University of Sheffield

Visual Analytics of Temporal Event Sequences

Jessica Gisela Magallanes Castañeda

Supervisor: Dr. Maria-Cruz Villa-Uriol

A thesis submitted for the degree of
Doctor of Philosophy

in the

Department of Computer Science

November 23, 2021

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and are my own work, result of my PhD research. This work has not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

Name: Jessica Gisela Magallanes Castañeda

Date: May 31, 2021

Acknowledgements

First of all I wish to thank my supervisor, Maria-Cruz Villa-Uriol, for her incredible support, guidance, kindness and patience these four years - thank you for encouraging me to continue improving my work and do the best possible. This PhD would have not been possible without my supervisor's support. I would like to express my gratitude to my PhD panel, Dawn C. Walker and Eleni Vasilaki, for their valuable feedback from the start of this PhD. I wish to thank Steven Wood and Lindsey Van Gemeren for providing the initial problem of this PhD and for their valuable collaboration. I would also like to thank Luis Montaña for his incredible help programming some of the features of the system Sequen-C. I am very grateful with CONACYT (National Council of Science and Technology in Mexico) and The Health Foundation for funding my PhD. I would also like to thank the visualization community at the IEEE VIS conferences, it was very inspiring to meet and share ideas with all the amazing people working on visualization.

A big thank you to my friends and all the people I met during this journey. Thank you for making it a very enjoyable and exciting time of my life - thank you for being my family in the UK. I wish to thank James for always being by my side and supporting me in the difficult times (specially during the pandemic). I am eternally grateful for my family, for their love and support these four years: Victor, Mayela, Fernanda, Christopher, Hilda, Tania, Sarah, Virginia, Cristian, Alan, Diego, Angel.

This thesis is dedicated to my grandfather Manuel Castañeda Luevano.

Abstract

Temporal event sequence data (such as event logs) is collected in a wide variety of domains ranging from healthcare to cyber security, vehicle fault diagnosis, population living activities, and web clickstream records. Visual analytics aims to obtain a summary or overview of the data to allow knowledge discovery and support the improvement of the process being studied. Despite the great advances in visual analytics of event data, two main gaps were found in the literature. First, existing visualisations provide an overview of event sequences where its level-of-detail can be transformed by drilling down certain elements, but do not provide dynamic levels of detail simultaneously across sequences and longitudinally. Second, current overviews of event data focus on the visual encoding of sequential patterns but present limitations when representing temporal and multivariate attributes: the attributes are not encoded in the overview or if present, these are oversimplified (e.g. using average values).

This thesis tackles both gaps by proposing a technique to build a *multilevel and multivariate overview* of temporal event sequences. The overview is **multilevel** as its level of granularity can be transformed across sequences *(vertical level-of-detail)* or longitudinally *(horizontal level-of-detail)*, using hierarchical aggregation and a novel cluster data representation *Align-Score-Simplify*. By default, the overview shows an optimal number of sequence clusters obtained through the average silhouette width metric – then users are able to explore alternative optimal sequence clusterings. The vertical level-of-detail of the overview changes along with the number of clusters, whilst the horizontal level-of-detail refers to the level of summarisation applied to each cluster representation. The overview is **multivariate** as it allows to visualise event types in the overview using an *EventBox*, a novel visual encoding that aggregates temporal and multivariate attributes for a set of event occurrences of the same type. The overview allows the identification of trends and outliers involving multivariate attributes within and across clusters.

The proposed technique has been implemented into a visualisation system called *Sequence Cluster Explorer (Sequen-C)* that allows detail-on-demand exploration through three coordinated views, and the inspection of data attributes at cluster, unique sequence, and individual sequence level. The technique is demonstrated through four case studies using three different types of real-world datasets in the healthcare domain: patient flow, hospital admissions and prescription history, and calls made to the emergency services. The case studies show how the technique can aid experts in exploring and defining a set of pathways that best summarise the dataset, while exploring data attributes for selected patterns. Moreover, Sequen-C was evaluated with 13 non-expert users. The results indicate that the system Sequen-C can allow novice users to *quickly* familiarise with the proposed visualisations and successfully obtain insights from the data according to the objective analytic tasks. Furthermore, the results of the System Usability Scale questionnaire indicate that Sequen-C has a *good* usability level.

Contents

List of Figures

List of Tables

Definitions

- Let the event $e = (\tau, t_s, t_e, c)$ be a timestamped occurrence of an action of the type τ (e.g. arrived, closed, logged-in, purchased), where t_s is the start time or ***time of occurrence***, and t_e is the end time. The ***duration*** of an event is computed as $t_e - t_s$, and c is the list of ***event attributes*** which contain information about the occurrence (e.g. room number, item purchased, price).
- A ***point event*** is an event that has a start time but no end time, and therefore has no duration.
- In order to demonstrate the EventBox visualisation (see chapter 5), it is assumed that the ***duration of a point event*** is the same as the time gap between events (i.e. the event continued until the next one in the sequence started). Computing its duration as $t'_s - t_s$, where t'_s is the start time of the next event in the sequence.
- An ***event sequence*** $s = [e_1, e_2, ..., e_n]$, also known as an individual sequence, is an ordered list of events that occur for the same entity (e.g. patient, customer, account, session), where n is the ***length*** of the sequence.
- Let $U = \{s_1, s_2, ..., s_\eta\}$ be the finite set of all the event sequences in a dataset. The set of ***unique sequences*** $S = \{s_1, s_2, ..., s_N\}$ is a subset of U, obtained by removing duplicate event sequences that contain the exact same ordered list of events, where $N \leq \eta$.

Chapter 1

Introduction

1.1 Motivation

Digitalisation is profoundly and rapidly transforming the way everyday activities are carried out [23]. As a result, the world is overflowing with data, about 2.5 quintillion bytes of data are created everyday [64]. Nevertheless, the amount of data collected generally tends to exceed the ability to analyze and interpret such data [82].

Temporal event sequence data, event data for short, is collected and analysed in a variety of domains including healthcare [80], cyber security [18], web clickstreams [96; 58], vehicle fault diagnosis [19], strategy game analysis [65], population daily activities [95], among others. This type of data contains a collection of timestamped event occurrences from which event sequences are obtained. An ***event sequence*** is an ordered list of events that occur for a given entity (e.g. patient, user, client, session). For example in the context of healthcare, an event sequence can indicate the admission, diagnosis, prescriptions, and interventions for a given patient. Alternatively, in the case of web clickstreams, an event sequence can contain the clicks and websites visited by a given user.

The analysis of event data allows to obtain insights from real-world processes, for example: to study the prescriptions that lead to a specific output (e.g. died or lived) or the clickstreams that lead to a customer purchase. The analysis of event data is a complex problem due to its usually high volume and variability [25], caused by large number of individual sequences or event types. Visual analytics offers the opportunity of tackling this complexity by processing and presenting the data in an understandable format to allow knowledge discovery. Numerous visual analytics techniques for event data have been proposed. These techniques usually follow the information-seeking mantra "overview first, zoom and filter, then details on demand" [84]. An ***overview*** of event sequences is a visual summary that usually contains the most common pathways or sequential patterns found in the data. For example, in the context of clinical pathways, a sequential pattern could be in the form:

$$\textit{patient arrived} \rightarrow \textit{waiting consultant} \rightarrow \textit{in consultation} \rightarrow \textit{drug A} \rightarrow \textit{patient left},$$

which indicates that those events, in that order, occurred for a given group of patients.

This thesis presents a visual analytics technique to extract and visualise patterns from event data, where domain experts can explore the data to explore hypotheses and obtain in-

sights. The technique is generalisable to any domain where event data is collected (e.g. in the form of event logs) - however, this work focuses in its application to the healthcare domain. Healthcare information systems such as Electronic Health Records (EHR) have become commonplace around the world [34]. These type of systems generate enormous amounts of data, and it is estimated that around 30% of the world's data is generated in a healthcare setting [43]. Processing and interpreting this wealth of information is an important problem that could enable policymakers to make better decisions towards reducing costs and improving healthcare delivery to patients [85; 8; 7]. In this thesis, the proposed technique is demonstrated through three different types of real-world datasets in the healthcare domain: patient flow, hospital admissions and prescription history, and calls made to the emergency services. The case studies show how visual analytics can enable analysts to explore and understand the data, with the potential of impacting their decision making process.

According to the literature review on visual analytics of event data presented in this thesis, two main gaps were found:

1. Current visualisation systems provide an overview of event sequences where the level-of-detail can be transformed by drilling down certain visualisation elements, but do not provide dynamic levels of detail across both sequences and longitudinally. At the same time, there is no existing technique that allows users to explore different sequence clusterings to obtain a set of distinct pathways that best describe the data.

2. Existing overviews of event data focus on sequential patterns (i.e. patterns with respect to the order of events). However, multivariate event attributes such as duration, time of occurrence, and categorical attributes (e.g. age, gender, medical condition) are usually not included in the overview and can only be accessed through secondary views, or if included in the overview they tend to be oversimplified (e.g. by using average values).

Addressing these gaps solves an important problem that could help obtain more complex patterns involving multivariate data attributes at multiple levels-of-detail.

1.2 Research questions

The main research question addressed in this thesis is:

"How can we use visualisation techniques to detect patterns and anomalies in event data, and explore their causes?"

This question is broken down into the following questions and is expressed in terms of the research gaps in the current literature:

1. How can the analyst control the level of simplification applied to the overview? What visual encoding and interaction controls are necessary to allow analysts to explore different sequence clusterings to define a set of distinct pathways that best summarise the sequences? How to identify deviating pathways? How can long sequences be simplified? How to create a data representation of a sequence cluster that is easy to interpret?

2. How can multivariate attributes be aggregated and visually encoded within the context of a visualization of sequential patterns? How does the addition of this type of visualization element ease the interpretation of the sequential pattern visualization? What interaction controls should be allowed to avoid visual clutter?

3. How to visually encode outliers with respect to multivariate attributes in the overview?

4. How can we use interactive visualisation techniques to concisely provide summaries of key performance indicators (e.g. waiting time and length of stay), to facilitate the identification of operational anomalies, trends and their causes; and to use these to support the optimisation of patient flow along care pathways using visual analytics of event data?

1.3 Contributions

To answer the previous research questions, this thesis has made the following contributions:

1. *Multilevel overview:* A technique to build and explore a multilevel overview of event sequences, from a coarse to a fine *horizontal or vertical level-of-detail*, using hierarchical aggregation and a novel data cluster representation *Align-Score-Simplify*. This contribution answers research question one.

2. *Multivariate overview:* A methodology for integrating temporal, categorical and sequential patterns into a single overview via EventBox, a novel visual encoding that aggregates multivariate attributes of a set of events of the same type. This contribution answers the second and third research questions.

3. *Sequence Cluster Explorer (Sequence-C)*: A visual analytics system that implements the proposed (multilevel and multivariate) overview, and allows detail-on-demand exploration and the analysis of data attributes at cluster, sequence, or individual record level. This contribution answers all the research questions.

4. *Real-world case studies in healthcare*: These explore the application of Sequence-C to four real-world datasets obtained from different clinical environments, demonstrating the impact and generalisability of the proposed methodologies. This contribution answers research question four.

5. *Evaluation of Sequen-C*: the system was evaluated with 13 non-expert users. The results indicate that Sequen-C can allow novice users to *quickly* familiarise with the proposed visualisations and successfully obtain insights from the data according to the objective analytic tasks. Furthermore, the results of the System Usability Scale questionnaire indicate that Sequen-C has a *good* usability level

1.4 Overview of the report

Chapter 2 reviews the topics most related to this thesis, including: Visual analytics in healthcare, Visual analytics of event data, Sequence Clustering, and Multiple Sequence Align-

ment. Firstly, the relevance of visual analytics is explained in the context of healthcare. Secondly, a classification of visualisation techniques is presented and these are compared with the present work. Then, clustering methods and similarity measures are reviewed, highlighting the methods adopted in this work. Lastly, the algorithm for multiple sequence alignment is explained step by step, which is relevant to the cluster representation proposed in Chapter 4.

Chapter 3 outlines the analytic tasks that guided the development of the methodology and system proposed in this thesis.

Chapter 4 presents a novel technique to build and navigate a multilevel overview of temporal event sequences (first contribution). This technique is implemented into a first version of the system Sequen-C (third contribution), which is then demonstrated using two real-world datasets: MIMIC-III and Antenatal Care Unit (fourth contribution). This chapter is based on a paper submitted to the IEEE VIS 2021. Notification of first review cycle will be given on 13 June 2021.

Chapter 5 builds on the technique presented in Chapter 4 by integrating temporal and multivariate attributes in the multilevel overview via EventBox, resulting in a multilevel and multivariate overview of event sequences (second contribution). Sequen-C is extended to implement the analytic tasks of EventBox and it is demonstrated using a real-world dataset from a Rheumatology outpatient clinic (fourth contribution). This chapter is based on [61], a paper published at the IEEE VIS 2018 conference.

Chapter 6 presents a case study where the full technique is evaluated on a real-world dataset that contains calls to the emergency department (fourth contribution). A set of findings is discussed, which show how Sequen-C could have a real impact in supporting the decision making process of policymakers and call handlers.

Chapter 7 evaluates the technique with fifteen participants, thirteen novice and two expert users, in terms of user performance and usability. The results indicate that the system Sequen-C can allow novice users to quickly familiarise with the proposed visualisations and successfully obtain insights from the data according to the analytic tasks. The results of the System Usability Scale questionnaire indicate that Sequen-C has a *good* usability level.

Chapter 8 outlines the main conclusions of the thesis and discusses future work.

Chapter 2

Literature review

This chapter reviews the related work on the topics: visual analytics in healthcare, visual analytics of event data, sequence clustering, and Multiple Sequence Alignment. Section 2.1 reviews the relevance of visual analytics in the context of healthcare. The section contrasts the huge amount of data being generated everyday in healthcare settings versus a shortage in analytical capability to interpret such data. Visual analytics has the potential of tackling this shortfall and support quality improvement in healthcare. The section also outlines the current challenges of visual analytics in healthcare. Section 2.2 presents a classification of visualisation techniques for event data, focusing on the methods for aggregating and summarising event sequences. These methods are then compared with the technique presented in this thesis. Section 2.3 reviews clustering methods and similarity measures for event sequences, highlighting the methods adopted in this work. Section 2.4 explains in detail the algorithm for multiple sequence alignment, which is adopted in the cluster representation proposed in Chapter 4.

2.1 Visual analytics in healthcare

As indicated by Bardsley *et al.* [8] in the report "Untapped potential: Investing in health and care data analytics":

> "An organisation's analytical capability is their ability to analyse information and use it to make decisions"

Current healthcare systems face the challenge of improving the outcomes of care delivery while also reducing costs [42]. This big challenge is usually approached by healthcare organisations through a set of systematic quality improvement methods [46; 5]. These methods include [74]: Lean [2], model for improvement [20], Six Sigma [54], Plan-Do-Study-Act [88], among others. Such improvement methods are traditionally implemented through lengthy projects that follow a life cycle consisting of the following steps [55]: 1) identify a problem, 2) define standards and criteria, 3) collect data, 4) analysis, 5) implement changes, and 6) re-audit. Unfortunately, most of these steps tend to be too time consuming and could greatly benefit from current technology advances.

On the other hand, there are enormous mountains of data being generated every day in healthcare systems - with the increasing adoption of information systems such as Electronic Health Records (EHR) [34] and the increasing popularity of apps and websites to record personal health data [8]. It is estimated that around 30% of the world's data is generated in a healthcare setting [43]. Understanding this data could complement current quality improvement methods, and enable stakeholders to identify problems and evaluate changes towards improving the quality of care. Making better use of the available data and digital technologies is one of the five points included in the long term plan of the NHS [28] to overcome their current challenges, where service improvement is meant to be driven by the analysis of patient and population data. Nevertheless, a recent report by The Health Foundation [8] has identified a shortage of analytical capability in the healthcare system - there are not enough people with the required skills or advanced tools to support good data analysis. Visual analytics techniques and tools offer the opportunity of increasing the analytical capability of organisations, by processing the data and presenting it in an interpretable format that allows knowledge discovery.

Visual analytics in healthcare is an important but challenging problem. Previous review papers [3; 34] have identified the main challenges including:

1. **Data scale and complexity**, referring to large number of patients and large number of variables, including the challenge of linking data across multiple data providers.

2. **Data quality and uncertainty**, missing or incomplete data can lead to misinterpretation.

3. **Scalable analysis**, to allow one to discover insights from single patients or large cohorts.

4. **User interaction and interfaces**, the design should account for different backgrounds and analytic tasks of each type of user (e.g. physician, researcher, patient).

The present work focuses on the first, third, and fourth challenges. Data scale and complexity, specifically in event data, is approached by proposing a Multilevel and Multivariate

Overview of Temporal Event Sequences (see chapters 4 and 5). The proposed technique allows a scalable analysis as the data can be analysed at cluster or individual patient level. Moreover, the system that implements the proposed technique was designed according to the required analytic tasks of the domain experts.

2.2 Visual analytics of event data

A variety of visualisation techniques for event data have been proposed - they can vary according to the analytic task at hand (e.g. cohort comparison, outcome analysis, multivariate attribute analysis), but they all share the goal of creating a visual summary or *overview* of event sequences able to represent common or anomalous sequence patterns. Section 2.2.1 describes current techniques to create an overview of event sequences, and section 2.2.2 presents the type of user interaction options to explore such overview.

2.2.1 Event data summarisation

Creating an overview of event sequences is a challenging task given the complexity of this type of data (e.g. long sequences, large number of individuals, large number of event types). An overview of event sequences can be created through explicit summarisation, inexplicit summarisation, progression analysis, or clustering [41]. Explicit summarisation arranges the complete event sequences into the overview, whereas inexplicit summarisation does not take the complete sequence and extracts only relevant sub-sequences from the dataset. Explicit and inexplicit summarisation produce overviews where the order of events is visually represented, whereas, progression analysis divides events into stages where the order of events is not important. For example, the order of the events "Blood Test - Waiting Consultation - In Consultation" is irrelevant in progression analysis and they are instead part of the stage "monitoring patient symptoms". On the other hand, clustering techniques divide sequences into groups to obtain a simplified set of pathways that summarise the sequences.

