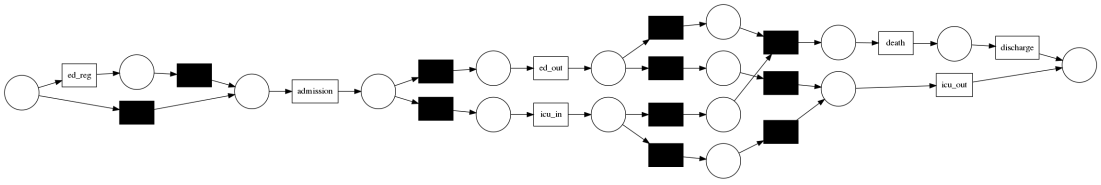


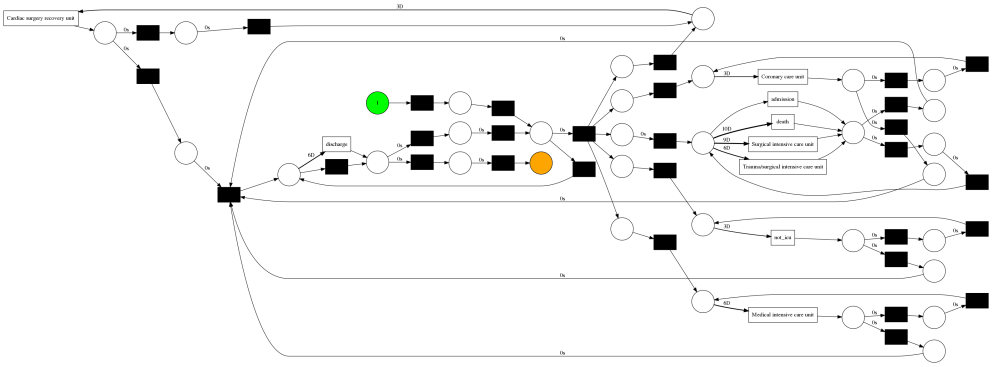
Figure 4.3: The model for the same group of cancer patients as in Figure 4.1 created using PM4Py's implementation of HeuristicMiner. Fitness: 0.635, Precision: 1.0.

As seen in Figure 4.2 and Figure 4.3, the models generated by InductiveMiner and HeuristicMiner with PM4Py dictate order of events more strictly. For example, both

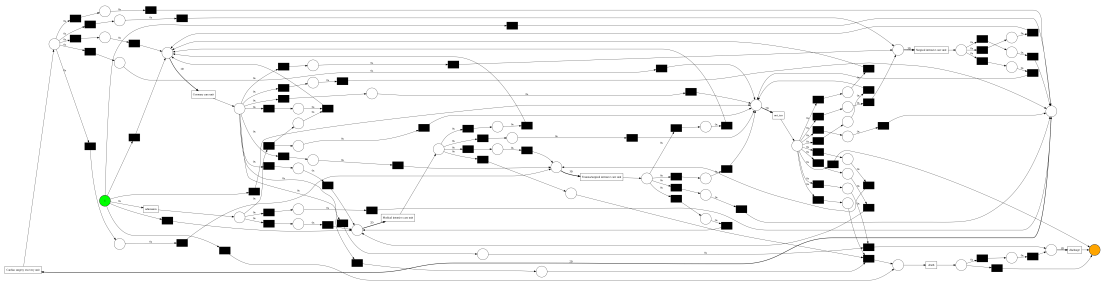
models only accept ED registration and admission at the start of a process and discharge only at the end. The InductiveMiner model has a higher fitness (0.929) than the HeuristicMiner model (0.635). This is because the Inductive Miner Directly Follows algorithm used by PM4Py is know to build high-fitness, sound models, while the HeuristicMiner generalises more by accounting for frequency. Since HeuristicMiner is used often for building models of noisy data, the model contains many hidden transitions, represented by black rectangles in the Petri net, making the visualisation of the model more difficult to interpret. However, the HeuristicMiner's aptitude for representing noise and hidden transitions is what gives it high precision scores, in this case a perfect precision score of 1.0. This inital work demonstrated that I was able to apply InductiveMiner and HeuristicMiner successfully in a pre-defined and previously studied sub-set of the MIMIC III dataset, and established that these different approached produces process models with different orders of events due to differences in process discovery.

4.4 Pathways Through Coronary Care Unit

To explore the usefulness of process mining the in the CCU setting, I first applied InductiveMiner and HeuristicMiner to the subset of patients in the MIMIC III dataset who had been through the CCU. Of 58,976 hospital stays recorded in the dataset, there were 8,288 pathways through the coronary care unit containing a total of 46,749 events from the `ADMISSIONS` and `ICU_STAYS` tables used in the study above, as well as the `TRANSFERS` table to provide more granularity to the process model by including transfers between wards during patients' hospital stays. The models generated by InductiveMiner and HeuristicMiner for the event log for this patient group is shown in Figure 4.4. At this point, I consulted with the clinical experts in DataLoch to defin the types of questions that they would be interested in answering using rocess mining in this CCU setting, which have been outlined in **??**. For this case study, I chose to examine process models of three groups of patients that might be of interest to a care unit manager. The first group are those who were diagnosed with myocardial infarction (MI) (ICD9 code 410.x). This group contained 3,483 patients and 119,036 events. The second group is of patients who visited the CCU but spent less than a day there. This might indicate that this was not the right ward for them to be treated. This group contained 2,525 patients and 15,650 events. The final group of patients who spent over 7 days in CCU, indicating that there were issues with treatment or some kind of delay.



PM4Py InductiveMiner model for patients travelling through CCU.



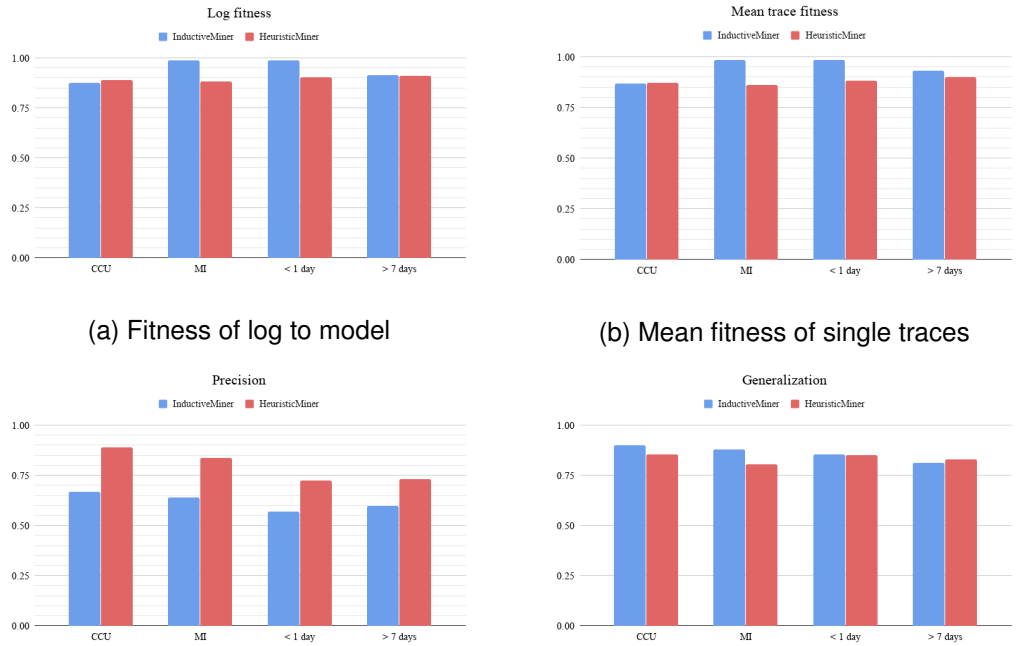
PM4Py HeuristicMiner model for patients travelling through CCU.