2.2.1.1 Explicit summarisation

Techniques based on **explicit summarisation** aggregate the complete event sequences and arrange them using Sankey-like diagrams [101; 24] or icicle plots [102; 90; 65]. A Sankey diagram represents the flow between two or more items using a directed weighted graph. Items are visually represented using *nodes* and are connected by links called *edges*, where the sum of the incoming weights of a node should equal its outgoing weights [79] (e.g. Fig. 2.1). An icicle plot is used to represent a hierarchical structure, where the nodes of a tree are visually represented as adjacent rectangles. When using a horizontal layout, the root node is represented by a square at the top and children nodes are placed under parent nodes [102] (e.g. Fig. 2.3-1).

Visualisations based on Sankey diagrams [83] use nodes to represent the event types and edges to represent the transition between event types. The width, height, and colour of the edges and nodes can be visually encoded to aggregate event attributes. For instance, as observed in Fig. 2.2-(3), Outflow [101] uses the height of the node to represent the number of records entering that node, the colour of the edge is used to represent average outcome of the sequence (e.g. winning or losing a soccer match), and width of the edge to show average

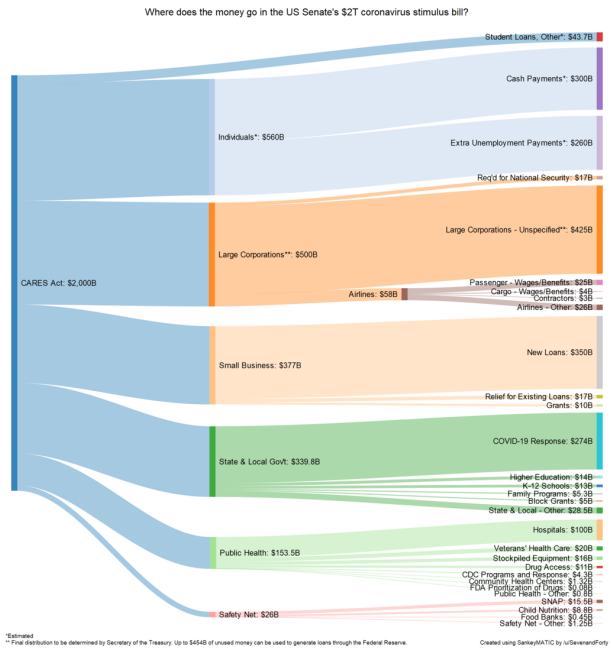


Figure 2.1: *Example of a Sankey diagram representing the allocation of a $2 trillion fund (i.e. CARES Act) to relief the economic impact of the COVID-19 pandemic in the United States. Image by SevenandForty via Wikimedia Commons.*

duration. A disadvantage of Sankey-like visualisations is the overlapping of edges as the number of event types increase. To tackle this issue, Zhao et al. [106] propose MatrixWave as an alternative to Sankey-like diagrams (see Fig. 2.2-(4)), where the transitions are shown as zig-zag patterns going through a series of connected matrices.

Icicle plot visualisations order sequences in alphabetical order forming a tree-like structure that branches as sequences diverge from each other. For instance, as shown in Fig. 2.2-(1), EventFlow [65] represents events using vertical bars, the height of which is encoded according to the number of records and width according to the average duration. To cope with visual clutter, users can simplify the overview by removing or grouping similar event types. Despite the available user interaction, icicle plot visualisations become difficult to interpret as the number of event types increase. Overall, methods that use an explicit summarisation approach do not handle well large numbers of sequences or event types, as this generates more variability or very large sequences (i.e. sequences containing dozens or hundreds of events).

2.2.1.2 Inexplicit summarisation

In order to address volume and variability issues, **inexplicit summarisation** approaches aggregate sequences by extracting a set of relevant (e.g. frequent) events or sub-sequences [95; 77; 58; 49; 76], as opposed to explicit summarisation where the complete sequences are visualised. This approach uses techniques such as sequential pattern mining [4]. For example, Frequence [76] builds an overview of frequent sequential patterns using a Sankey-like diagram (see Fig. 2.2-(2)), where users can select a coarse pattern from which sub-patterns with finer level-of-detail are mined. Given the mentioned scalability limitations of Sankey-based

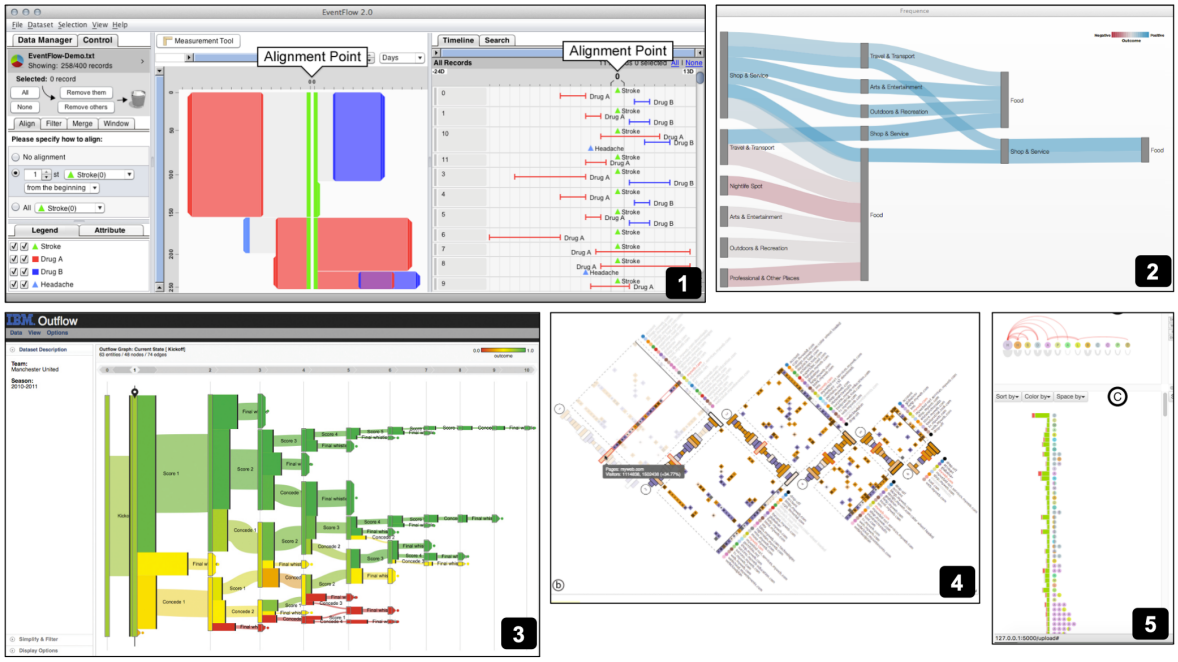


Figure 2.2: *(1) Eventflow [65] visualisation, an example of explicit summarisation using an icicle plot, the visualisation shows 4 event types. (2) The Frequence [76] visualisation uses a Sankey-like diagram to show frequent sequential patterns (inexplicit summarisation). (3) Outflow [101] visualisation using a Sankey-like diagram (inexplicit summarisation), where events in the same layer with similar outcome are merged. (4) MatrixWave [106] presents an alternative to Sankey visualisations to avoid overlap of edges. (5) Peekquence [49] shows sequential patterns as a list of event glyphs, as an alternative to Sankey representations.*

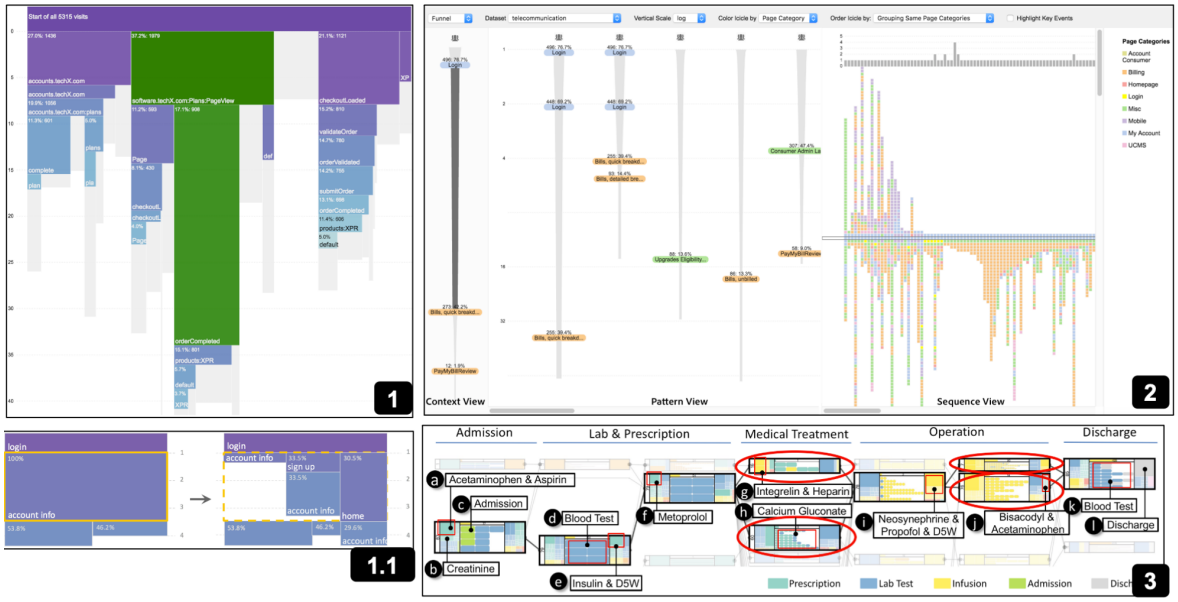


Figure 2.3: *(1) Coreflow [56] visualisation, showing branching patterns in an icicle plot. (1.1) A selected branch (e.g. login - account info) is drilled down to see sub patterns. (2) CoreflowVis [58] shows sequential patterns using a funnel visualisation. (3) Progression analysis in EvenThread2 [39], the figure shows 5 phases of clinical progression.*

diagrams, Peekquence [49] presents such sequential patterns as a list of event glyphs next to a bar representing their frequency (see Fig. 2.2-(5)). However, sequential pattern mining can generate large number of patterns. Liu et al. [58] propose a method to reduce this number by pruning those with a certain level of overlap in their support set - where sequential patterns are presented using a funnel visualisation (Fig. 2.3-(2)). Alternatively to the sequential pattern mining algorithm, CoreFlow [56], obtains a tree-like overview of branching patterns using the algorithm Rank-Divide-Trim (Fig. 2.3-(1)), in which users can click on a branch to compute more detailed sequential patterns in that sub-sequence (Fig. 2.3-(1.1)).

2.2.1.3 Progression analysis

Progression analysis consists of obtaining a set of high-level stages across event sequences. A stage can be seen as a *bag* of events, where the ordering of the events does not matter. Taking the example from Guo *et al.* [39], in the context of healthcare, a patient may first get a blood test and see a consultant afterwards; in some cases, the order of these events might not be relevant as they are simply part of the "monitoring symptoms" stage which can then lead to other stages (such as getting a specific medical intervention). EventThread [40] obtains progression patterns, called threads, through tensor analysis. The threads are then segmented into stages using fixed time windows. Building on this work, EventThread2 [39] removes the limitation of defining fixed time windows by proposing an unsupervised algorithm for obtaining stages, and presents stages in a node-link visualisation (Fig. 2.3-(3)).

2.2.1.4 Clustering

Visualisation techniques based on clustering aim to divide data into groups of similar items using features such as event types or data attributes, and to provide an interpretable visualisation of the composition of each cluster. These techniques are generally applied either to the clustering of events or the clustering of sequences.

Event clustering: Outflow [101] visualises common pathways using a Sankey-like diagram, and to reduce visual clutter, events are clustered according to their outcome. However, the clustering is limited to events within the same layer. Gotz *et al.* [37] propose a technique for dynamic hierarchical aggregation of events, where users can choose alternative groupings within a hierarchy of events, according to their correlation with outcome. Scribe Radar [103] builds a hierarchy of event types based on event occurrence frequency and a previously defined six-level naming hierarchy. These two last approaches rely on an existing hierarchy of event types (e.g. hierarchical classification of clinical codes), which is a limitation in application domains that do not have such hierarchies.

Sequence clustering: Wang *et al.* [96] build an overview of user behavioral patterns by clustering users with a similar clickstream history, visualised using a Packed Circle view. This visualisation successfully captures the hierarchy of nested clusters, however, the event sequences are not included in the overview (Fig. 2.4-(1)). Wei *et al.* [100] build an overview of clickstream clusters mapped on a 2D plane. The sequential information is visually encoded but the separation of different clusters is difficult to interpret (Fig. 2.4-(2)). Gay *et al.* [32] visualise clusters of event sequences in a 2D heatmap grid including event type and time interval information. However, it might be laborious to derive the original event sequences from the cluster representation. Other strategies explicitly encode the event sequences stacked and grouped by cluster, comparing them side by side [17; 87] (see Fig. 2.4-(3)); however, this approach complicates the comparison of sequences within or amongst clusters. Treemaps are commonly used to visualise hierarchical clusters [62; 36]. However, this type of view is not suitable for clusters of sequences as it does not encode sequential information (see Fig. 2.4-(4)).

Vasabi and Sequence Synopsis are the techniques most similar to this work. Vasabi [72], see Fig. 2.5-(1), builds an overview of sequence clusters by first extracting the most common events (e.g. tasks) in the dataset and then clusters sequences using those events as features of the clustering. This technique successfully extracts and represents sequence clusters, but it presents the following limitations: 1) the cluster representation does not allow one to see the events that were omitted in the task extraction step, which could mean missing interesting anomalous event occurrences. 2) it allows one to see the probability of an event within a cluster but it does not allow one to see the order in which events occur to identify event permutations. 3) the number of clusters cannot be changed, so users cannot explore alternative clusterings. Sequence Synopsis [19], see Fig. 2.5-(2), proposes a method to cluster event sequences, based on the minimum description length where clusters are represented by a sequential pattern and a set of corrections; as indicated by the authors, a way to improve the scalability of the approach is to support hierarchical visual summary (e.g. explore alternative number of clusters).

Overall, the previous clustering techniques have one or both of the following limitations:

- the cluster representation does not allow one to easily derive the original sequences,

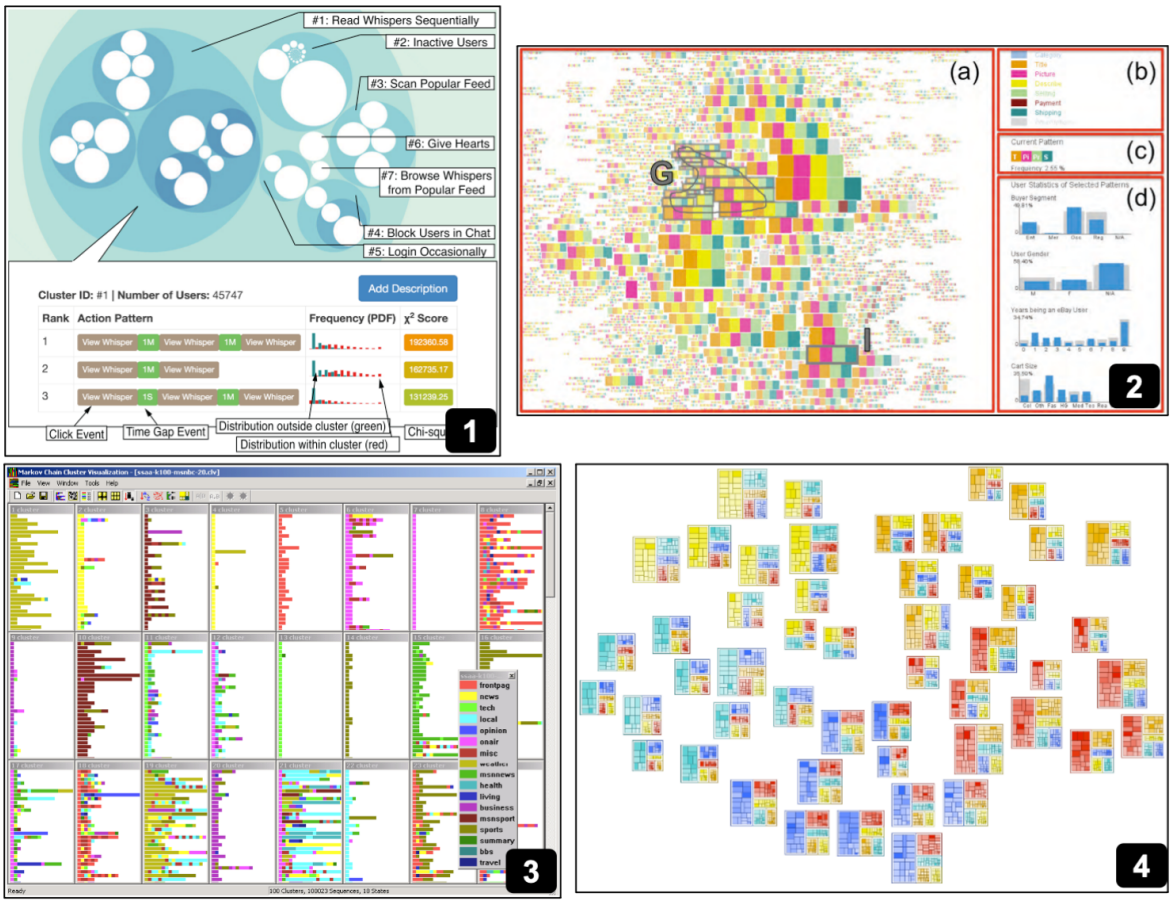


Figure 2.4: *Visualisation designs to represent sequence clusterings. (1) Packed Circle view [96] representing clusters of users clickstream history, each white circle represents a cluster of users which are further clustered using blue circles. (2) 2D plane cluster view [100], each coloured square represents a type of event. (3) Clusters of stacked sequences [17], each sub window represents a cluster of sequences. (4) Cluster visualisation using treemaps [36], one treemap for cluster, each treemap encodes the most prominent events or features of the cluster.*

- the overview of clusters is static as it does not allow one to change the number of clusters or the level of detail shown in all cluster representations.

According to this review, there is no existing technique to explore different sequence clusterings and that, at the same time, provides a clear interpretable representation for each cluster. The present work aims to tackle this problem.

2.2.2 User interaction

Following the information-seeking mantra "overview first, zoom and filter, then details on demand" [84], current visual analytic techniques allow a number of user-driven operations to simplify or rearrange the overview to retrieve further details of interesting components, or to remove non interesting records. In general, these operations can be grouped into three categories: transformations, queries, and alignment. Transformations allow the simplification of the data overview [65], for example either by merging similar event types into a single one or by removing records that are not of interest. Queries allow the filtering of data being visualised according to a set of events or temporal constraints. Existing strategies include: visual queries [66; 29], regular expressions [18; 105] or milestone events [35]. In general, alignment of event sequences by a selected event target the exploration of the subset of events happening right before and after the alignment point [65; 19; 101].

2.3 Sequence clustering

Clustering is a well-known technique that groups objects into clusters so that similar objects are in the same set and dissimilar objects are separated [69]. There is a wide variety of clustering methods. The two most popular approaches are: k-means clustering and hierarchical clustering [44] (see Fig. 2.6).

In k-means the observations are partitioned into the clusters C_1, C_2,..., C_k, where k is the predefined number of clusters. The objective is to minimise the within-cluster variation, which can be computed as the sum of all squared distances between pairs of observations in each cluster [44]. In hierarchical clustering, the result is a nested tree of partitions called dendrogram [44; 69]. A dendrogram depicts an upside-down tree, constructed by combining clusters towards the trunk, where the observations sit on the leaves [44]. k-means is usually faster than hierarchical clustering. However, a disadvantage of k-means is that it requires the pre-specification of k [69]. Given such disadvantage and due to the requirement of Multiple Sequence Alignment to rely on a hierarchical structure (see section 2.4), in this thesis, a hierarchical clustering approach is applied to the clustering of event sequences.

Hierarchical clustering methods can be categorised into: agglomerative and divisive. Agglomerative approaches, also known as bottom up, assign one cluster for each observation, then pairs of similar clusters are iteratively merged until one cluster containing all the observations is obtained. In contrast, divisive methods initialise all observations in the same cluster, then clusters are iteratively divided into the two most different sub-clusters until a desired number of clusters is reached or each observation is alone in a cluster [1]. This thesis focuses on the clustering of temporal event sequences using an agglomerative hierarchical clustering approach.

Figure 2.5: *Most similar techniques to the work presented in this thesis. (1) Vasabi [72] presents an overview of 10 sequence clusters, numbered from G1 to G10 (top left). The most common events are numbered from T0 to T11. The size of the circle represents the probability of event occurrence in that cluster. (2) Sequence Synopsis [19] presents sequence clusters as lists of coloured squares, each representing an event type, triangles represent hidden sub-sequences. A higher level-of-detail can be obtained by clicking the triangles.*

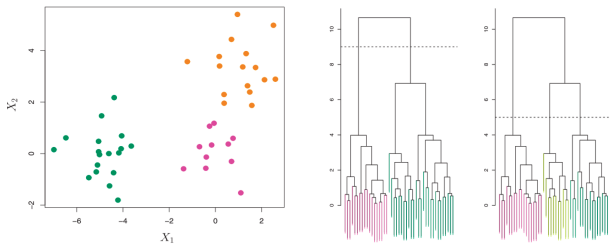


Figure 2.6: *k-means (left) and hierarchical clustering (right). Original images obtained from [44]. Result of the clustering of 45 observations visualised in two dimensions using features X_1 and X_2 (left). In the case of k-means, three clusters are obtained ($k = 3$). The resulting tree by hierarchical clustering shows either two (middle) or three (right) clusters selected by the branches touched by the dashed line. Clusters are colour-coded.*

2.3.1 Similarity measures

The choice of the metric to measure similarity or distance between event sequences is one of the core elements of clustering. Similarity measures for event sequences can be categorised into: vector-based distances and string distances. Clustering methods that use vector based distances (e.g. [38; 86]), turn each event sequence into a vector of n features. For example, n could be the total number of event types in the whole dataset, where each element of the vector indicates the frequency of occurrence of an event within a sequence [14]. Following this example, the sequences *bcbba* and *acbd* are turned into the vectors $[1, 3, 1, 0]$ and $[1, 1, 1, 1]$ respectively - then the distance between vectors can be computed using metrics such as the Euclidean distance, Hamming distance, or Jaccard distance [86]. Vector-based distances can present disadvantages when the interest is to find patterns in the structure of the sequences (i.e. sequential patterns) rather than focusing exclusively on the presence or absence of events in a sequence. On the other hand, string distances allow one to measure permutations in the order of events or quantify frequent subsequences.

String distances treat event sequences as words or strings of characters. This type of distance can be categorised into edit distances and q-gram distances [94].

The edit distance defines the dissimilarity between two sequences as the minimum number of event operations (i.e. insertion, deletion, and substitution) needed to transform one sequence into another, where each operation has a given cost value. Edit distances include: Hamming distance, Longest Common Subsequence, and Levenshtein distance; these distances differ in the types of operations allowed [70].

The **Hamming distance** is the number of event substitutions necessary to transform a sequence s_1 into s_2. For example, the *Hamming* distance between $s_1 = abcde$ and $s_2 = acbdf$ is 3; assuming that the cost for substituting a character is 1, the distance is obtained by replacing b,c,e in s_1 for c,b,f from s_2, which totals 3 substitutions. A disadvantage of Hamming distance is that it can only be applied to sequences of the same number of events, as it only allows substitution operations.

The longest common subsequence (LCS) is the longest set of events shared by both sequences while keeping their order intact. **The longest common subsequence distance** is the number of events not in the LCS [94]. This distance allows insertion and deletion operations. For example, for the sequences $s_1 = abcde$ and $s_1 = acbdf$, the LCS distance is

4; this distance is obtained as the LCS of s_1 and s_2 is *abd* and the events not in the LCS are four: c,e,c,f.

The Levensthein distance is the most common string distance and it is known simply as the edit distance [94]. **The Levenshtein distance** [52] is the minimum number of insertions, deletions, and substitutions to turn one sequence into another. For example, assuming all operations have an equal cost of 1, the Levenshtein distance between $s_1 = abb$ and $s_2 = aa$ is 2; because to turn *abb* into *aa*, it requires at least the substitution of the first b with a and the deletion of the last b [10]. A disadvantage of edit distances is that they can be affected by permutations introduced by unimportant events. Moreover they can be affected by the length of sequences or repetitions of sub-sequences. For example, the sequences $s_1 = abc$ and $s_2 = abcabc$ have a Levenshtein distance of 3, whereas the sequences $s_1 = abc$ and $s_3 = hij$ also have a Levenshtein distance of 3. As observed s_1 is closer related to s_2 than it is to s_3.

The **q-gram distance** compares the occurrences of q consecutive events between sequence s_1 and s_2; for example, the 1-gram distance between $s_1 = abc$ and $s_2 = cba$ is zero, as each of the events a,b,c occur in both sequences [94]. More formally, the q-gram distance of two sequences is defined as the distance of their q-gram profiles, where the q-gram profile of a sequence is the vector of all sub-sequences of q consecutive events [92; 94]. The 2-gram profiles of *abc* are *ab* and *bc*, and for *cba* the 2-gram profiles are *cb* and *ba*; therefore the 2-gram distance between these two sequences is 4, as there are no common 2-gram profiles between the sequences. Different types of q-gram distances can be derived depending on the metric used to measure the distance between q-gram profile vectors (e.g. jaccard, cosine). Q-gram distances have the disadvantage that the value q needs to be specified and the appropriate q value might depend on the specific dataset. However, a constant $q = 1$ was chosen in this thesis, as 1-gram distance returned good clusters for the presented case studies, moreover using $q = 1$ allows for the clustering of sequences according to the count of shared events and is not affected by noise such as repetitions or permutations in the order of events.

2.4 Multiple sequence alignment

Alignment is a common analytic strategy in temporal event data that allows one to explore the events happening right before and after a given event [25]. Existing techniques usually allow alignment by a single event [65; 19; 101] or more recently by multiple events [18].

Multiple Sequence Alignment (MSA) [26] was initially proposed to align biological sequences and understand how they relate to each other. Bose *et al.* [13] applies this algorithm for temporal event sequences to find common behaviour and deviations in a process, with applications in domains such as clinical workflows [107; 15]. MSA has also been used to obtain a consensus sequence (i.e. a set of common events) to represent a set of event sequences [33; 97; 51; 24]. In this thesis, an MSA approach is used to represent the common events in a sequence cluster to allow the comparison of commonalities and deviations within and across clusters.

The objective of MSA is to insert gaps (–) in the input sequences so that the number of equal events column-wise is maximised. For a given set of sequences $S = \{s_1, s_2, ..., s_N\}$, the alignment matrix λ, or ***alignment*** for short, contains all the sequences in S aligned. All elements in an alignment are either single characters or gaps (–). The dimensions of matrix λ are $N \times M$, where N is the number of input sequences and M the length of the final

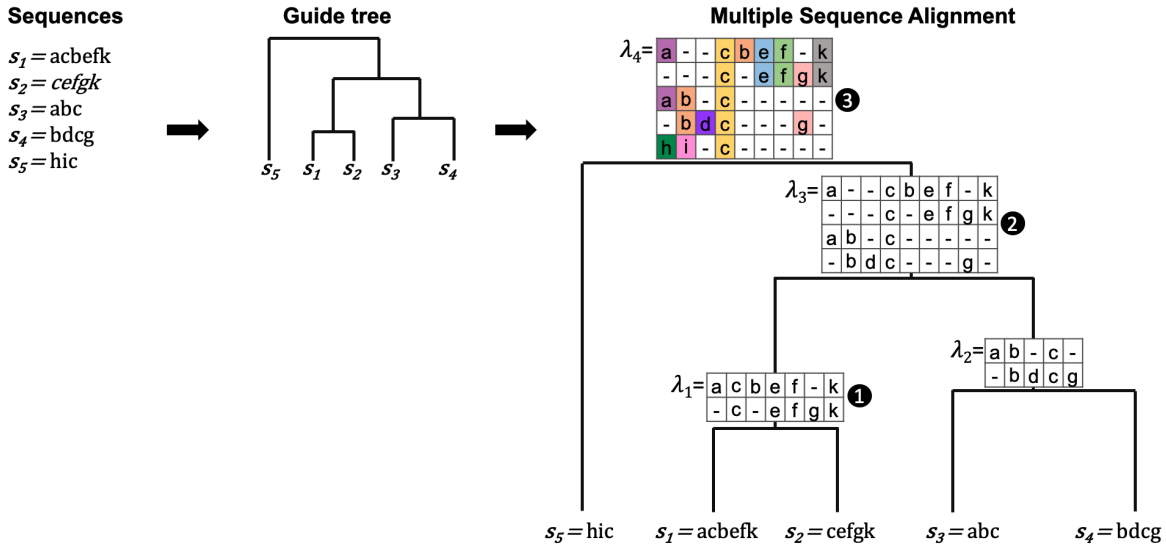


Figure 2.7: *Progressive approach for Multiple Sequence Alignment. Given a set of event sequences: firstly, a hierarchical clustering or guide tree is produced which indicates the order to pair and align sequences; secondly, by following the guide tree, it is possible to align a pair of sequences (1), align a pair of alignments (2), or align a sequence and an alignment (3). The final alignment is situated at the top node of the guide tree (i.e. λ_4).*

alignment [13].

The alignment of multiple sequences (i.e. more than two sequences) is commonly solved through dynamic programming, using an approach called Progressive Alignment [30]. The overall problem of aligning a set of sequences is broken down into three types of sub problems [11]:

1. Align two sequences (see Fig. 2.7-1), also known as pairwise alignment.
2. Align an alignment with another alignment (see Fig. 2.7-2).
3. Align a sequence with an alignment. (see Fig. 2.7-3).

Section 2.4.1 outlines the method to solve the first sub problem (pairwise alignment). Section 2.4.2 explains the method to solve the first and second sub problems and perform multiple sequence alignment through Progressive Alignment.

2.4.1 Pairwise alignment

There are two approaches to aligning a pair of sequences: local alignment and global alignment [26]. Local alignment aligns a section of the first sequence with a section of the second sequence. This is applied when the pair of sequences share similarities in only one section. Global alignment aims to align the sequences from the first to the last character or event [11]. In the application to temporal event sequences, the global alignment approach is usually used [13], as the purpose is to align the entire sequences.

The Needleman-Wunsch [71] algorithm is the standard algorithm for global pairwise alignment. This algorithm performs the alignment by using two matrices: the score matrix F and the traceback matrix T. The element i, j of the F matrix will contain a score to measure the overlap between the sub-sequences $s_1[1:j]$ and $s_2[1:i]$. The matrix T is used to track the elements in F that lead to the highest score, based on which gaps will be inserted in s_1 and s_2 accordingly. Given a pair of sequences $S = s_1, s_2$, F and T have the same dimension defined as $n \times m$, where $n = |s_2| + 1$ and $m = |s_1| + 1$. The symbol "||" denotes the number of events in a sequence. For example, for the sequences $S = \{s_1 = \text{acbef}, s_2 = \text{cef}\}$, the dimension of F and T is 4 x 6. In other words, the number of rows is given by the number of events in s_1 plus one, and the number of columns is the number of events in s_2 plus one (see Fig. 2.8).

Pairwise alignment consists of two steps: 1) compute matrices F and T, and 2) construct pairwise alignment based on F and T.

2.4.1.1 Compute score and traceback matrices

The score matrix F is computed by initialising $F[0,0] = 0$ and by filling the rest of the matrix, from left to right starting at $F[0,1]$, according to the formula [11]:

$$F[i,j] = max \begin{cases} F[i,j-1] - gap\_penalty \\ F[i-1,j-1] + substitution\_score(s_1[j], s_2[i]) \\ F[i-1,j] - gap\_penalty \end{cases} \tag{2.1}$$

$$substitution\_score(s_1[j], s_2[i]) = \begin{cases} match\_score, & \text{if } s_1[j] == s_2[i] \\ mismatch\_score, & \text{otherwise} \end{cases} \tag{2.2}$$

where i is the row number, j is the column number, the *gap_penalty* is the cost of adding a gap to either s_1 or s_2, and the substitution score depends on whether $s_2[i]$ and $s_1[j]$ are the same event type (*match_score*) or not (*mismatch_score*). Generally, *mismatch_score* should be negative to discourage different events from being aligned and *match_score* should be positive. The matrix F is filled from top left to bottom right, $F[i,j]$ takes the maximum between: 1) the value of the cell on the left ($F[i,j-1]$) minus the gap penalty; 2) the value of the cell up and left ($F[i-1,j-1]$) plus the substitution score; or 3) the value of the cell above ($F[i-1,j]$) minus the gap penalty. For example, for the sequences $S = \{s_1 = \text{acbef}, s_2 = \text{cef}\}$, $gap\_penalty = 1$, $match\_score = 3$, and $mismatch\_score = -1$, the score matrix would look as shown in Fig. 2.8. The traceback matrix T is filled along F, the element $T[i,j]$ is filled with an arrow that points towards the cell in F used to fill $F[i,j]$: left arrow $\leftarrow$ for $F[i,j-1]$, diagonal arrow $\nwarrow$ for $F[i-1,j-1]$, or up arrow $\uparrow$ for $F[i-1,j]$.

2.4.1.2 Build pairwise alignment

The aligned sequences $\bar{s_1}$ and $\bar{s_2}$ are the aligned versions of s_1 and s_2 respectively, and contain the same events as s_1 and s_2 plus zero or more gaps. Gaps are added by following the direction of the arrows in T starting from the bottom right element $T[n-1, m-1]$ until $T[0,0]$ is reached, for a given $T[i,j]$ element [11]:

1. **Up arrow** $\uparrow$: a gap (–) is added at the start of $\bar{s_1}$, and the i_{th} event of s_2 is added at the start of $\bar{s_2}$.

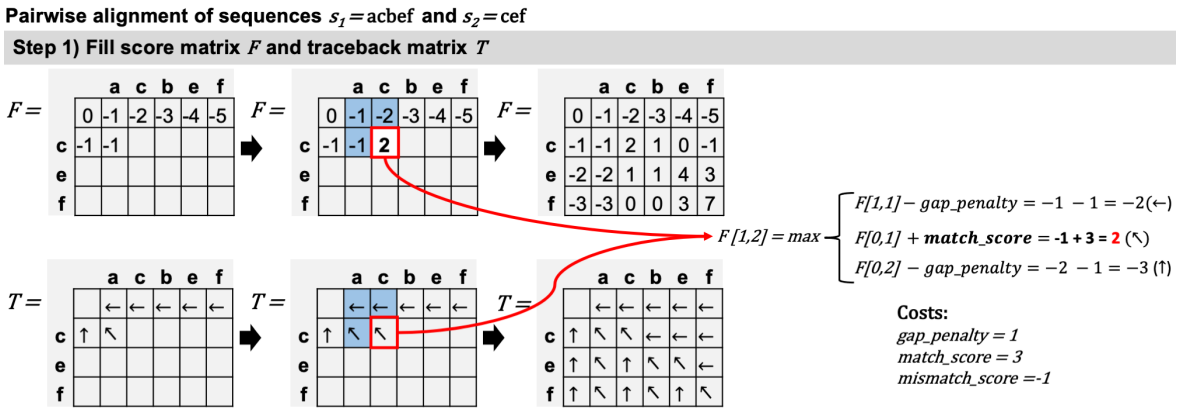


Figure 2.8: *Score matrix F and traceback matrix T for the alignment of two sequences $S = \{s_1 = acbef,\ s_2 = cef\}$. The element $F[0,0]$ is initialised with zero and the matrix is filled from top left to bottom right according to Eq. (2.1). The figure shows how the element $F[1,2]$ is computed according to the three values surrounding the element (highlighted in blue) and the chosen costs (bottom right). The traceback matrix T is filled along F, for the element $F[1,2]$, a diagonal arrow ($\nwarrow$) is added to $T[1,2]$ to indicate that the second condition of Eq. (2.1) resulted in the maximum score.*

2. **Left arrow** $\leftarrow$: a gap ($-$) is added at the start of $\bar{s_2}$, and the j_{th} event of s_1 is added at the start of $\bar{s_1}$.