Figure 4.4: Models generated for patients travelling through the coronary care unit (CCU). The start point are indicated by the green cells and the end points are indicated by the orange cells. Larger versions of these models can be viewed in Appendix A

This group contained 721 patients and 4,29 events.

To start, I assessed the quality of the models across different scanarios using standard quantitative evaluation methods [9]. Figure 4.5 shows that all models had overall fitness of over 0.88. Additionally, it is shown that InductiveMiner models the traces with better fitness than HeuristicMiner for all four patient groups. This is to be expected, as InductiveMiner is known to generate models with very high, if not perfect, replay fitness.

When evaluating the precision of the model as shown in Figure 4.5, HeuristicMiner outperforms InductiveMiner in all patient groups. This result is not unusual, as InductiveMiner builds models that fit well to the given data, while HeuristicMiner is specifically designed to handle noise. On the other hand, results are mixed for the generalisation metric. values ranged from 0.806 to 0.899, and InductiveMiner had



(a) Fitness of log to model

(b) Mean fitness of single traces

(c) Precision of model

(d) Generalization of model

Figure 4.5: Log fitness, mean trace fitness and precision of models generated by InductiveMiner and HeuristicMiner for the four patient groups in the the case study.

the higher generalisation for all CCU patients, MI-diagnosed patients, and short-stay patients while HeuristicMiner had the higher generalisation for long-stay patients.

Simplicity measures of the generated models ranged from 0.679 to 0.733 (Figure 4.6. The results for this metric are high on account of the limited number of activities which were possible in the event log. This is on account of the fact that the event log was restricted to contain only events from the `admissions`, `icu_stays` and `transfers` tables. These values are much higher than they would be if activities from tables such as `procedure_events` were included, greatly increasing the number of unique activity concepts and complicating the model.

From these results we can conclude that there is minimal difference between the models in terms of technical measures of accuracy, precision and simplicity, and that all model perform well in these metrics. This is in line with other results using InductiveMiner and HeuristicsMiner [2, 6]. However, to asses usefulness in a clinical setting requires more in-depth qualitative analysis. To explore this further, I chose to focus on two specific case studies.

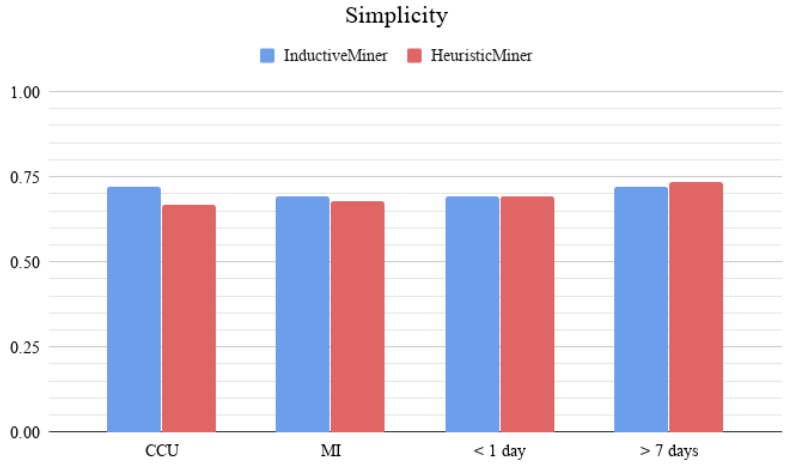


Figure 4.6: Simplicity of generated models

4.5 Case Study: Long CCU Stays

One of the key groups that the DataLoch clinical team identified as being a patient group whose pathways should be analysed was all patients who spent over a week in CCU. Table 4.1 shows the top paths through CCU where the CCU stay was less than 7 days and where the CCU stay was greater than or equal to 7 days. The top variant was the same for both groups of patients (START→admission→CCU→non-ICU ward→discharge→END), and was followed by 23% of patients staying in CCU for less than 7 days and 13% of patients staying in CCU over a week. When these paths were shown to a clinical team, a couple pathways of note were identified. For example, in paths 1-4 in the $\geq$ 7 days group, patients were admitted directly to the coronary care unit, suggesting that the condition was severe at the beginning of their hospital stay. Also, the second most common path for this group includes a death event while there are not deaths in the top five pathways for shorter CCU-stay patients.

Table 4.2 shows the distribution of sex and age for each pathway group. Additionally, it shows the percentage of patients following that path variant with a history of chronic ischaemic heart disease (ICD9 codes 414.8 and 414.9). As expected, ischaemic heart disease is more prevalent in long CCU-stay patients. Additionally, the average age for longer-stay patients is in general higher than for short-stay patients.

In Figure 4.7, we compare the number of patients for the most prevalent diagnoses in the short- and long- stay groups in CCU. It can be identified that respiratory and congestive heart failure are more prevalent in patients who stayed longer in CCU, as well as severe conditions such as cardiogenic shock.

Days in CCU	Rank	Pathway	Cases	Percentage
<7	1	START→admission→CCU→non-ICU ward→discharge→END	1798	23%
	2	START→admission→non-ICU ward→CCU→non-ICU ward→discharge→END	743	10%
	3	START→admission→CCU→discharge→END	455	6%
	4	START→admission→CCU→non-ICU ward→non-ICU ward→discharge→END	374	5%
	5	START→admission→non-ICU ward→non-ICU ward→CCU→non-ICU ward→discharge→END	261	3%
≥ 7	1	START→admission→CCU→non-ICU ward→discharge→END	96	13%
	2	START→admission→CCU→death→discharge→END	51	7%
	3	START→admission→CCU→discharge→END	49	7%
	4	START→admission→CCU→non-ICU ward→non-ICU ward→discharge→END	28	4%
	5	START→admission→non-ICU ward→CCU→non-ICU ward→discharge→END	24	3%

Table 4.1: The top five pathways for patients travelling through the coronary care whose stay was under a week long (shown in the top half of the table) and over a week long (shown in the bottom half of the table). The right two columns show the number of patients who followed a specific path and what percentage of event log for the total patient group the pathway represents.