3. **Diagonal arrow** $\nwarrow$: the j_{th} event of s_1 is added at the start of $\bar{s_1}$, and the i_{th} event of s_2 is added at the start of $\bar{s_2}$.

Fig. 2.9 illustrates the path followed in T (red arrows) and the insertion of gaps/events to $\bar{s_1}$ and $\bar{s_2}$, given the example sequences $S = \{s_1 = \text{acbef}, s_2 = \text{cef}\}$. The resulting pairwise alignment is $\bar{s_1} = \text{acbef}, \bar{s_2} = \text{-c-ef}$.

2.4.2 Progressive alignment for multiple sequence alignment

The pairwise alignment method in section 2.4.1 can be generalised so that it is applied not only to a pair of sequences but also to the alignment of two alignments or to one sequence and an alignment [11].

Following a progressive alignment approach: firstly, the pairs of most similar sequences are aligned, then, each pairwise alignment is aligned to the alignment of its closest pair of sequences or to a single sequence until a single alignment containing all the sequences is obtained. A *guide tree* is a hierarchical structure that indicates the order in which sequences are aligned, usually obtained through hierarchical clustering (see section 2.3). As observed in Fig. 2.7, the alignment of a node (e.g. λ_3) is based on the alignment of its child nodes (e.g. λ_1 and λ_2).

To allow the alignment of two alignments or a sequence and an alignment, Eq. (2.1) is

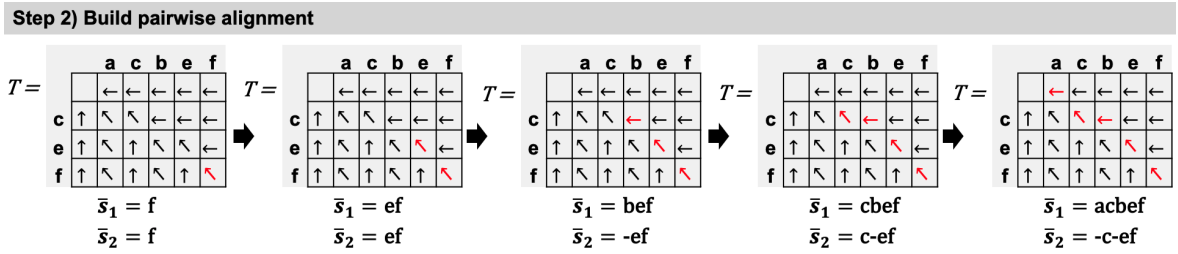


Figure 2.9: *Building pairwise alignment according to the traceback matrix T. $\bar{s_1}$ and $\bar{s_2}$ are the aligned version of $s_1 = acbef$, and $s_2 = cef$ respectively. Starting from the bottom right cell, depending on the arrow at $T[i,j]$, a gap or event are added to $\bar{s_1}$ and $\bar{s_2}$. The next position in T is selected by following the direction of the current arrow (in red). The pairwise alignment is finished when $T[0,0]$ is reached. A gap (–) is added to the start of $\bar{s_1}$ if only if $T[i,j] == \uparrow$, otherwise the j_{th} event of s_1 is added. Similarly, a gap (–) is added to the start of $\bar{s_2}$ if only if $T[i,j] == \leftarrow$, otherwise the i_{th} event of s_2 is added.*

rewritten to the following:

$$F[i,j] = max \begin{cases} F[i,j-1] - gap\_penalty \\ F[i-1,j-1] + substitution\_score\_multiple() \\ F[i-1,j] - gap\_penalty \end{cases} \tag{2.3}$$

$$substitution\_score\_multiple() = \sum_{a=1}^{|\lambda_1|} \sum_{b=1}^{|\lambda_2|} \frac{substitution\_score(\lambda_1[a,j], \lambda_2[b,i])}{|\lambda_1| \times |\lambda_2|} \tag{2.4}$$

where λ_1 is the first alignment, λ_2 is the second alignment, $\lambda_1[a,j]$ is the event at the a_{th} row and j_{th} column of λ_1, $\lambda_2[b,i]$ is the event at the b_{th} row and i_{th} column of λ_2, and $|\lambda_1|$ and $|\lambda_2|$ are the number of rows in λ_1 and λ_2 respectively. The $substitution\_score\_multiple()$ function computes the average $substitution\_score$ for all events in the j_{th} column of λ_1 versus all events in the i_{th} column of λ_2, where the $substitution\_score$ of two events is computed as per Eq. (2.2). The matrix T is filled along F as per the previous section 2.4.1.1. After computing matrices F and T, the matrix T is traced and the alignment is built as per section 2.4.1.2. For clarity, the rules to build a new alignment from T are rewritten as follows. For each position $T[i,j]$, a new column is inserted to the left of the first column of the new alignment λ_3, and is filled with:

1. **Up arrow** $\uparrow$: a gap (–) is added in the new column for the rows corresponding to λ_1, and the events in the i_{th} column of λ_2 are added to the new column.

2. **Left arrow** $\leftarrow$: a gap (–) is added at the rows corresponding to λ_2, and the events at the j_{th} column of λ_1 are added to the new column.

3. **Diagonal arrow** $\nwarrow$: the events at the j_{th} column of λ_1 and the events at the i_{th} column of λ_2 are added to the new column.

Fig. 2.10 shows the matrices F and T for two example alignments and illustrates how to trace the matrix T to build their alignment.

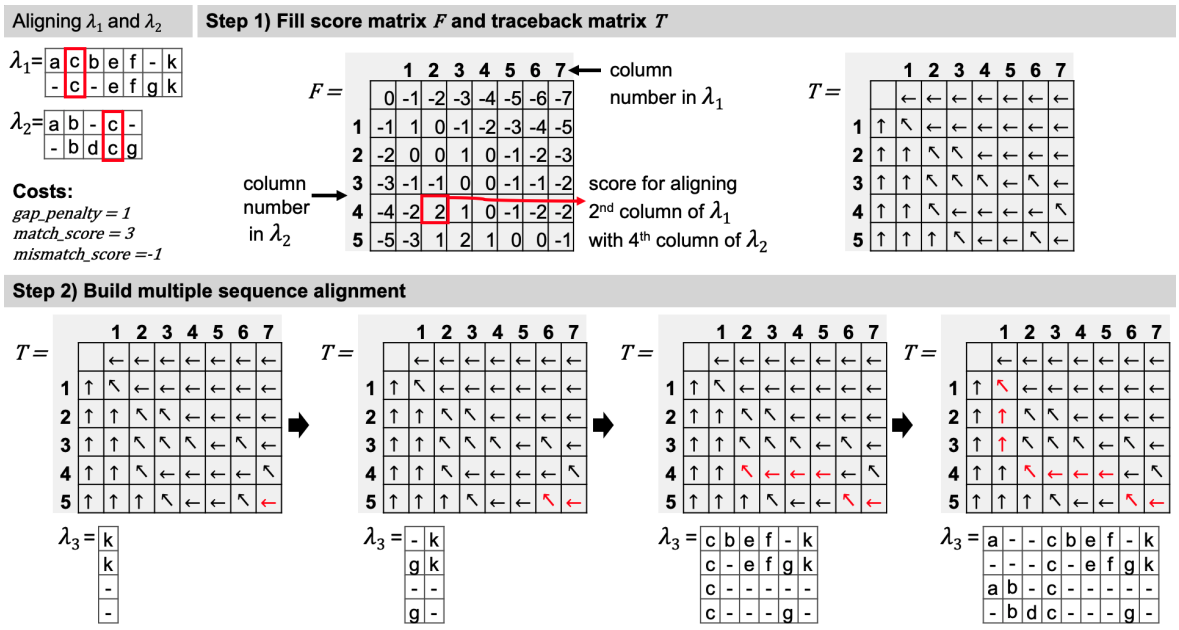


Figure 2.10: *Aligning two alignment matrices λ_1 and λ_2 to obtain λ_3. Step 1) the matrices F and T are computed as per Eq. (2.3). Step 2) The multiple alignment λ_3 is obtained by following the direction of the arrows in T (highlighted in red), starting from the bottom right element.*

2.5 Summary

This chapter introduced the challenges of visual analytics in the healthcare domain (section 2.1), presented an overview of visualisation techniques for temporal event data (section 2.2), and reviewed two techniques in which this thesis is based: sequence clustering (section 2.3) and multiple sequence alignment (section 2.4).

The importance of visual analytics in the context of healthcare was introduced. In one hand, huge amount of data are generated everyday in healthcare settings, on the other hand there is a shortage in analytical capability to interpret such data. Visual analytics has the potential of tackling this shortfall and support quality improvement in healthcare. The current challenges of visual analytics in healthcare were presented, these include: 1) data scale and complexity, 2) data quality and uncertainty, 3) scalable analysis, and 4) user interaction and interfaces. The present work focuses on the first, third, and fourth challenges.

Creating a visual summary or overview of event data is a challenging task given its complexity in terms of sequence length, number of records, and number of event types. Techniques to create an overview of event data can be classified into four groups: explicit summarisation, inexplicit summarisation, progression analysis, and clustering. The present work falls in the category of clustering. According to this literature review, there is no current technique to explore different sequence clusterings and that, at the same time, provides a clear interpretable representation for each cluster. This thesis aims to tackle these two gaps.

Clustering is a technique that aims to group similar objects into clusters, being k-means and hierarchical clustering the two most popular approaches. The present work adopts an agglomerative hierarchical clustering approach, where similar objects are iteratively merged

until one cluster containing all the objects is obtained. A comparison of string distance metrics was presented, these include: hamming distance, longest common subsequence, the Levenshtein distance, and q-gram distance. The q-gram distance was adopted for the case studies presented in this work.

Multiple Sequence Alignment (MSA) was initially proposed to align biological sequences, but it has been adopted in the literature to align event sequences and discover common behaviour. The objective of MSA is to insert gaps (-) in the input sequences so that the number of equal events column-wise is maximised. Aligning a set of sequences is commonly solved through dynamic programming, using a progressive alignment approach [30], in which the problem is broken down in three sub problems: align two sequences, align two alignments, and align a sequence with an alignment. These alignments are achieved by using two matrices: the score matrix F and the traceback matrix T. Figures 2.10 and 2.9 were presented, to illustrate how these matrices are filled while gaps are inserted in the input sequences to obtain a new alignment.

The hierarchical clustering and MSA algorithms are further explored in Chapter 4, where these are applied in the proposed cluster representation.

Chapter 3

Analytic Tasks

The visualisation technique proposed in this thesis was implemented into a visual analytics system called Sequen-C, which enables users to carry out the analytic tasks presented in this chapter. The tasks were defined taking into account the most common tasks in event data analysis according to the literature review, and through a series of interviews and feedback sessions with three distinct groups of stakeholders in the clinical domain. These stakeholders are the same people who participated in the case studies presented in this thesis. The stakeholder groups were composed by six people: two data and clinical software architects, two clinical scientists, and two clinicians.

All stakeholders shared a common goal: understanding their internal operation and identifying operational problems, to optimise the delivery of healthcare to patients. Despite the fact that stakeholders are all in the healthcare domain, in these sessions, the aim was to produce a list of analytic tasks that could also be generalised to other application domains. At the start of this project, Sheffield Teaching Hospitals (STH) provided two datasets from the Rheumatology and the Antenatal Care Unit outpatient departments. Multiple meetings were held between the two stakeholders from STH, the PhD student and her supervisor. In these meetings, the stakeholders explained the real-world context of the datasets and discussed the aspects of the data they wanted to visualise, including waiting times and the flow of patients through the different workflows. Based on these meetings: an initial set of analytic tasks were derived, prototypes of the visualisations and a mock-up of the system were developed to preview how the tasks would be implemented. The development of Sequen-C was performed iteratively, a new version of the system would be presented to the stakeholders every couple of weeks, and changes were made based on their feedback. After several months of working with STH, the Centre for Urgent and Emergency Care Research (CURE) provided the CUREd dataset. This dataset was visualised using an early version of Sequen-C. The produced visualisations were presented to the stakeholders from CURE, who provided feedback about what other aspects of the data they wanted to explore, which resulted in additional analytic tasks.

The following analytic tasks were obtained:

T1. Explore common and deviating pathways: help users to explore and discover a set of common pathways that best summarise the dataset, while also being able to identify deviating pathways.

T2. Interpret the sequences that constitute a cluster: the visualisation should allow users to interpret and compare the sequences in a cluster, as well as identifying differences across clusters.

T3. Focus the analysis on a selected set of records: allow queries in the dataset to focus on sequences with specific characteristics.

T4. Obtain details on demand: provide coordinated views so that users can request finer details of interesting items in the overview. Users should be able to go from the highest level of aggregation (i.e. clusters), passing through sequences grouped by their unique sequence, to individual sequences and events including their raw data (e.g. attributes, timestamp and duration).

T5. Aggregate and compare context information for selected groups of records: the system should allow to aggregate and compare data attributes (e.g. age, gender, country) for selected clusters, unique sequences, or individual sequences.

T6. Compare the distribution of attributes within and across sequential patterns: Explore the distribution of temporal and multivariate attributes within and across sequential patterns. Enable users to establish relations between a sequential pattern and the distribution of an attribute, for example if the duration of an event is longer in a sequential pattern in comparison to another sequential pattern.

T7. Identify trends involving multiple variables such as duration, time of occurrence, and categorical attributes: The proposed visualisation should allow users to identify trends of duration through time. For example, if the duration of an event increases as the time goes by or if longer durations occur only in the early hours or a specific day. Also, one may relate these trends with a selected categorical attribute (e.g. gender, age, country).

T8. Identify anomalous scenarios: Identify outliers such as infrequent sequential patterns or data points with an anomalous duration or time of occurrence.

The analytic tasks listed above are later referenced in sections 4.5 and 5.4 of Chapters 4 and 5 respectively, where it is explained how the tasks are implemented on the system Sequen-C.

Chapter 4

Sequen-C: a multilevel overview of temporal event sequences

Building a visual overview of temporal event sequences with an optimal level-of-detail (i.e. simplified but informative) is an ongoing challenge - expecting the user to zoom into every important aspect of the overview can lead to missing insights. This chapter proposes a technique to build and explore a multilevel overview of event sequences, whose granularity can be transformed across sequence clusters (*vertical level-of-detail*) or longitudinally (*horizontal level-of-detail*), using hierarchical aggregation and a novel cluster data representation *Align-Score-Simplify*. By default, the overview shows an optimal number of sequence clusters obtained through the average silhouette width metric – then users are able to explore alternative optimal sequence clusterings. The vertical level-of-detail of the overview changes along with the number of clusters, whilst the horizontal level-of-detail refers to the level of summarisation applied to each cluster representation. The proposed technique has been implemented into a visualisation system called Sequence Cluster Explorer (Sequen-C) that allows multilevel and detail-on-demand exploration through three coordinated views, and the inspection of data attributes at cluster, unique sequence, and individual sequence level. This chapter presents two case studies using real-world datasets in the healthcare domain: 1) Antenatal Care Unit and 2) MIMIC-III. Moreover, Chapter 6 presents an extended case study where this technique is evaluated using a real-world dataset containing calls made to emergency services. The case studies demonstrate how the technique can aid users in exploring and defining a set of distinct pathways that best summarise the dataset, while also being capable of identifying deviating pathways and exploring data attributes for selected patterns.

This chapter is based on the publication [60]:

Magallanes, J., Stone, T., Morris, P. D., Mason, S., Wood, S., and Villa-Uriol, M. C., 2021. Sequen-C: A Multilevel Overview of Temporal Event Sequences. IEEE Transactions on Visualization and Computer Graphics, in press.