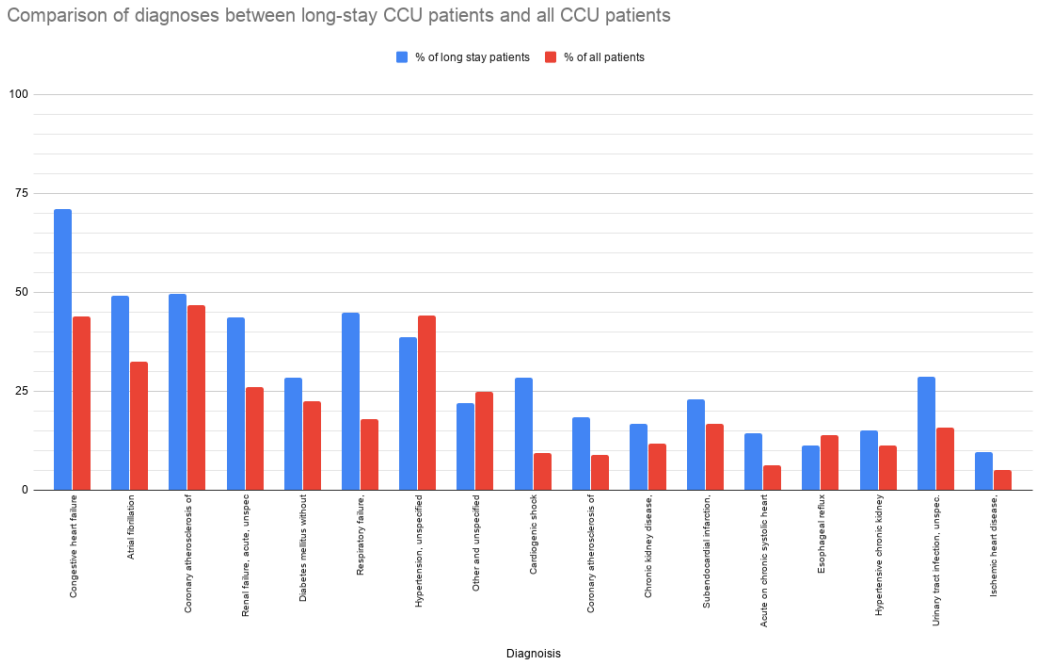


Figure 4.7: Comparison of diagnoses between long-stay CCU patients and all CCU patients

From the pathways in Table 4.1, it is interesting to note that in the top four variants for patients staying over a week in CCU, the patients are admitted directly to that care unit. However, the fifth most common path is the same as the first most common path except patients spend some time in a non-ICU ward before being admitted to CCU.

Days in CCU	Pathway	% M, % F	Ischaemic Heart Disease	Age (mean)	Age (SD)
	1	59%, 41%	7%	65	14.59
	2	53%, 47%	9%	66	14.11
<7	3	63%, 37%	7%	62	16.82
	4	58%, 42%	8%	67	13.78
	5	55%, 45%	10%	67	15.10
	1	62%, 38%	14%	69	12.45
	2	67%, 33%	4%	71	11.36
≥ 7	3	48%, 52%	12%	66	14.25
	4	61%, 39%	11%	66	14.87
	5	79%, 21%	7%	68	12.77

Table 4.2: Age, sex and ischaemic heart disease statistics for the top patient paths defined in Table 4.1.

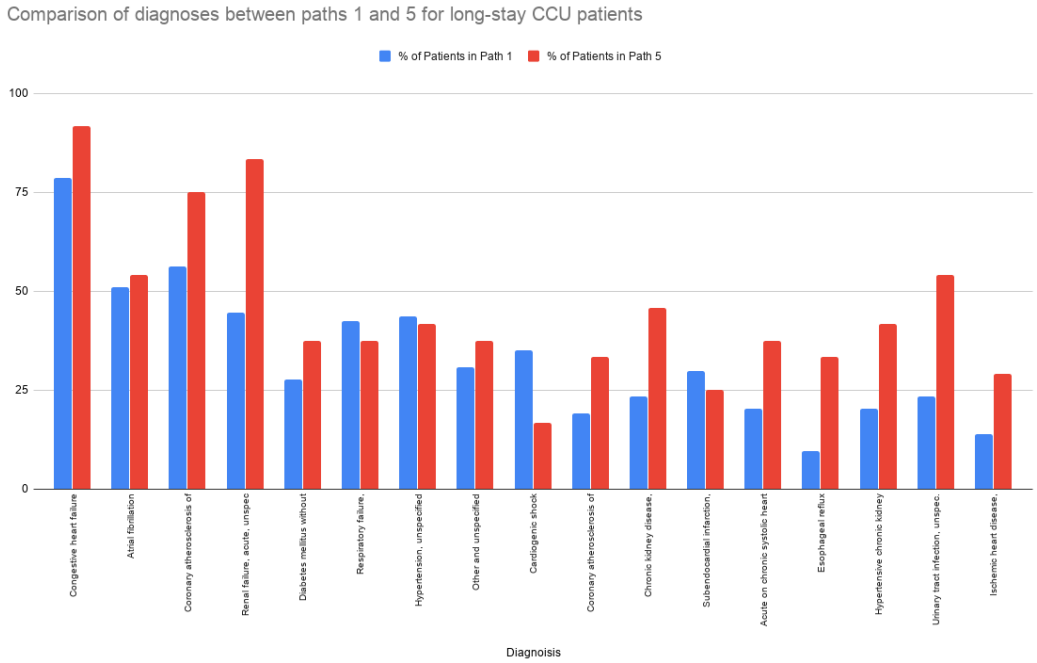


Figure 4.8: Comparison of diagnoses between paths 1 and 5 for long-stay CCU patients

This might indicate that the patient's condition got worse during their hospital stay. To further examine this, we chart the most common diagnoses compared between path 1 and path 5 patients in Figure 4.8. It can be noted that cardiogenic shock appears twice as often in Path 1 than in Path 5, suggesting it occurred before admission, rather than during the patient's hospital stay. On the other hand, Path 5 patients are more likely to have chronic conditions such as kidney disease, esophageal reflux and ischaemic heart

disease. I presented these results to a clinician, who noted that Path 1patients were more likely to have acute conditions such as cardiogenic shock makes sense, since this is a clear indication that CCU is needed.

4.6 Case Study: Patients with Myocardial Infarction

Table 4.3 shows the top five most common paths for patients who were diagnosed with a myocardial infarction (MI) and those who were not. It can be seen that the top four paths for both groups are the same, indicating that in these cases, there are no special underlying processes for MI patients. However, when shown to the clinical team for evaluation, Path 5 for MI patients (START→non-ICU ward→non-ICU ward→CCU→non-ICU ward→discharge→END) was identified as a problematic path. This is because the patients following this path visited two non-ICU wards before being admitted to the coronary care unit for their myocardial infarction. This indicates that it took a longer time than necessary to identify the MI, and the patient was needlessly admitted to other wards when it would have been best for them to be in CCU. This is a pathway which the clinical team identified could be further investigated, to identify common features of patients going down this path and identify them before they spend too much time in wards before eventually being admitted to CCU.

MI diagnosis	Rank	Pathway	Cases	Percentage
	1	START→admission→CCU→non-ICU ward→discharge→END	1033	21%
	2	START→admission→non-ICU ward→CCU→non-ICU ward→discharge→END	447	9%
no MI	3	START→admission→CCU→discharge→END	259	5%
	4	START→admission→CCU→non-ICU ward→non-ICU ward→discharge→END	222	5%
	5	START→admission→CCU→death→discharge→END	191	4%
	1	START→admission→CCU→non-ICU ward→discharge→END	861	25%
	2	START→admission→non-ICU ward→CCU→non-ICU ward→discharge→END	319	9%
MI	3	START→admission→CCU→discharge→END	245	7%
	4	START→admission→CCU→non-ICU ward→non-ICU ward→discharge→END	180	5%
	5	START→admission→non-ICU ward→non-ICU ward→CCU→non-ICU ward→discharge→END	125	4%

Table 4.3: The top five pathways for patients travelling through the coronary care who were not diagnosed with an MI (shown in the top half of the table) and those who were diagnosed with an MI (shown in the bottom half of the table). The right two columns show the number of patients who followed a specific path and what percentage of event log for the total patient group the pathway represents.

In Table 4.4, statistics for sex, age and ischaemic heart disease history are listed. It can be noted that the only path where there are more female patients than male is Path

MI Diagnosis	Pathway	%M, %F	Ischaemic Heart Disease	Age (Mean)	Age (SD)
no MI	1	61%, 39%	7%	64	15.68
	2	57%, 43%	12%	65	14.39
	3	63%, 37%	8%	58	18.71
	4	60%, 40%	11%	66	14.59
	5	55%, 45%	5%	70	13.27
MI	1	57%, 43%	5%	67	12.73
	2	51%, 49%	5%	68	13.49
	3	49%, 51%	6%	67	12.69
	4	55%, 45%	3%	69	12.69
	5	51%, 49%	6%	68	14.73

Table 4.4: Pathway statistics for patients who had an MI and patients who did not. The table shows distribution of sex, percent of patients following that pathway which had a history of chronic ischaemic heart disease, and mean and standard deviation of age.

3 for MI patients. In this path, the patient is admitted directly to CCU, meaning the MI was quickly identified. However, for the same path way for the non-MI patient group (the third most common pathway followed by this group of patients), the sex balance was 63% male and 37% female. This could indicate that patients were admitted directly to CCU when a severe condition such as an MI was suspected, but perhaps this initial evaluation was incorrect and the patient was discharged directly from CCU, and that most of such patients are male. This fits with published literature showing that there are differences in how men and women are treated for myocardial infarction [3].

4.7 Comparison of Models

4.7.1 Performance

As demonstrated above, the models created using PM4Py with InductiveMiner and HeuristicMiner had high fitness measures meaning they could accurately replay much of the behaviour of the event log. HeuristcMiner is cited most commonly in literature applying process mining in healthcare contexts on account of its ability identify common paths by taking transition frequencies into about. In healthcare, it is important to build a model that generalises well as it need to be able to model the noisy and highly variable pathways which occur in the data. By analysing singular paths one by one as explored in section 4.5 and section 4.6, the data which the user is interpreting only represented at maximum 43.6% of patients.

4.7.2 Usability

When these models were presented to the DataLoch clinical team, the models representing entire patient groups such as the full models created with InductiveMiner and HeuristicMiner in section 4.4, were found to be difficult to read. Although hidden transitions greatly improve the models' fitness replay scores, they clutter the Petri net, making transitions and paths much harder to clearly identify.

User-friendly tools such as Disco were also shown to the clinical team. These tools are meant facilitate visualisation of processes and help with statistical analysis of path variants. However, the clinical team identified several functions which they would like a software like Disco to have which it was lacking. These include:

- The ability to examine patient statistics and trends for individual pathway groups as was done for the analysis in section 4.5 and section 4.6.

- The ability to filter for time spent in a singular event, for example filtering for patients who spent an unusual amount of time in CCU, e.g. less than a day or over a week. Disco allows for filtering by whole case length, but can not do more fine tuned duration filtering.

- The ability to compare statistics for patients on the same path who spent different lengths of time in a certain ward. For example, for instances where patients first went to a non-ICU ward and then to coronary care (START→non-ICU ward→CCU→discharge→END), we can compare patients who were in a non-ICU ward for a short time before going to CCU to patients who were in a non-ICU ward for a long time before going to CCU. Through this analysis we can gain insights about what characteristics are prevalent in patients who needed to go straight to CCU, but went for a short time to a different ward first.