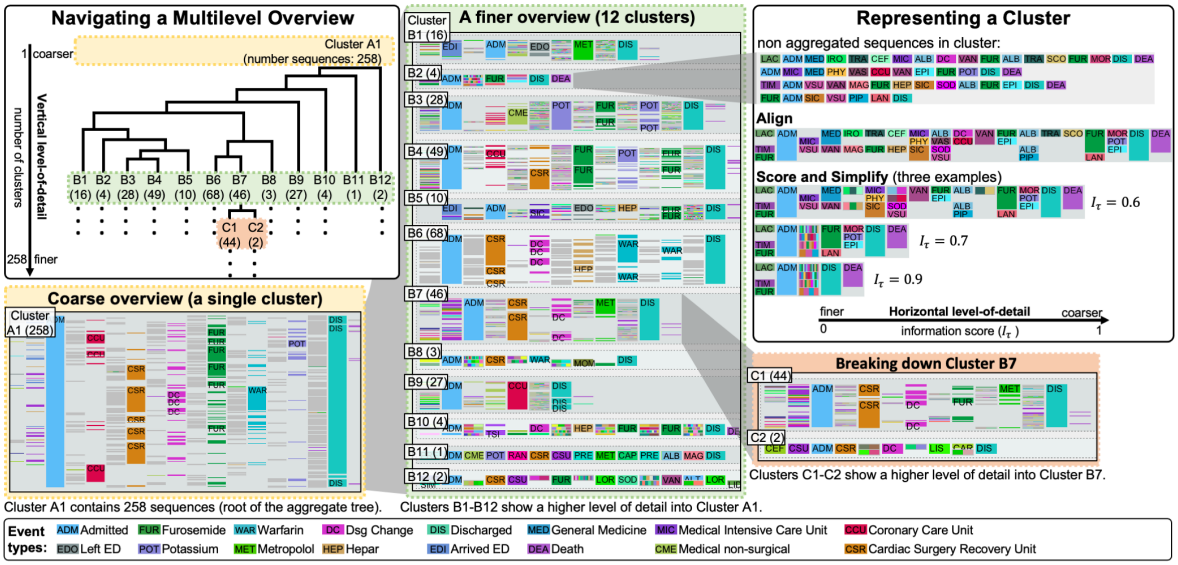


Figure 4.1: *Screenshots of Sequen-C for a dataset with 258 event sequences illustrating the methodology. The hierarchical aggregation tree (top left) allows changing the number of clusters shown. The vertical level-of-detail of the multilevel overview can be transformed from coarse (bottom left) to fine (middle and bottom right). Sequence clusters are represented using an Align-Score-Simplify strategy (top right), which allows controlling the horizontal level-of-detail according to the information score threshold (I_τ).*

4.1 Introduction

Visual analytics of temporal event sequence data has applications in various domains such as electronic health records (e.g. [35; 39]) and web clickstream analysis (e.g. [58; 96]). This type of data usually presents high variability and volume. Existing visual analytic techniques obtain a visual summary (overview) of event sequences using techniques such as sequential pattern mining or sequence clustering, with the purpose of understanding common and deviating pathways. These techniques commonly follow the information-seeking mantra: "overview first, zoom and filter, then details on demand" [84]; which means that the starting point of the exploration is the given overview.

Finding the optimal level-of-detail of the initial overview, simplified but informative, is an ongoing challenge. Current visualisation systems provide an overview of event sequences where the level-of-detail can be transformed by drilling down certain visualisation elements, but do not provide dynamic levels of detail across both sequences and longitudinally. At the same time, there is no existing technique that allows users to explore different sequence clusterings to obtain a set of distinct pathways that best describe the data.

This thesis presents a technique to build and explore a multilevel overview of event sequences through hierarchical aggregation and the cluster data representation Align-Score-Simplify, in which users can interactively transform the overview from a coarse level-of-detail to a fine one, allowing a seamless analysis of alternative overviews.

The multilevel overview presents a given number of sequence clusters k retrieved from a hierarchical aggregation or *aggregate tree* of event sequences. Such aggregate tree is built

using a bottom-up approach [1], in which the leaves of the tree contain the original sequences and each node corresponds to a sequence cluster. A data representation to summarise the sequences in a cluster is built using the steps Align-Score-Simplify (see Fig. 4.3). First, Multiple Sequence Alignment (MSA) [30] is used to align the sequences in each cluster (Align). Secondly, an information score of each column in the alignment matrix is computed (Score). Thirdly, the alignment is simplified by merging consecutive columns with a low information score (Simplify). The alignment of sequences aims to represent the common events in a sequence cluster to allow the comparison of commonalities and deviations within and across clusters.

The aggregate tree along with the proposed data representation allow one to transform the level-of-detail of the overview vertically and horizontally. The *vertical level-of-detail* is directly proportional to the number of clusters k in the overview. The higher the number of clusters, the finer the detail. For example, in Fig. 4.1 one cluster ($k = 1$) will show the coarsest summary for the dataset, whereas twelve clusters ($k = 12$) will show details that were not visible when visualising a single cluster. In the example, users are able to change k between 1 and 258 (the number of original sequences in this dataset). *The horizontal level-of-detail* depends on the information score threshold I_τ which controls the level of simplification applied to the clusters in the overview. That is, a higher value of I_τ means the clusters show a coarser representation.

By default, the overview presents the best clustering according to the Average Silhouette Width metric (ASW) [47], then users can interactively change the number of clusters. In order to facilitate that exploration, a subset of alternative optimal values of k is obtained from the ASW curve, which represent alternative good overviews.

This thesis presents Sequence Cluster Explorer (Sequen-C), a visual analytics framework where multilevel and detail-on-demand exploration of temporal event sequences is possible through three coordinated views: *multilevel overview*, *unique sequence view*, and *individual sequence view*. Moreover, multivariate data attributes can be inspected at cluster, unique sequence and individual sequence level using bar charts.

The technique allows the exploration of interesting sequence clusterings, regardless of permutations in the order of events, to define a set of pathways that best summarise the event sequences. The benefits of this technique are demonstrated using two real-world medical datasets: 1) MIMIC-III (see Fig. 4.9) and 2) Antenatal Care Unit (see Fig. 4.12). The contributions of the present work are:

- A technique to build and explore a multilevel overview of event sequences, from coarse to fine *vertical or horizontal level-of-detail*, using hierarchical aggregation and a novel cluster data representation using an *Align-Score-Simplify* strategy.
- A novel approach to explore sequence clusterings, where the most optimal and alternative optimal number of clusters are provided.
- A visual analytics system called Sequen-C that implements the proposed multilevel overview, and allows detail-on-demand exploration and the inspection of multivariate data attributes at cluster, unique sequence, or individual record level.
- Two case studies using two real-world datasets, that demonstrate the technique in obtaining a set of distinct pathways to summarise the event sequences and inspecting the characteristics of such pathways.

4.2 Related work

This chapter presents a technique to build and explore a multilevel overview of event sequences using sequence clustering and sequence alignment. Chapter 2 provides a comprehensive review of current techniques to create an overview of event sequences (section 2.2), sequence clustering (section 2.3) and sequence alignment methods (section 2.4).

According to the literature review, Vasabi and Sequence Synopsis are the techniques most similar to this work. Vasabi [72] builds an overview of sequence clusters by first extracting the most common events (e.g. tasks) in the dataset and then clustering sequences using those events as features of the clustering. The technique successfully extracts and represents a fixed number of sequence clusters. However, their cluster representation does not allow one to identify event permutations or see the events that were omitted in the event extraction step, and the number of clusters cannot be changed. Sequence Synopsis [19] clusters event sequences based on the minimum description length, in which clusters are represented by a sequential pattern and a set of corrections. As indicated by Chen *et al.* [19], a potential drawback of their cluster representation is that missing events are not explicitly encoded and that scalability could be improved by supporting hierarchical visual summary (e.g. explore alternative number of clusters). To the best of our knowledge, there is no existing technique to explore different sequence clusterings that at the same time provides an interpretable representation of the sequences in a cluster. The present work aims to tackle this problem.

4.3 A multilevel overview of event sequences

We propose a technique to build and explore a multilevel overview of event sequences through hierarchical aggregation. A multilevel overview is a visual summary which can be interactively transformed from coarse to fine level-of-detail [27]. The multilevel overview presented in this thesis displays a given number of sequence clusters retrieved from a hierarchical aggregation or *aggregate tree*; where each sequence cluster is summarised and represented using the steps Align-Score-Simplify.

The overview can be transformed vertically and horizontally. The vertical level-of-detail is interactively controlled with the number of clusters retrieved from the tree. Fig. 4.2-C shows how the higher in the hierarchy (i.e. smaller number of clusters), the coarser the overview; whereas the lower in the hierarchy (i.e. larger number of clusters), the finer the details provided. On the other hand, the horizontal level-of-detail refers to the level of simplification of each cluster representation according to its information score (see Fig. 4.3). By default the overview presents the most optimal number of clusters obtained using the overall average silhouette value [81]. However, we also offer a set of alternative optimal number of clusters that might result in good alternative overviews of the data. The algorithm to build the aggregate tree is presented in subsection 4.3.1, the cluster data representation is explained in subsection 4.3.2, and last, subsection 4.3.3 describes how to obtain alternative optimal number of clusters.

4.3.1 Building the aggregate tree

To build the aggregate tree from the input unique temporal event sequences, we use a bottom-up aggregation approach [1]. Every input unique sequence starts in a single cluster, then pairs

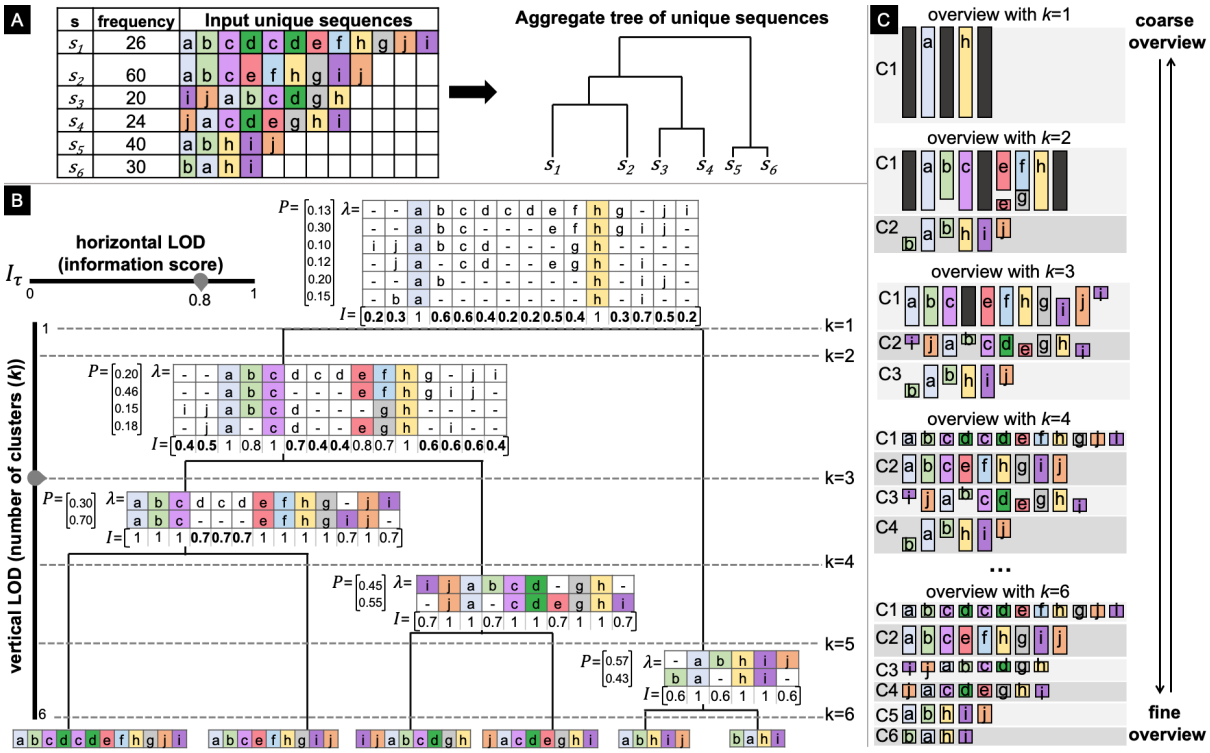


Figure 4.2: *(A) Building aggregate tree T for input unique sequences $S = \{s_1, ...s_6)\}$. (B) Each node in T has an alignment matrix λ for its child sequences, a row-wise probabilities vector P, and a column-wise information score vector I. Two or more consecutive columns in λ with $I_j < 0.8$ are not coloured. (C) Multi-level overviews for a range of number of clusters k retrieved from T, where black blocks represent merged columns.*

of similar clusters are iteratively aggregated until a single cluster is obtained. The leaf nodes of the final tree contain the input unique sequences, each internal node in the tree contains an aggregated representation of their child nodes, and the root node aggregates the whole dataset (see Fig. 4.2-B).

Algorithm 1 shows in detail how to build the aggregate tree T. The algorithm receives as input the list of unique sequences $S = \{s_1, s_2, ..., s_N\}$ in the dataset, being N the number of input unique sequences in the dataset. In the last iteration, T is the root node of the tree. As T is a binary tree, all possible sub-trees can be obtained from T by retrieving its left and right children recursively until the leaf nodes are reached.

The algorithm starts by initialising the distance matrix d. The method D_{cqgram} returns an $N \times N$ matrix containing the pairwise distances of all the input unique sequences in S. The pairwise distance of two sequences is computed using the cosine distance of their q-gram profiles (i.e. cosine q-grams) [92], where the q-gram profile of a sequence is the vector of all sub-sequences of q consecutive events. The implementation of the `stringdist` package in R [94] was used, with $q = 1$. Using $q = 1$ allows to cluster sequences according to the count of shared events regardless of their permutations in order. For example, the distance between sequences "abcde" and "deabc" is zero and both sequences are likely to end up in the same

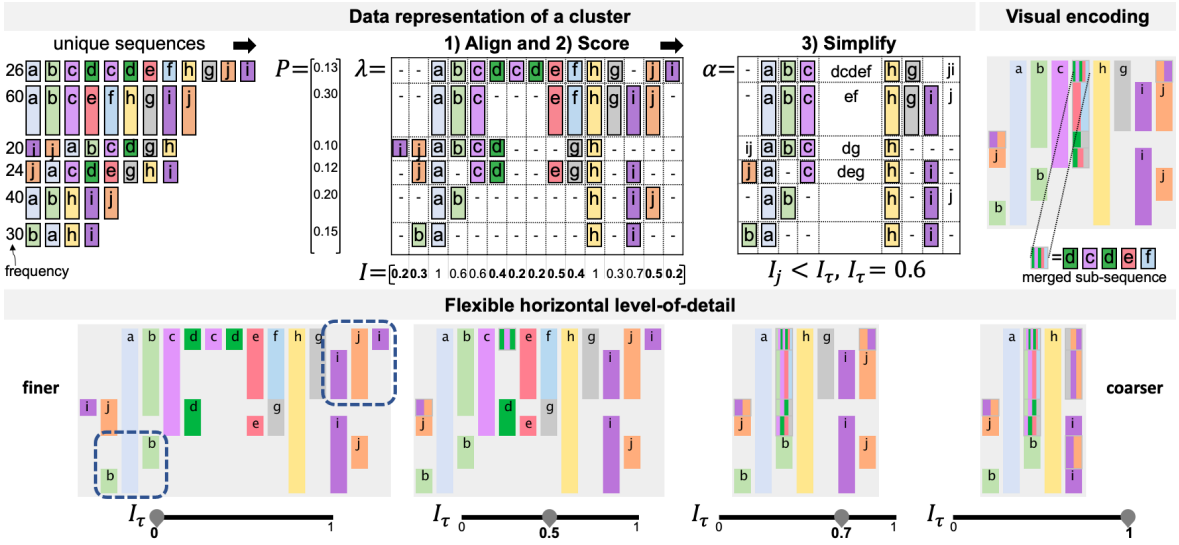


Figure 4.3: *Cluster data representation (top left) and visual encoding (top right) for $S = \{s_1, ...s_6\}$ and $I_\tau = 0.6$ using the Align-Score-Simplify strategy. The Simplify step produces the simplified alignment matrix α. Using α, the visual encoding is built and event types in the merged sub-sequences are represented using ordered coloured bars. Changing I_τ (bottom) transforms the horizontal level-of-detail. Note how this shows the most common event orderings (e.g. a-b-c-h) and permutations in the order (e.g. b-a and j-i highlighted with dashed lines).*

cluster. See subsubsection 4.3.1.1 for a discussion on the choice of distance metric.

A node in the aggregate tree is defined as $Node(l, r, \lambda, \alpha)$; where l is the left child node, r the right child node, λ the alignment of the sequences at that node, and α the simplified version of λ which is used as the representation of the cluster. The second step (line 3) in the algorithm is to initialise the set T with the leaf nodes. For each input unique sequence s in S, a leaf node is defined as $Node(l = \emptyset, r = \emptyset, \lambda = \alpha = s)$; where the children nodes l and r are empty, and the alignment λ and cluster representation α are the sequence s itself.

In line 4 of Algorithm 1, the iterative process stops when T contains a single node. Each iteration consists of two steps. First, a new node $n_{a \cup b}$ is created by merging the closest pair of nodes (n_a, n_b), and second, d and T are updated according to the newly created node.

The closest pair of nodes (n_a, n_b) is that for which the value in the distance matrix d is the minimum (line 5). The method *aggregate* returns a new node $n_{a \cup b}$ whose left child is n_a and right child is n_b, with alignment λ and a summarised representation α of the sequences in n_a and n_b. Algorithm 2 outlines how to build the cluster data representation of the newly aggregated node $n_{a \cup b}$.

Lines 7 to 10 of Algorithm 1 update d and T by adding the new node $n_{a \cup b}$ and removing the nodes n_a and n_b used for the aggregation. The distance matrix d is updated by inserting the pairwise distance from the new node $n_{a \cup b}$ to all nodes in T (line 7), and by removing from d the i_{th} row and j_{th} column containing n_a and n_b. The method D_{cqgram} in line 7 computes the distance between two clusters using the average agglomeration method [1], which defines the distance between two clusters as the average of all pairwise distances between the sequences in both clusters.

Algorithm 1: Build aggregate tree

Data: input unique sequences $S = \{s_1, s_2, ..., s_N\}$
Result: aggregate tree T

Function `buildAggregateTree(`S`):`
    `/* initialise pairwise distance matrix` d `*/`
    $d[i, j] = D_{cqgram}(s_i, s_j)$; $\forall$ pairs, $s_i, s_j \in S$
    `/* initialise` T `as a list of leaf nodes */`
    $T = \{Node(l, r, \lambda, \alpha) \mid l = \emptyset, r = \emptyset, \lambda = \alpha = s, \ \forall\, s \in S\}$;
    `/* loop until one node remains (root node) */`
    while $|T| > 1$ **do**
        `/* aggregate the closest pair of nodes */`
        $(n_a, n_b) = \text{argmin}\, d[n_i, n_j]$; $\forall$ pairs, $n_i, n_j \in T$
        $n_{a \cup b} =$ `aggregate`(n_a, n_b);
        `/* update distance matrix */`
        $d[n_{a \cup b}, n] = d[n, n_{a \cup b}] = D_{cqgram}(n_{a \cup b}, n)$; $\forall\, n \in T$
        remove distances containing n_a and n_b from d;
        `/* update nodes set` T `*/`
        remove n_a and n_b from T;
        add $n_{a \cup b}$ to T;
    end
return T

4.3.1.1 Distance metric

The cosine q-grams, used as the distance metric in Algorithm 1, does not take into account event order, which represents a limitation in certain datasets. However, in the context of the case studies presented in this thesis, this metric was chosen to ensure that sequences with a similar set of event types are in the same cluster, regardless of event permutations, as opposed to enforcing a strict event ordering within clusters. Note that event order is not completely ignored, as the sequence alignment (MSA) will ensure the most common ordering is visualised in the cluster representation (see Fig. 4.3).

The chosen distance metric works well in situations where there are common events across sequences that act as milestone events, or in situations when certain events are semantically associated and will have a natural order between them. However, the current choice of distance metric will not work, or a different distance metric will be necessary, in cases where the interest is in studying the impact of event order in the outcome. For example, in the context of patients in cardiac arrest, event order is important when investigating whether the order of certain diagnostic tests impact patient outcome (e.g. length of stay, death), such as an echocardiogram being performed before an angiogram. In these cases, other distance metrics such as the Levenshtein distance [52] could be used (see subsection 2.3.1 in Chapter 2 for a full definition and comparison of the available distance metrics). Nevertheless, a disadvantage of the Levenshtein distance is that it can be affected by: permutations introduced by unimportant events, the length of sequences, or repetitions of sub-sequences. For example, the sequences $s_1 = abc$ and $s_2 = abcabc$ have a Levenshtein distance of 3, whereas the sequences

$s_1 = abc$ and $s_3 = hij$ also have a Levenshtein distance of 3. As observed, s_1 is closer related to s_2 than it is to s_3. Future work is needed to propose a distance metric that addresses these limitations, and accounts for event importance and event categorisation according to their context.

Algorithm 2: Data representation of a cluster

Data: nodes n_a and n_b
Result: aggregated node $n_{a \cup b}$
Function `aggregate`(n_a, n_b)**:**
    `/* Align: compute` λ `from children nodes */`
    $\lambda =$ `MSA`$(n_a.\lambda, n_b.\lambda)$;
    `/* Score: column-wise information score` I `*/`
    for $j \leftarrow 1$ ***to*** m **do**
        Compute I_j according to Eq. (4.1) and Eq. (4.2)
    end
    `/* Simplify: collapse columns based on` I `*/`
    $listRemove = \emptyset$;
    for $j \leftarrow 1$ ***to*** $m - 1$ **do**
        if $I_j < I_\tau$ *and* $I_{j+1} < I_\tau$ **then**
            for $i \leftarrow 1$ ***to*** n **do**
                $\lambda_{i,j+1} =$ `concatenate`$(\lambda_{i,j}, \lambda_{i,j+1})$;
            end
            add j to *listRemove*;
        end
    end
    `/* assign the simplified alignment to` α `*/`
    $\alpha =$ delete columns in *listRemove* from λ;
    `/* create new node` $n_{a \cup b}$ `*/`
    $n_{a \cup b} = Node(n_a, n_b, \lambda, \alpha)$;
return $n_{a \cup b}$

4.3.2 Cluster data representation: Align-Score-Simplify

We propose a data representation to aggregate and represent the sequences in a cluster using three steps: 1) Align, 2) Score, and 3) Simplify. Firstly, the sequences in a given cluster are aligned using Multiple Sequence Alignment (MSA) [30]. Secondly, an information score is computed for each column in the alignment matrix. Finally, the columns in the alignment matrix with an information score below a threshold are merged to simplify the representation of the cluster. Fig. 4.3 illustrates these steps for a set of input unique sequences.

4.3.2.1 Align

To perform the alignment step and obtain the alignment matrix λ, the input unique sequences in S are formatted as sequences of characters, where each character represents an event type. All elements in λ are either single characters or gaps (–). The objective of the MSA algorithm

is to insert gaps (–) in those input sequences so that the number of equal events column-wise is maximised. To allow this, a cost is assigned to the insertion of gaps (typically known as gap open penalty) and equal column-wise events are encouraged using a gap substitution score.

We use the progressive approach [30] to carry out the multiple alignment of sequences, which iteratively constructs a series of pairwise alignments by following a tree that represents the similarity between sequences, where the alignment of a node is built using the alignment of its child nodes (see Fig. 4.2-B). This progressive approach is explained in detail in section 2.4 of Chapter chapter 2. Alignments can be constructed over a pair of sequences, a sequence and an alignment, or a pair of alignments. For a given set of sequences S, the matrix of alignment λ contains all the sequences in S aligned. The dimensions of matrix λ are $N \times M$, where N is the number of input unique sequences and M the length of the final alignment [13]. The method MSA in line 14 of Algorithm 2 returns the alignment matrix λ for the sequences in the new node, computed using the alignments of its child nodes $n_a.\lambda$ and $n_b.\lambda$. The '.' in $n_a.\lambda$ and $n_b.\lambda$ denotes that n_a and n_b contain an instance of λ.

In this work, the alignment algorithm was written in R and was based on the implementation of the Python library `scikit-bio` [89], using $gap\_open\_penalty = 0.8$, $match\_score = 3$ for equal events, and $mismatch\_score = -1$ for non equal events. These parameters were chosen empirically, by testing different values with the datasets of the case studies. The factors considered were an adequate alignment of frequent events across sequences, whilst providing a balance in the number of gaps inserted. The same parameter values were used in all the case studies presented in this thesis, however these might need to be adjusted in the future, depending on the dataset or on the focus of the alignment.

As observed in Fig. 4.4, the number of gaps and columns containing equal/non-equal events in an alignment, will vary depending on those parameters with the absolute largest value:

- ***gap_open_penalty***: a large value of this parameter will make it more *expensive* to add gaps, resulting in smaller alignments (e.g. Fig. 4.4-B), whilst a low value will result in more gaps being inserted.
- ***match_score***: a large value will encourage equal events to be in the same column, resulting in more columns with equal events (as long as *gap_open_penalty* is low enough).
- ***mismatch_score***: a large absolute value of this parameter will discourage different events from being in the same column, resulting in more gaps inserted. For example, in Fig. 4.4-C, a gap is inserted in s_1, between the events x and e, to avoid e and d from being in the same column (see highlighted in red).

4.3.2.2 Score

Once the alignment matrix λ for all the input unique sequences in a cluster (or node) is obtained, this alignment is simplified to obtain the simplified alignment matrix α. This simplification will be performed according to the *information score* for each column in λ. Based on [13], we define the information score I_j for column j in matrix λ as:

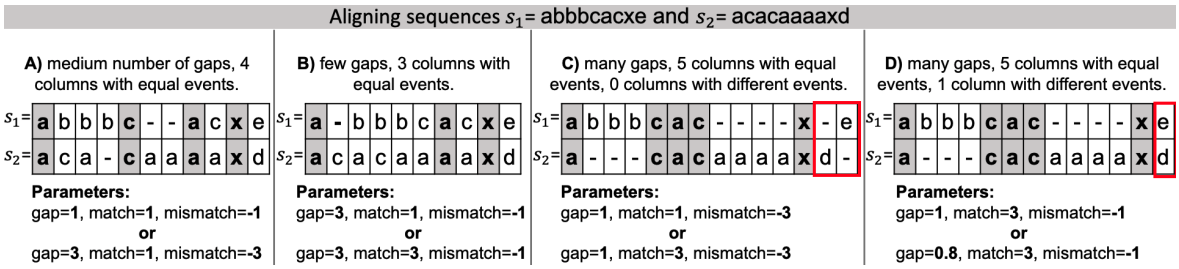


Figure 4.4: *Four alignment matrices obtained through the MSA algorithm, using eight value combinations of the parameters gap_open_penalty (gap), match_score (match), and mismatch_score (mismatch). These are varied so either one or two parameters contain the highest value - which determine the number of gaps, and columns containing equal or non-equal events. For example, matrix (A) shows the alignment obtained when the three parameters have the same absolute value ($gap = 1$, $match = 1$, $mismatch = -1$), or when gap_open_penalty and mismatch_score are the highest ($gap = 3$, $match = 1$, $mismatch = -3$). Columns highlighted in grey are those whose event types match. The red square in (C) and (D) compare the effect of increasing the absolute value of mismatch_score.*

$$I_j = 1 - \frac{E_j}{\log_2(|A| + 1)} \tag{4.1}$$

being A the set of unique event types in the alignment matrix λ, $|A|$ the length of A, and E_j the entropy of the event types in that column:

$$E_j = \sum_{a \in A_j \cup \{-\}} \begin{cases} -P_a \log_2\left(\dfrac{P_a}{G_j}\right), & \text{if a = '–'}, \\ -P_a \log_2(P_a), & \text{otherwise.} \end{cases} \tag{4.2}$$

G_j is the count of gaps in column j, P_a is the probability of the event type a in that column, and A_j is the set of unique event types in column j. To avoid I_j from becoming negative, when $E_j > \log_2(|A|+1)$, $E_j = \log_2(|A|+1)$. The probability P_a is computed as the sum of probabilities of the unique sequences to which a belongs to, defining the probability of a unique sequence as its frequency divided by the total frequency of the cluster. Fig. 4.3 shows the row-wise probability vector P for the example unique sequences and the column-wise information score I for each alignment matrix λ.

The information score provides a measure of how homogeneous a column is, with values in the range $0 \leq I_j \leq 1$. If the column contains mostly a single type of event, the information score is closer to one. If the column contains mostly gaps or many distinct event types, the information score is closer to zero. The information score vector I is computed in Lines 15-17 of Algorithm 2 according to Eqs. (4.1) and (4.2).

4.3.2.3 Simplify

The simplification of an alignment matrix λ consists of merging columns with a relatively low information score. Lines 19 to 26 of Algorithm 2 outline the iterative process used to

simplify an alignment λ to obtain the simplified cluster representation matrix α. The matrix α is $N \times M'$, where M' is the length of the simplified alignment, being $M' \leq M$.

Algorithm 2 shows in line 20, how given a pair of consecutive columns with information score I_j and I_{j+1} below a threshold I_τ (line 20), the characters in $\lambda_{i,j}$ are concatenated to the beginning of $\lambda_{i,j+1}$. Such concatenation is repeated for each row in the alignment for the selected columns, then column j is added to the list of columns to be removed (*listRemove*). The matrix α is the resulting simplified alignment and contains the same columns as λ except for the columns in *listRemove*, i.e. the columns categorised as candidates for a horizontal merge (line 27). Note that matrix α will have elements that are a concatenation of characters, whereas λ only contains single characters. Those elements in α that are a concatenation of characters represent the sub-sequences that have been merged.

Fig. 4.3 shows an example about how the Simplify step works for a given alignment matrix. The events (represented as characters) in columns 1 to 2 and 6 to 10 are row-wise merged into a single position in the final simplified matrix of alignment α. The visual encoding of these row-wise merged events is further explained in section 4.5. Algorithm 2 finishes by creating the new node $n_{a \cup b}$, assigning n_a and n_b as its child nodes, λ as its alignment, and α as its data representation (Line 28). Note that the original alignment matrix λ is also kept so that it can be used to build the alignment of subsequent nodes.

4.3.2.4 Multilevel data representation

The proposed data representation allows one to explore the overview across clusters (i.e. *vertical level-of-detail*) and longitudinally (i.e. *horizontal level-of-detail*). Note that these names map to the vertical and horizontal axes of the user interface for the system SequenC (see section 4.5). However, the research contribution is related to the clustering and longitudinal aspects themselves regardless of the orientation of the axis in the interface. For example, if the interface was to be rotated to place sequential order on the vertical axis, the summarisation technique would still apply.

Vertical level-of-detail: The vertical level-of-detail is proportional to the number of clusters k in the overview. The larger the value of k, the finer the overview, where $1 \leq k \leq N$ and N is the number of input unique sequences. The change in the vertical level-of-detail according to k is thanks to the proposed cluster representation. For example, Fig. 4.2-B shows how, as the aggregate tree is cut at a higher level in the hierarchy (e.g. $k = 1$ or $k = 2$), clusters have a higher intra-cluster variation so the number of merged columns increase, resulting in a coarser overview. As the tree is cut at a lower level in the hierarchy (e.g. $k = 4$ or $k = 5$), clusters have a lower intra-cluster variation so the column-wise information score gets closer to 1, resulting in a finer overview. Ultimately, when one cluster contains a single sequence ($k = 6$), all columns in the alignment matrices have an information score of 1, showing an overview with the highest level-of-detail possible.

Horizontal level-of-detail: The horizontal level-of-detail of the overview depends on the threshold I_τ, where $0 \leq I_\tau \leq 1$. The larger the value of I_τ, the representation of clusters become coarser horizontally. Fig. 4.3 shows the representation for an example cluster with different values of I_τ, when $I_\tau = 0$ no events are merged showing full detail and as it moves towards $I_\tau = 1$ the number of merged events increase.

4.3.3 Finding optimal overviews

Users are able to explore all clustering combinations in the aggregate tree. The average silhouette width metric [47] measures the quality of a clustering to find the optimal number of clusters which reflect homogeneous and well-separated distinct groups. In this thesis, the metric is used to suggest a set of optimal number of clusters, which result in a set of overviews with an optimal vertical level-of-detail. For a given number of clusters k, the *average silhouette width* $\bar{z}(k)$ is defined as the mean $z(s)$ of all the elements in the dataset, where $z(s)$ is the *silhouette value* of the element s. In this case, s is each of the input unique sequences used to build the aggregate tree. As defined by Rousseeuw [81], $z(s)$ is given by:

$$z(s) = \frac{v(s) - u(s)}{\max(u(s), v(s))},$$

where $u(s)$ is the average distance between element s and the other elements in the same cluster, $v(s)$ is the average distance between s and the elements in the nearest cluster (neighboring cluster), and $-1 \leq z(s) \leq 1$. Kaufman and Rousseeuw [47] suggest that the most optimal k is that one for which $\bar{z}(k)$ is the largest (global maxima). In some cases, the most optimal k might still be too many or too few clusters for the user. To provide a balance between number of clusters and quality of clustering, this thesis proposes to obtain the peaks (local maxima) in the $\bar{z}(k)$ function as alternative optimal number of clusters. A set of optimal overviews are indicated by the global and local maxima in $\bar{z}(k)$ (see Fig. 4.5). These will indicate relative good partitioning of the sequences and therefore provide a good visual overview.

With increased number of sequences, the number of suggested alternative optimal overviews can become too large (e.g. > 100), making it challenging for users to choose amongst all these options. This number could be reduced by only selecting those k values that are a local maxima and are not greater than the most optimal k. For example in Fig. 4.5, only $k = 3$ would be selected as an alternative optimal overview and $k = 16$ would be excluded. Another solution could be to select only those number of clusters that are a local maxima and whose average silhouette width is greater than a threshold (e.g. $\bar{z}(k) > 0.2$). In other cases, local maximas can be too close to each other (e.g. $k = 8$ and $k = 10$), producing overviews that might not have very meaningful differences. Future work is needed to select meaningful yet distinct alternative optimal overviews amongst close local maximas, specially with large number of clusters.

4.4 Complexity Analysis

The analysis reported in this section was obtained in a laptop with the following characteristics:

- macOS Big Sur Version 11.4
- MacBook Pro (15-inch, 2017)
- Processor 2.9 GHz Quad-Core Intel Core i7
- Memory 16 GB
- Graphics Radeon Pro 560 4GB

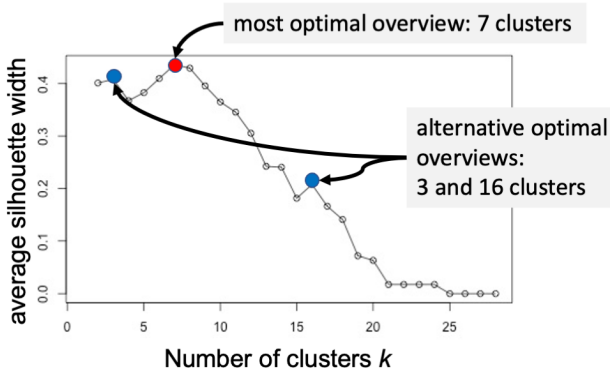
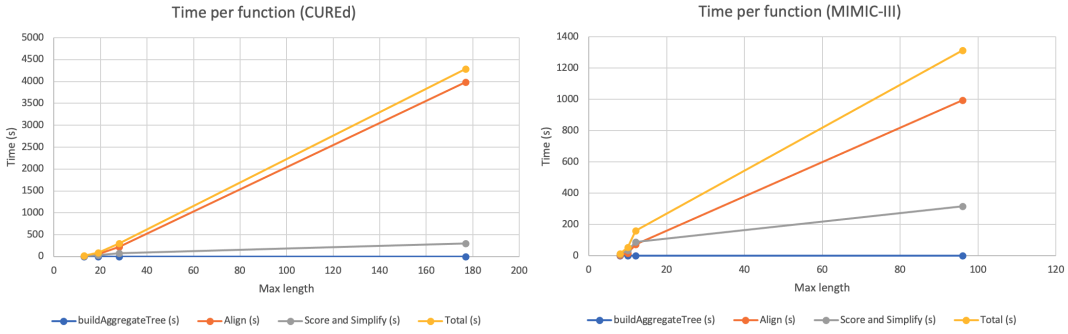


Figure 4.5: *Average silhouette width for different number of clusters. The optimal number of clusters is 7 (highlighted in red), as this is the one with the highest average silhouette width. Alternative optimal number of clusters are indicated by the peaks in the curve (highlighted in blue).*



(a) (b)

Figure 4.6: *Running time per function in Algorithm 1 (`buildAggregateTree`) and Algorithm 2 (`aggregate`), with respect to the maximum sequence length of the datasets shown in table 4.1. Figure (a) shows the times for the CUREd dataset, using the maximum sequence lengths 13, 19, 28, and 177. Figure (b) shows the times for the MIMIC-III dataset, using the maximum sequence lengths 8, 10, 12, and 96. The function `aggregate` includes the functions Align, Score and Simplify. The Align function is the most time consuming.*

The time complexity of Algorithm 1 (`buildAggregateTree`) is $O(Nnl^2)$, where N is the number of input unique sequences, l is the maximum sequence length in the dataset, and n is the average number of sequences per node. The function `aggregate` (Algorithm 2) is repeated $N-1$ times, from which the alignment step (MSA) is the most time consuming with a complexity of $O(nl^2)$. Note that l is the maximum sequence length in the dataset, as opposed to the average sequence length, because the length of a new alignment $n_{a\cup b}$ will be at least the maximum sequence length amongst the sequences in the nodes n_a and n_b. If a sequence with a high length is aggregated at an early iteration, this length will be carried to all subsequent alignments.