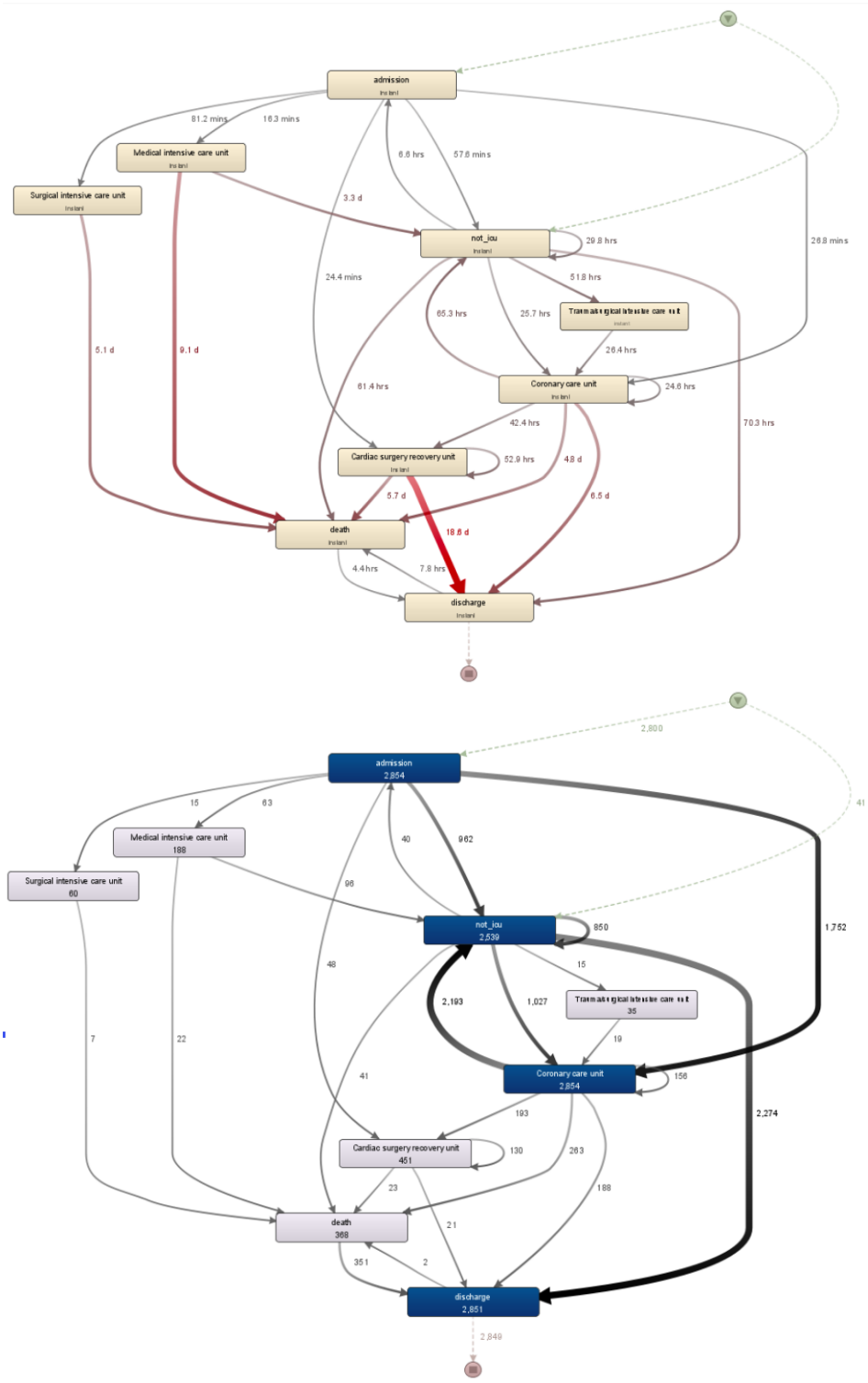


Figure 4.9: Models created by the software Disco for patients who stayed in the coronary care unit during their hospital stay. The left model represents average length of stay in each ward, while the right shows frequency of transitions between wards.

Chapter 5

Conclusions

5.1 Discussion of Implementations and Results

To conclude, in this project I have shown that there are a number of different ways to implement process mining techniques with the MIMIC III dataset. The successful application of these is proven through the implementations of process mining to the specific use-cases of examining pathways of patients who spent over a week in CCU and patients who were diagnosed with an MI.

In this study, I analysed the fitness, precision, geneararalisation and simplicity of these model. I found that the InductiveMiner and HeuristicMiner both perform very well when evaluated on these metrics. However, upon evaluation with input from the DataLoch clinical team, it was found that not many insights could be gained from the data-encompassing models generated by these algorithms, and that actually more exploitative clinical questions could be answered by analysing patients travelling down individual path variants. This suggests that there is more to process mining than just looking at a process model. If knowledge is to be gained about how to address problematic patient paths, clinical and administrative staff need to be able to clearly see the flow of patients and what attributes differentiate patients who take different pathways.

5.2 Future Work

Based on these conclusions, the main way to improve the insights we can gain from process mining with healthcare data systems is to create tools which present process models in an easy-to-read way. Although the main goal of process mining is to create models which accurately model behaviour in an event log, I have found that the accu-

racy of models does not indicate that clinicans will be able to extract the knowledge that interests them most about a given process. To remedy this, I propose that process mining tools should be created with these additional features:

- The ability to filter patients more finely. Many times, clinicians desire to see models with fairly complex filters on them, consisting of a combination of selecting activities which come before or after others, time spent in a certain activity or between activities and patient attributes such as diagnoses.

- The ability to look at smaller sub-groups once the process model is complete, even if they are just a single pathway variant. This would allow for analysis of not only the most common paths, but also the most common exceptional paths.

- Much more statistical representation of the characteristics of the cases in each path variant. Seeing the path only does not give nearly enough insight as to why different cases follow different paths. I would be interested to see a tool which allows the user to easily plot and compare any statistical data available to them in their database. Although hospital information systems hold a wealth of data about patients, the usefulness of it is lost if it can not be clearly presented to the user.

Process mining is a powerful tool which has the potential to model processes and uncover trends which are invisible to the human eye. However, the main first step in implementing these techniques in hospitals across the world is to create tools which are accessible and intuitive to use for health professionals. The vast amount of logged data which healthcare information systems hold are promising of giving incredible insights into the underlying infrastructure of health and social care paths, which once uncovered can be used to improve these systems, saving lives as well as time and resources.

Bibliography

[1] *International classification of diseases. 9th revision, Clinical modification.* Medicode, Salt Lake City, UT, 5th ed. edition.

[2] S.J. van Zelst A. Berti and W.M.P. van der Aalst. Process mining for python (pm4py): Bridging the gap between process- and data science. *CoRR*, abs/1905.06169, 2019.

[3] O.A. Alabas, C.P. Gale, M. Hall, M.J. Rutherford, K. Szummer, S.S. Lawesson, J. Alfredsson, B. Lindahl, and T. Jernberg. Sex differences in treatments, relative survival, and excess mortality following acute myocardial infarction: National cohort study using the swedeheart registry. *Journal of the American Heart Association*, 6(12):n/a, 2017.

[4] D. Hogg A.P. Kurniati, G. Hall and O. Johnson. Process mining in oncology using the MIMIC-III dataset. *Journal of Physics: Conference Series*, 971:012008, 2018.

[5] F.R. Blum. Metrics in process discovery. *Technical Report TR/DCC-2015-6*, 2015.

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

[7] H. Edmiston. Usher institute project business case. In *Edinburgh and South-East Scotland City Region Deal Joint Committee*, pages 21–26, 3 September 2019.

[8] C.W. Günther and W.M.P. van der Aalst. Fuzzy mining – adaptive process simplification based on multi-perspective metrics. In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *Business Process Management*, pages 328–343, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[9] B. Dongen J. Buijs and van der W.M.P. Aalst. Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. *International Journal of Cooperative Information Systems*, 23:1440001, 03 2014.

[10] A.E.W. Johnson, T.J. Pollard, L. Shen, L.W.H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, and R.G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1), 2016.

[11] S.J.J Leemans and ProM. Inductive visual miner manual. 2017.

[12] R. Mans. *Process Mining in Healthcare Evaluating and Exploiting Operational Healthcare Processes*. SpringerBriefs in Business Process Management. Cham, 1st edition edition, 2015.

[13] J. Muñoz-Gama and J. Carmona. A fresh look at precision in process conformance. In *Business Process Management: 8th International Conference, BPM 2010, Hoboken, NJ, USA, September 13-16, 2010. Proceedings*, volume 6336 of *Lecture Notes in Computer Science*, pages 211–226. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[14] C. Adam Petri and W. Reisig. Petri net. *Scholarpedia*, 3(4):6477, 2008. revision #91647.

[15] L. García-Bañuelos R. Conforti, M. Dumas and M. La Rosa. Bpmn miner: Automated discovery of bpmn process models with hierarchical structure. *Information Systems*, 56:284–303, 2016.

[16] van der-R. Vanwersch R. Mans, W.M.P Aalst and A. Moleman. Process mining in healthcare: Data challenges when answering frequently posed questions. pages 140–153, 01 2013.

[17] A. Rebuge and D.R. Ferreira. Business process analysis in healthcare environments: A methodology based on process mining. *Information systems (Oxford)*, 37(2):99–116, 2012.

[18] W.M.P. Aalst, van der R.S. Mans and R.J.B. Vanwersch. *Process mining in healthcare: evaluating and exploiting operational healthcare processes*. SpringerBriefs in Business Process Management,. Springer, Germany, 2015.