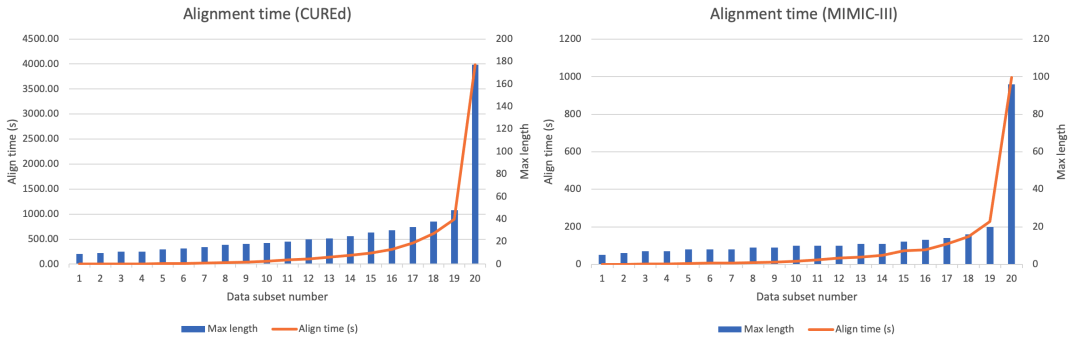
Fig. 4.6-(a) and Fig. 4.6-(b) show plots comparing the running time in seconds for the functions in both algorithms according to the maximum sequence length of four subsets of the CUREd and MIMIC-III data shown in Table 4.1. As observed, the maximum sequence length in the dataset highly impacts the time of the Align function. Note that the plots in Fig. 4.6 should show a quadratic curve given the term l^2 in the time complexity, however, they look quite linear due to the huge gap between the first three maximum lengths and the last one (e.g. 13, 19, 28, and 177 for CUREd).

Table 4.1 shows the time performance for subsets of 25%, 50%, 75%, and 100% of the data. Tables D.1 and D.2 in Appendix D show the time for subsets increasing every 5%. Unique sequences were ordered by length before selecting subsets, meaning that the 5% subset contains the shortest sequence. The additional subsets allow us to see how the maximum sequence length impacts the time performance, even when the number of unique sequences does not increase much between subsets - for example when comparing the alignment time between the 95% and 100% data subsets (see Fig. 4.7-(a) and Fig. 4.7-(b)).

Note that an increase in the number of event types does not directly impact the alignment time. The MSA algorithm only uses event types to obtain the *substitution_score*, by comparing whether the event is of the same type or not (see section 2.4.1.1). However, an increase in the number of event types might result in longer sequences, which as discussed, results in larger alignment times.

Dataset	Percentage	No. sequences		No. event types	Length of sequences		Execution time (s)			
		Individual	Unique		Average	Maximum	`build AggregateTree`	`aggregate`		Total
								Align	Score & Simplify	
CUREd	25%	20,124	240	11	9.96	13	0.0	9.3	10.1	19.4
	50%	20,901	481	11	13.07	19	0.1	59.1	30.9	90.1
	75%	21,520	722	11	16.24	28	0.2	225.5	74.5	300.2
	100%	**21,805**	**962**	**11**	**22.70**	**177**	**0.4**	**3984.7**	**301.0**	**4286.0**
MIMIC-III	25%	442	328	158	6.07	8	0.1	3.5	9.7	13.2
	50%	770	656	242	7.31	10	0.2	16.9	35.6	52.7
	75%	1,097	983	338	8.45	12	0.5	73.4	86.9	160.8
	100%	**1,425**	**1,311**	**448**	**10.67**	**96**	**1.5**	**995.4**	**317.5**	**1314.3**

Table 4.1: *Time performance analysis for algorithms* `buildAggregateTree` *and* `aggregate` *for different subsets of the MIMIC-III and CUREd datasets. The time for* `buildAggregateTree` *does not include the function* `aggregate`*, which is broken down according to the Align, Score and Simplify steps.*



(a) (b)

Figure 4.7: *Alignment time in seconds for 20 subsets of the CUREd (a) and MIMIC-III (b) datasets. The blue bars show the maximum sequence length of each subset. The data used for these plots can be found in Appendix D.*

4.4.1 Implementation details

The GUI of Sequen-C was implemented in Java, while the clustering and alignment steps were implemented in R.

As mentioned, a big factor in the performance is the maximum sequence length in the dataset (l). As observed in Table D.1 and Table D.2 in Appendix D, around 300 to 500 input unique sequences with an average sequence length of 7 to 10 events can be aligned relatively fast (under 10 seconds), as long as the length of certain sequences do not go too far from the average. For Sequen-C to have real time interaction, Algorithms 1 and 2 are precomputed, except for the Score and Simplify steps which are computed on the fly as the value of the number of clusters (k) or information score threshold (I_τ) change.

The R script precomputes the aggregate tree and the alignment of each node in the tree. The hierarchy of clusters and alignment matrices are saved to a file. For a selected number of clusters k, the alignment matrices of the k clusters are retrieved from the file by the Java program. Then, the Score and Simplify steps are computed on the fly over the retrieved alignment matrices according to the current value of I_τ. The times reported in Table 4.1 refer to the time required to compute Score and Simplify for all the nodes in the tree, however, in the implementation these steps are computed only for the selected values of k or I_τ. On average, it takes 0.11 seconds to recompute Score/Simplify and repaint the visualisation of a given number of clusters, this allows users to change the vertical and horizontal level-of-detail in real time.

4.5 Visualisation system: Sequen-C

Sequen-C was designed according to the analytic tasks presented in Chapter 3. The present chapter focuses on all tasks (**T1-T8**) except for **T6** and **T7**, which are addressed in Chapter 5. The system (see Fig. 4.8) is composed by three coordinated views: *multilevel overview*, *unique sequence view*, and *individual sequence view*; and a fourth view, the *attribute analysis view*. The next sections describe each of the views and the available user interaction, including the analysis of data attributes and filter options.

Figure 4.8: *Sequen-C visualisation system. In the multilevel overview (A), cluster C5 is selected and its unique sequences are shown in the unique sequences view (B), where unique sequence S9 is selected, showing its 411 individual sequences in the individual sequences view (C). The attribute analysis (D) shows how attributes of the selected data relate to the whole dataset. (E) highlights some of the available controls.*

The GUI/front-end and most of the back-end of the system Sequen-C were implemented in Java, except for the clustering and alignment algorithms which were implemented in R (see section 4.4.1). The Java libraries Swing and Graphics2D were used to build the GUI and visualisations. SQLite was used as the database management system to store the datasets used in the case studies. Appendix E provides further details of this implementation, including an Entity Relationship (ER) diagram of the database tables and main Java classes, and a diagram of how the front-end interacts with the back-end.

4.5.1 The multilevel overview: cluster view

The multilevel overview (see Fig. 4.8-A) shows a variable number of sequence clusters, where each cluster is visually encoded according to the data representation matrix α constructed using the steps Align-Score-Simplify. Users can interact with this view through two sliders to transform the horizontal and vertical level-of-detail of the overview.

4.5.1.1 Visual encoding

A cluster is visually encoded as shown in Fig. 4.3. Event types are represented as coloured boxes with a height proportional to the number of records and colour indicates the event type. Equal event types in consecutive rows are merged to reduce visual clutter. Sequences in a cluster are ordered by similarity, and gaps (–) in the alignment are encoded as spaces between events.

The final height of a cluster is proportional to the number of records it contains. However, sometimes clusters might contain too few records in proportion to the whole dataset and they would not be visible. In such cases the height is scaled up by a constant number of pixels

and the cluster is surrounded by a dotted line, allowing users to identify deviating pathways **(T1)**.

Fig. 4.3 shows how each of the elements in the representation matrix can contain either one or multiple event types; an element containing multiple event types corresponds to the row wise merged sub-sequences in the Simplify step. Sub-sequences contained in a single element are represented using a box divided by coloured bars, where each bar is coloured by event type and ordered as per the sub-sequence. This visual encoding allows one to derive the original sequences forming a cluster **(T2)**. However, with increasing number of events in the merged sub-sequence, to reduce visual clutter, bars can be ordered by event type to show proportion, or coloured in gray to show the number of merged records.

4.5.1.2 Transforming the level-of-detail

Fig. 4.8-E shows the two sliders used to transform the level-of-detail of the overview: the *cluster slider* and the *information score slider*. The cluster slider allows users to transform the vertical level-of-detail by changing the number of clusters k in the range $1 \leq k \leq N$, where N is the number of input unique sequences. Additionally, a combobox next to the cluster slider shows the current number of clusters and contains the list of alternative optimal number of clusters, obtained as per subsection 4.3.3, to guide users in finding a set of pathways that best summarise the data **(T1)**. Alternatively, users can break down a selected cluster into its two child sub-clusters, and so on, until a cluster with a single sequence is reached (see Fig. 4.1). The information score slider transforms the horizontal level-of-detail by changing the information score threshold I_τ in the range $0 \leq I_\tau \leq 1$.

4.5.2 Unique sequence view

This view shows individual sequences contained in selected clusters, grouped by unique sequence **(T4)**. Unique sequences are visually encoded as an ordered sequence of boxes arranged horizontally and coloured according to their event type (see Fig. 4.8-B), along with their unique sequence identifier and frequency. This view shows unique sequences without any simplification, allowing the inspection of the full sequences in the selected clusters **(T2,T4)**. Unique sequences in this view can be sorted by frequency or similarity, or aligned by a selected event.

4.5.3 Individual sequence view

This view shows the individual sequences (see Fig. 4.8-C) of the selections in the unique sequence view and the overview **(T4)**, along with their temporal information and raw data attributes. Following a Gantt chart approach, each individual sequence is visualised as a horizontal sequence of events, positioned along the horizontal axis according to their timestamp. A table of attributes is displayed next to the Gantt chart, where each column represents a data attribute at either individual sequence level or individual event level **(T5)**.

4.5.4 Attribute analysis view

The distribution of a data attribute can be analysed for a selected set of records **(T3)**, or compared amongst clusters and unique sequences **(T5)**. This view shows one stacked bar

chart per attribute in the dataset (see Fig. 4.8-D), where a chart contains one vertical bar per value, each bar is divided in sub-bars representing series, and series are identified by a unique colour. Series can be interactively hidden to focus on only one or compare a reduced number of series. Three types of charts are provided: **1) Selected data**: compares the selected data against the rest of the records in the dataset. For a given attribute, this type of bar chart shows one series coloured in red for the records contained in the selected clusters or unique sequences, and another series (in grey) for the rest of data. **2) Sequence**: it plots one series for each unique sequence shown in the unique sequence view. **3) Cluster**: it compares all clusters in the overview, and assigns one series per cluster.

4.5.5 Filters and selections

Records can be removed from the overview by applying filters based on data attributes, frequency, date range, event occurrence; including filters by day of the week, month, or year **(T3)**. A filter is specified by an attribute, operator, and value. For example, the filter *event* = *A* translates to "show only sequences that contain event A at least once".

Users can select sections of a cluster, such as events and sub-sequences, or sequences in the unique sequence view by drawing a square with the mouse. These selections are added to the unique sequence view and individual sequence view, and are plotted in the attribute analysis view **(T4)**.

4.6 Case studies

To demonstrate the effectiveness of the present technique to derive insights from the data in line with the analytic tasks, two case studies are presented using real-world datasets: 1) MIMIC-III [45] and 2) Antenatal Care Unit (ANC). Table 4.2 describes the characteristics of the sequences in these datasets.

Table 4.2: *Characteristics of MIMIC-III and ANC datasets.*

Dataset	Event types	Avg. / Max length	Individual sequences	Unique sequences
MIMIC-III	448	10.6 / 96	1,425	1,311
ANC	31	10.15/ 27	9,623	440

4.6.1 MIMIC-III

The MIMIC-III database [45] contains data related to 58,976 patient admissions to acute and critical care units at a tertiary hospital. The database comprises 26 tables containing demographic and timestamped data for all clinical events from admission to discharge (or death). In this case study, an individual sequence represents all the timestamped events for a single admission, obtained from the tables: admissions, transfers, and prescriptions.

This case study was developed in collaboration with a consultant cardiologist (i.e. the analyst). Given that MIMIC-III is a public access database and was not provided by a known

collaborator (like in the other case studies), the collaboration of a cardiologist was seek, to make sure that the data analysis had clinical relevance. The cardiologist suggested to explore patients with a diagnosis of Atrial Fibrillation, and the impact that medications and timely diagnosis could have in their length of stay. This case study was developed through online video calls (during the COVID-19 pandemic) and through email exchanges. The system Sequen-C and the visualisations produced with the MIMIC-III dataset were shown to the cardiologist, who then would make observations or request to obtain more details on certain aspects of the visualisation. Additionally, the PhD student would separately explore the dataset, and email any questions or insights to the cardiologist. Based on the cardiologist feedback, more visualisations would be produced and reviewed on the next video call. Such process was repeated iteratively.

A query was added to show patients with a primary or secondary diagnosis of Atrial Fibrillation (AF) (code 42731 in the DIAGNOSES_ICD table). The subset data contained 1,425 patient admissions, and 448 event types, from which 438 are types of prescriptions.

4.6.1.1 Overview of care unit and prescription patterns

The overview of the 1,425 individual sequences was explored by the analyst and us, selecting different number of clusters and breaking them down into more granular detail within interesting clusters, the final overview shows 33 sequence clusters. The commonest care units for patients with AF were the: *Cardiac Surgery Recovery Unit* (CSRU), *Coronary Care Unit* (CCU), *Medical Intensive Care Unit* (MICU), and *Surgical Intensive Care Unit* (SICU). By default, clusters were ordered by similarity. The first 18 clusters, comprising 55% of all the admissions, started attendance to the Emergency Department (ED) with subsequent transfer to an inpatient care unit. Clusters were ordered by frequency. Fig. 4.9-A shows that, in general, the selected clustering either groups patients sharing a specific drug but admitted to different care units (e.g. Furosemide and Potassium Chloride predominate in cluster C4), or patients admitted to a specific care unit that can be sub-divided into different treatments (e.g. admission to CSRU in cluster C1). Infrequent, less populous clusters (with less than 1% of frequency), represent more exceptional *outlier* scenarios with less intra-cluster variation and very similar set of drugs (Fig. 4.9-D).