[19] W.M.P. van der Aalst R.S. Mans and R.J.B. Vanwersch. Process mining in healthcare : opportunities beyond the ordinary. *BPM reports*, 1326, 2013.

[20] M. Song, C.W. Günther, and W.M.P. Van Der Aalst. Trace clustering in process mining. volume 17, pages 109–120. Springer Verlag, 2009.

[21] C.W. G ü nther and A. Rozinat. Disco: discover your processes. In Niels Lohmann and Simon Moser, editors, *Proceedings of the Demonstration Track of the 10th International Conference on Business Process Management (BPM 2012)*, CEUR Workshop Proceedings, pages 40–44. CEUR-WS.org, 1 2012.

[22] W.M.P. van der Aalst. *Process Mining Data Science in Action*. Springer Berlin Heidelberg : Imprint: Springer, Berlin, Heidelberg, 2nd edition, 2016.

[23] A.J.M.M. Weijters, W.M.P. Aalst, van der, and A.K. Alves De Medeiros. *Process mining with the HeuristicsMiner algorithm*. BETA publicatie : working papers. Technische Universiteit Eindhoven, 2006.

Appendix A

Petri Nets

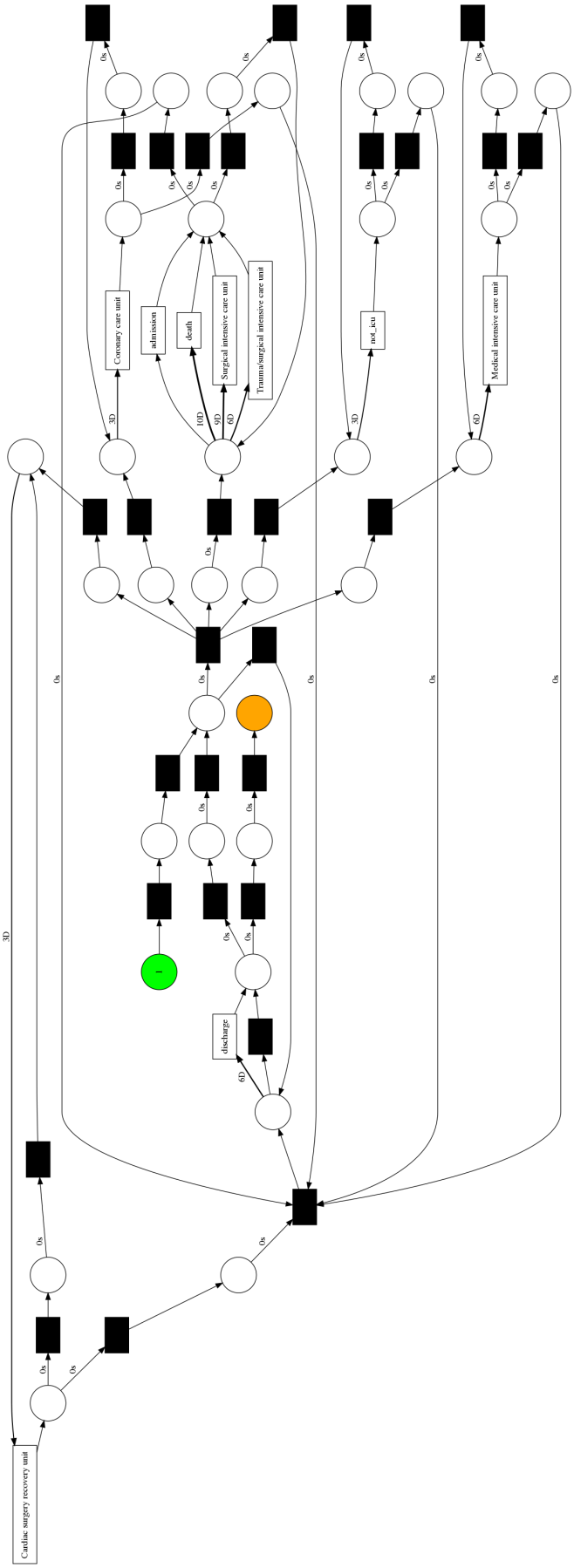


Figure A.1: Model of patients travelling through CCU built using InductiveMiner and PM4Py.

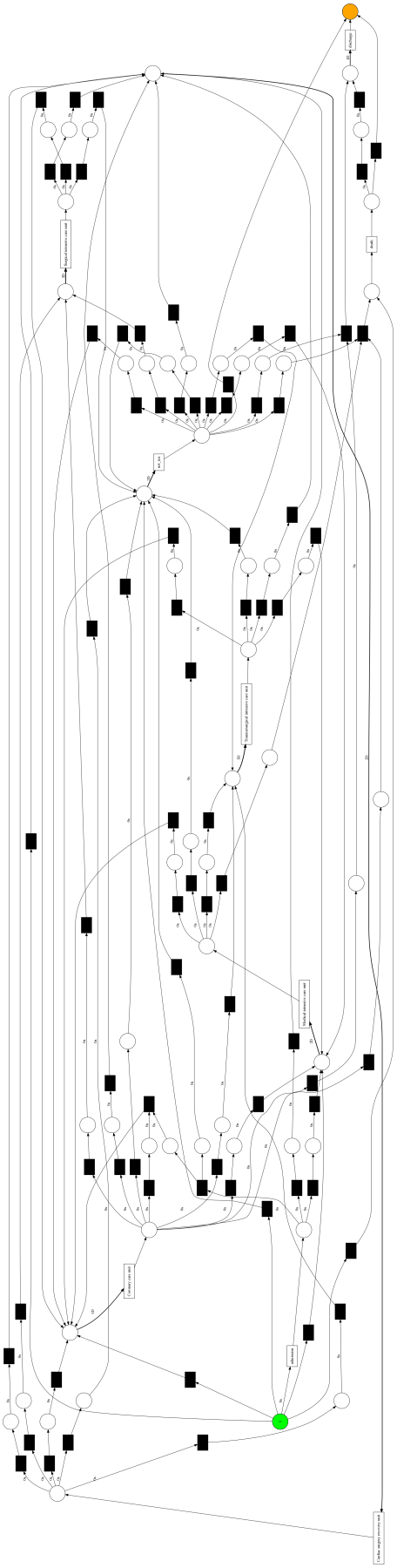


Figure A.2: Model of patients travelling through CCU built using HeuristicMiner and PM4Py.

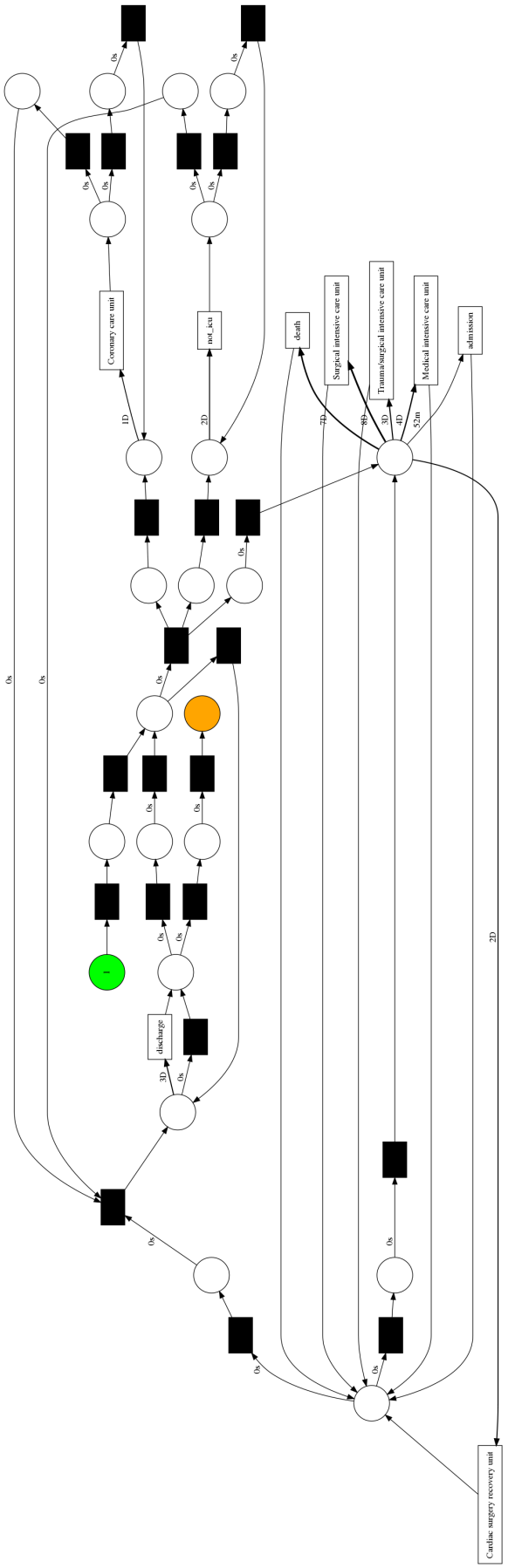


Figure A.3: Model of short-stay patients travelling through CCU built using InductiveMiner and PM4Py.

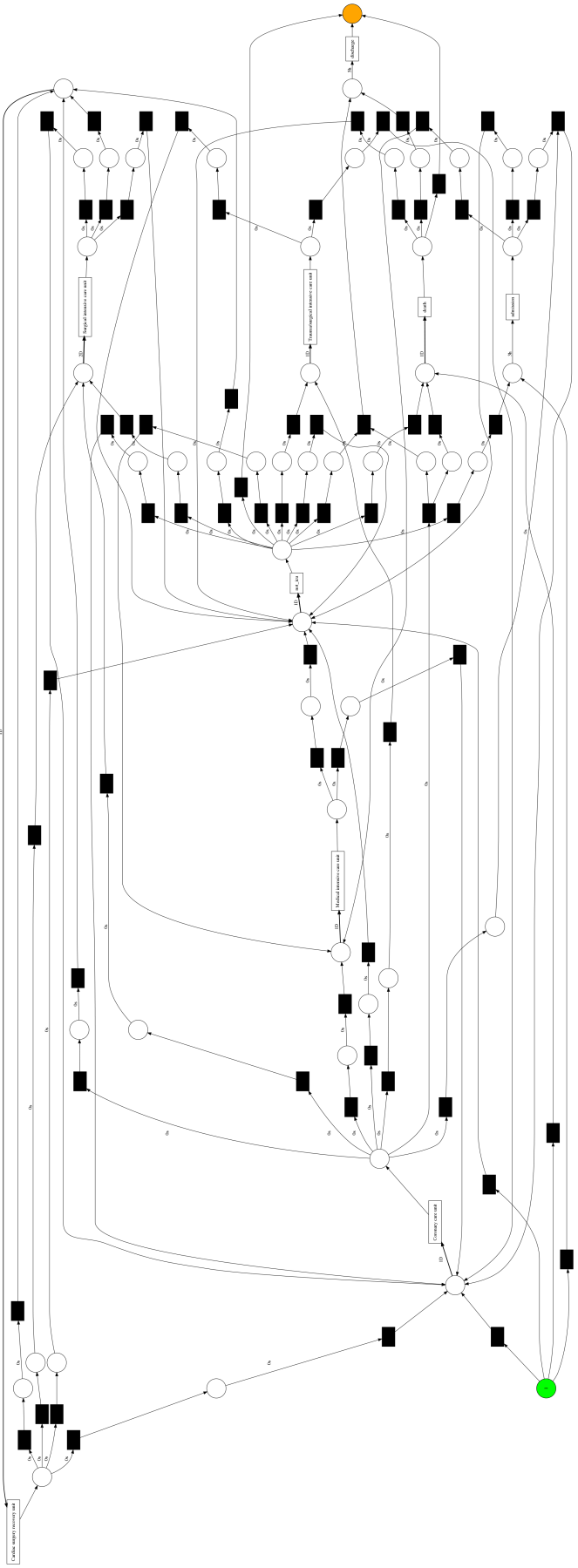


Figure A.4: Model of short-stay patients travelling through CCU built using HeuristicMiner and PM4Py.

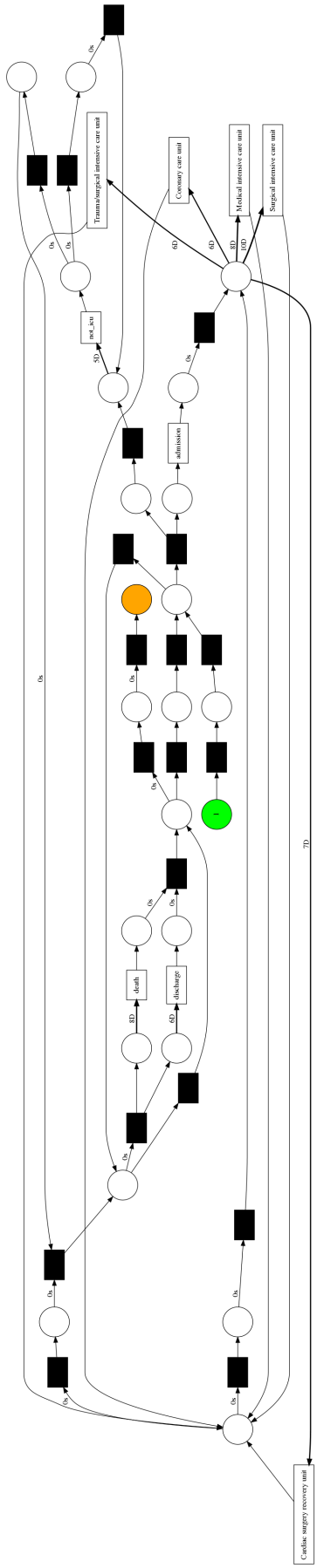


Figure A.5: Model of long-stay patients travelling through CCU built using InductiveMiner and PM4Py.

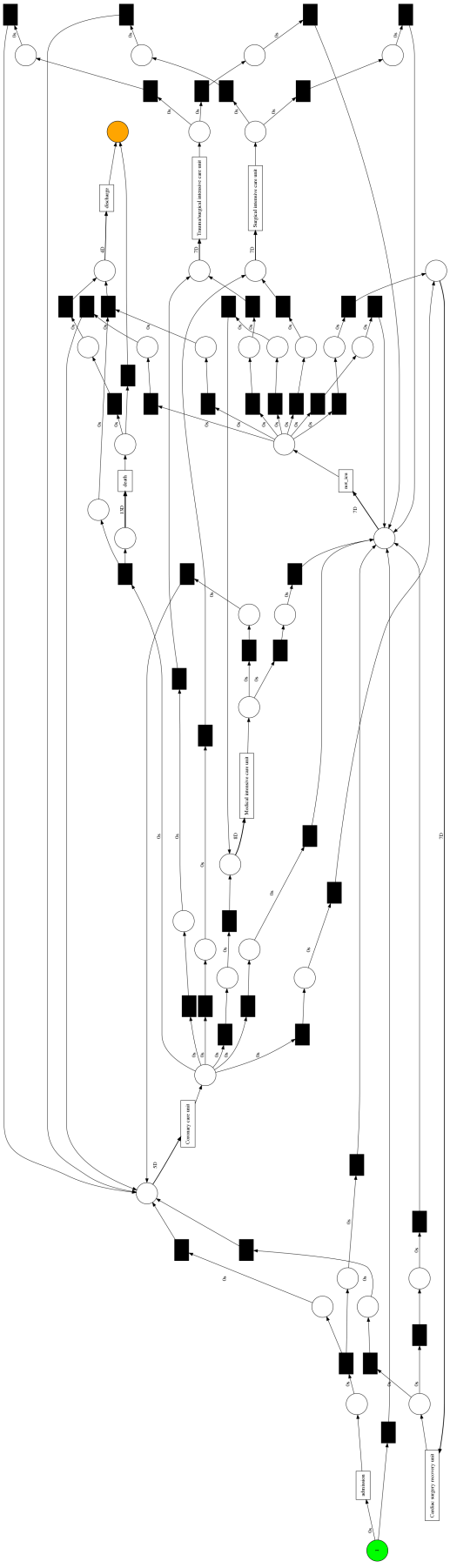


Figure A.6: Model of long-stay patients travelling through CCU built using HeuristicMiner and PM4Py.

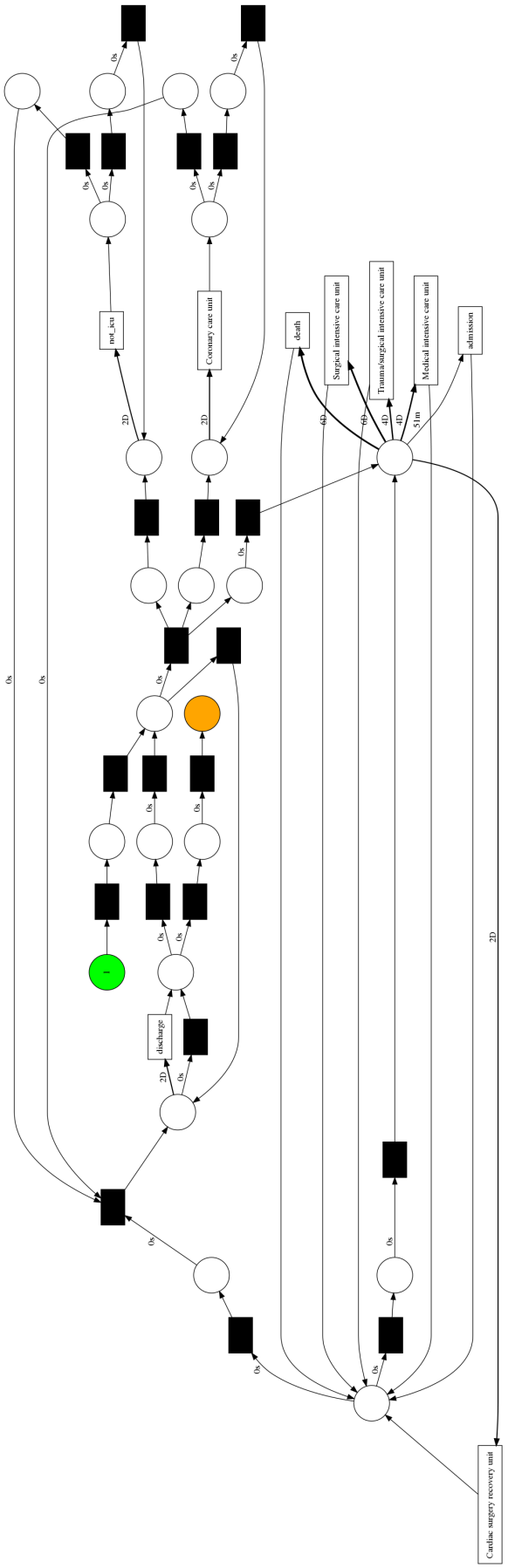


Figure A.7: Model of MI patients travelling through CCU built using InductiveMiner and PM4Py.

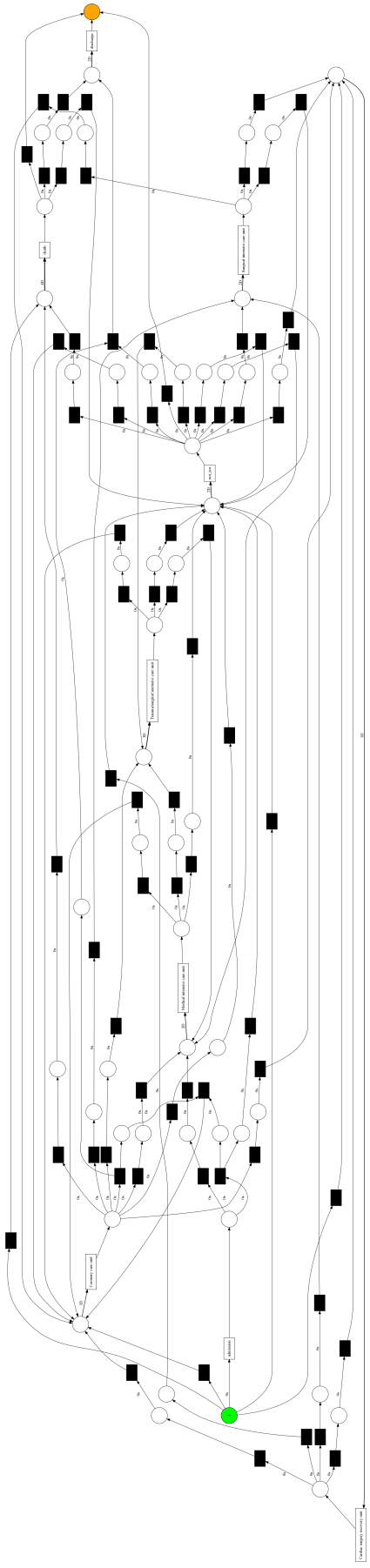


Figure A.8: Model of MI patients travelling through CCU built using HeuristicMiner and PM4Py.