4.6.1.2 Comparing attributes across clusters

Focusing on the main clusters (C1 to C9), for each care unit, there is a cluster of patients admitted directly to that care unit and a second cluster of patients passing through ED before being transferred to that unit (e.g. clusters C2 and C6). As observed in Clusters C1 and C5, this is different in the case of the CSRU unit, where most patients do not pass though ED first. Most of the patients in cluster C1 were treated with Metoprolol; whereas most of patients in cluster C5 were treated with Warfarin. To inspect the characteristics of these patients, clusters C1 and C5 were analysed in the attribute analysis view, Fig. 4.9-B shows that patients in cluster C5 tend to have longer lengths of stay (11 days in average) compared to cluster C1 (7 days in average). The analyst mentioned that this is likely to be associated with the requirement for careful dose *titration* with Warfarin. The analyst mentioned that "such observations are helpful in informing healthcare planning; outpatient dosing could justifiably be targeted at this cluster to reduce length of stay and free up valuable hospital

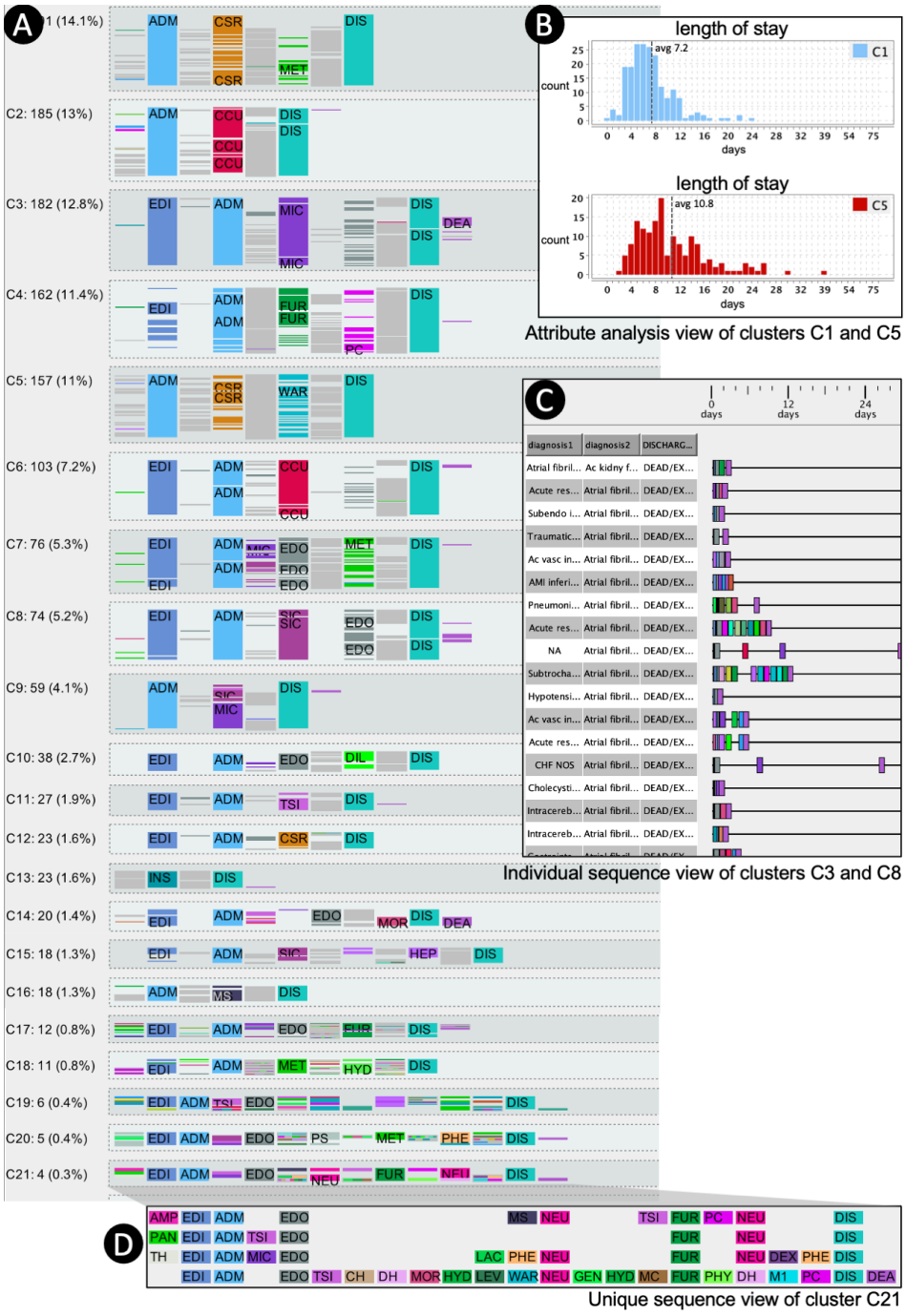


Figure 4.9: *Overview of 1,425 admissions of patients with a first or second diagnosis of Atrial Fibrillation, obtained from the MIMIC-III dataset.*

beds". Similarly, the analyst observed that there was a difference in length of stay between two clusters representing different rate control medication; the average length of stay was 9 days when Diltiazem was used (cluster C10), but just 7 days when Metoprolol was used (clusters C1 and C7).

4.6.1.3 Details on-demand for records of interest

The analyst was curious to explore the clusters showing a higher mortality (clusters C3 and C8) and their relation with a first or second diagnosis of AF. Clusters C3 and C8, which contain admissions to MICU and SICU respectively, were selected and added to the individual sequence view. The columns *diagnosis1*, *diagnosis2*, and *discharge_location* were selected to be visualised in the table next to the Gantt chart, then sequences were ordered by *diagnosis2* and *discharge_location*. This allows one to see that, for clusters C3 and C8, there is a significant higher number of deaths when AF is a second diagnosis compared to when it is a first diagnosis (Fig. 4.9-C). This might be probably because first non atrial fibrillation diagnoses might be more serious conditions.

4.6.2 Antenatal care unit (ANC)

Note that the Levenshtein distance was used in this case study, as opposed to the cosine q-grams distance used in the rest of the case studies presented in this thesis. This was the first case study to be developed, and at that point the Levenshtein distance was used. However, for the other case studies (MIMIC-III and CUREd), the choice of distance metric was changed to cosine q-grams as the Levenshtein distance can be highly influenced by the length of sequences or event repetitions (see section 2.3.1).

4.6.2.1 Background

This case study has been developed in collaboration with two analysts at Sheffield Teaching Hospitals in the United Kingdom. The studied dataset contains 73,279 events recorded during the visit of 9,623 patients to the Antenatal Care Unit (ANC) outpatient clinic over a period of 3 months.

Patients attending the ANC clinic are mostly pregnant women who require consultations, ultrasound scans, blood tests and other services. Events are recorded for every patient from her arrival to departure to the clinic. An event sequence represents a whole patient journey in a single visit. Events are recorded as members of the clinic staff use an in-house patient flow tracking software. This tracking system allows clinicians and nurses to select the next state in the visit of a patient given her current state, for example, if the patient needs to have a glucose-tolerance blood test after seeing the consultant. Patients are assigned to a pathway code depending on the purpose of their appointment; a ***pathway*** determines the possible states for a specific type of visit. Patients with the same pathway code will follow a similar journey. Pathways are embedded in the tracking software and provide the staff with a list of possible next steps for the patient according to their assigned workflow.

Pathways are specified in the tracking software using workflow diagrams - one workflow diagram per pathway. Fig. 4.10 shows an example of a workflow diagram, in this case for pathway code G. Individual patient journeys derived from the same pathway are likely to be very similar to each other but vary as the workflow branches. The workflows currently in

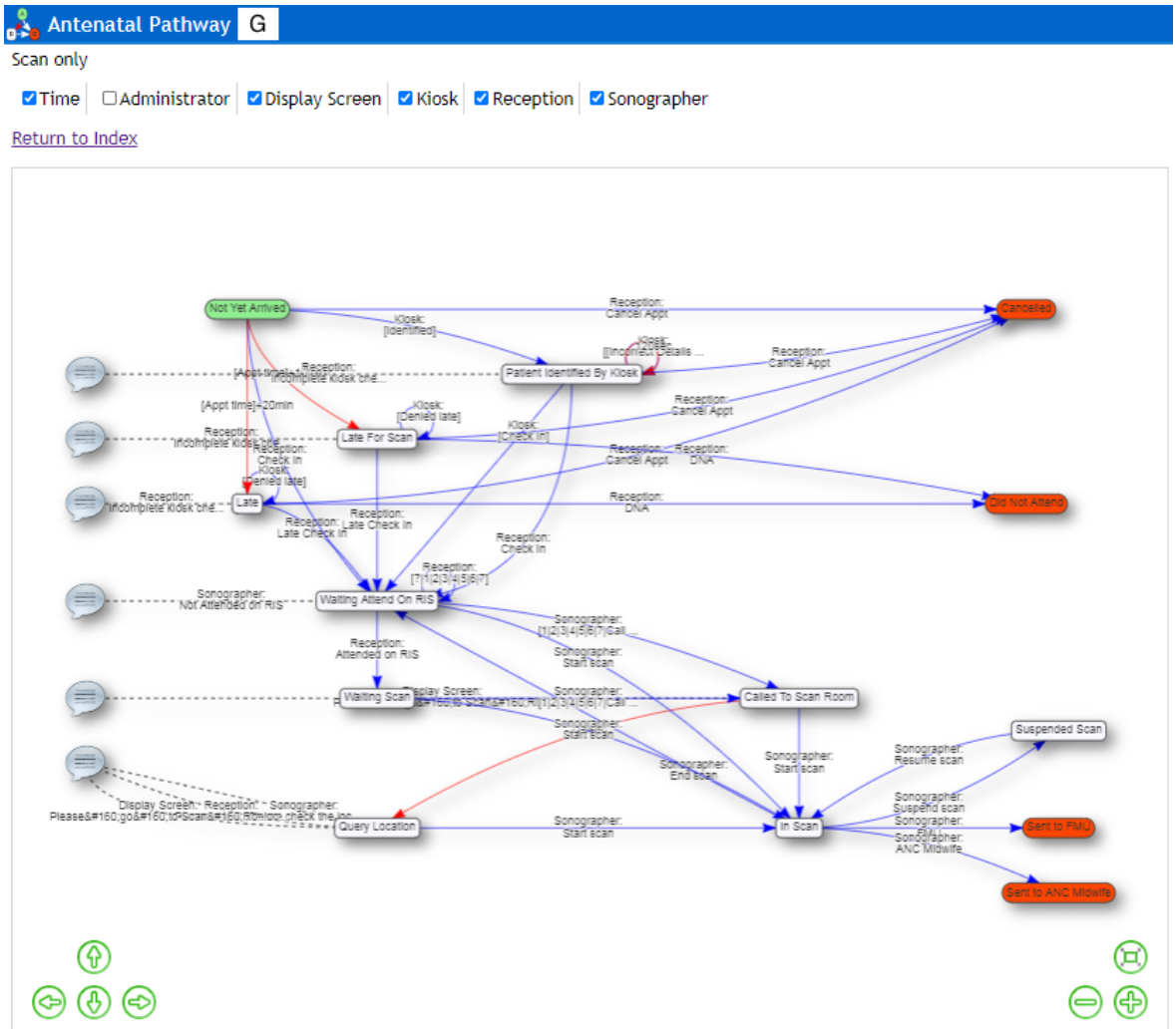


Figure 4.10: *Example of a Workflow diagram used in the in-house tracking software of the Antenatal Care Unit. The diagram corresponds to PathwayCode G.*

use have been designed by the Scientific Computing department through interviews with the clinical staff. Analysis of the current setup is intended to provide insight into and optimisation of the software configuration in view of real-world usage.

To provide a more realistic clustering, very infrequent sequences (considered to be input errors or outliers) were removed from the dataset using the cumulative frequency of the unique sequences in the dataset. The purpose was to find a relatively low number of unique sequences that can represent a high percentage of the data. A change rate of 0.001 was used to remove unique sequences occurring less than 0.1%, resulting in 74 out of 440 unique sequences being selected for the analysis (8,899 out of 9,623 individual sequences). As observed in Fig. 4.11, these 74 unique sequences represent 92.27% of the total data.

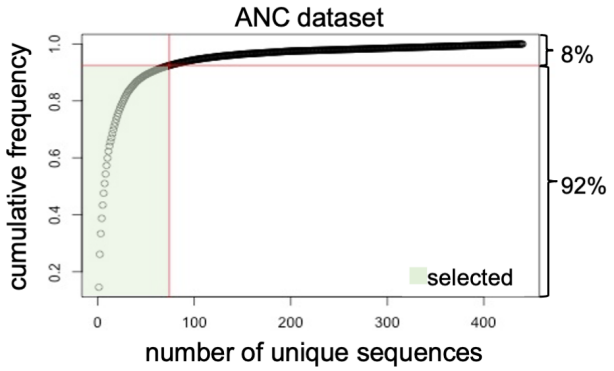


Figure 4.11: *Cumulative frequency function for the unique sequences in the Antenatal Care Unit dataset. A change rate cutoff point of 0.001 was used to remove outlier unique sequences, resulting in 92.27% of the data selected for analysis (highlighted in green).*

4.6.2.2 Analysis of pathways

The main goals of the present case study are:

- validate if the existing workflow design actually reflects the different journeys occurring in the clinic; and
- simplify, add, or remove pathway codes if necessary.

The present analytics methodology is suitable for these goals as each cluster could be interpreted as a separate pathway or workflow diagram.

An analysis session was conducted along with two analysts from the Scientific Computing department to explore the ANC dataset using the proposed visual analytics technique. After loading the data, the initial overview shows the sequences partitioned in two clusters, this clustering clearly divides the sequences into those related to the scan process and those without a scan (see Fig. 4.12-A).

The configuration of the tracking system should satisfy two requirements: 1) the pathways should be representative of the different treatment journeys and 2) there should be a balance in the number of pathways. Having too few pathways in a complex department (e.g. 1 or 2) would mean that the pathways have too many branches, and staff then have many unnecessary options available, risking errors and reducing usability. Having too many workflows (e.g. more than 30) could make maintenance of the tracking system complex, with overly specific or duplicated pathways. Clustering the real-world sequences indicates which branches are used in practice and thus suggests possible refactoring of the department's operation. The cluster slider in the overview allows analysts to explore any desired number of clusters. Moreover, optimal numbers of clusters are highlighted in the slider.

The analysts transform the overview by moving the slider to eight clusters, suggested as the next optimal number of clusters (see Fig. 4.12-B). The analysts commented that this clustering makes sense as it seems to separate the sequences into very different processes performed in the clinic; labels in Fig. 4.12-B (i.e. C1 to C8) indicate the process each cluster represents. In order to find out how much the existing pathway design complies with the suggested (optimal) clustering, the analysts navigate to the Attribute Analysis tab in the left

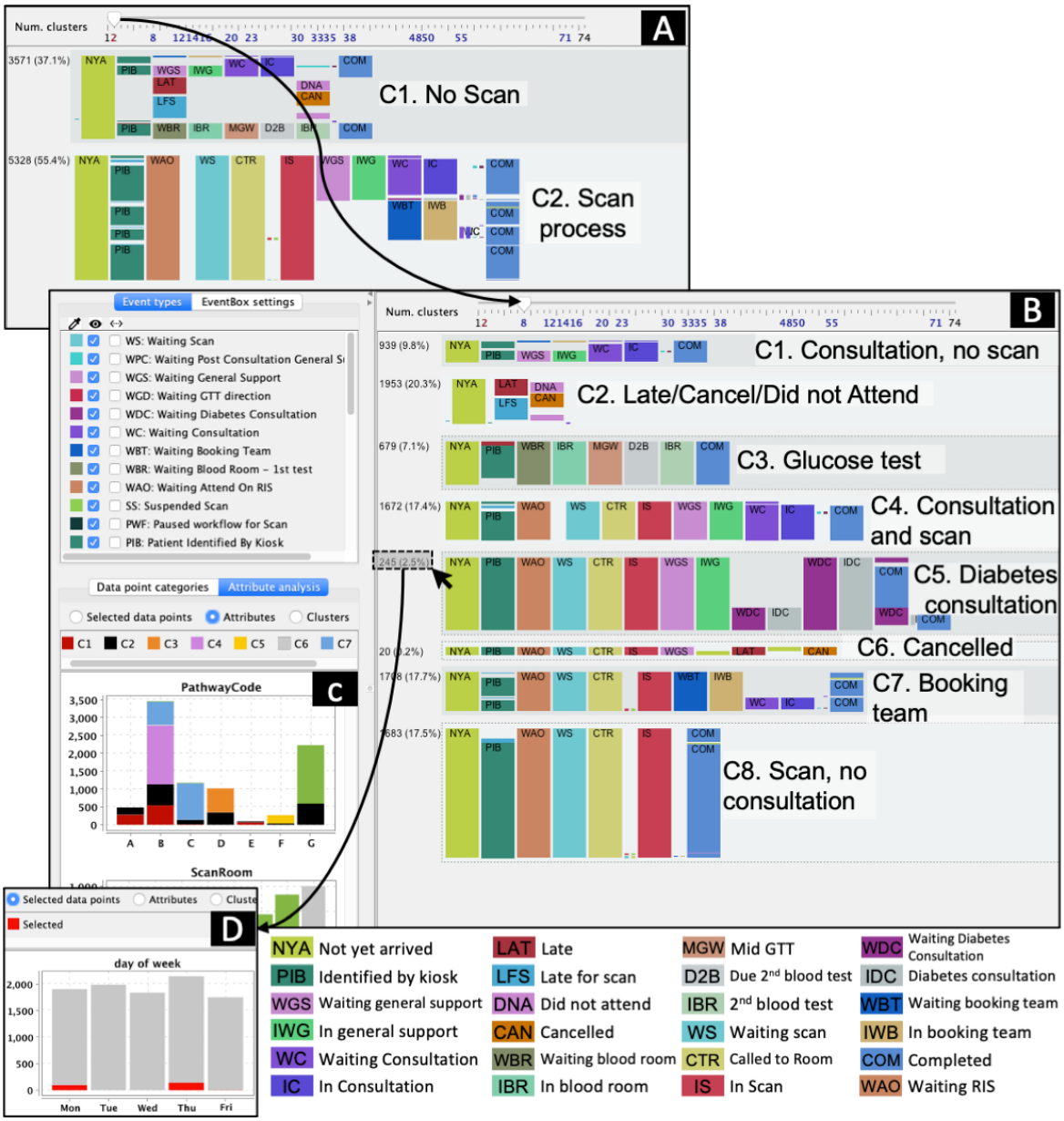


Figure 4.12: *Multilevel overview showing 2 and 8 clusters for the Antenatal Care Unit dataset. A) Overview showing two sequence clusters, indicated by C1 and C2. B) Overview showing eight clusters, indicated by C1 to C8, each cluster represents a process in the clinic. C) Distribution of clusters across the PathwayCode values. D) Bar chart for the day of the week indicating that the sequences in the selected cluster, C5, occur Mondays and Thursdays.*

panel where clusters are analysed in relation to each attribute. Fig. 4.12-C shows a stacked bar chart of the distribution of the eight clusters across the different values of the attribute *PathwayCode*. Figure 4.13 shows this bar chart in more detail.

There are 7 different values for the attribute PathwayCode identified with a letter from A to G, meaning there are 7 different workflow diagrams in the design of the tracking system of the ANC clinic. As mentioned, patients are assigned a PathwayCode when the visit is scheduled. According to the bar chart in Figure 4.13-A, cluster C2 is distributed across the 7 pathway codes. In other words cluster C2 contains sequences of all pathway codes. This can be explained as the sequences in this cluster represent visits that were scheduled but never happened. For this reason the analyst discards cluster C2 as a potential workflow.

The analyst also discards cluster C6 as this only contains 0.2% of the sequences and all the sequences in this cluster end with a cancel event. The analyst considers discarding cluster C5 as it contains only 2.5% of the sequences, however, this cluster is the only one containing sequences with Diabetes consultations (some of the sequences contain more than one consultation). Moreover, when analysing the bar chart for the day of the week attribute, the analyst notices the sequences in this cluster occur only on Mondays and Thursdays (see Fig. 4.12-D); this explains the lower frequency of the cluster.

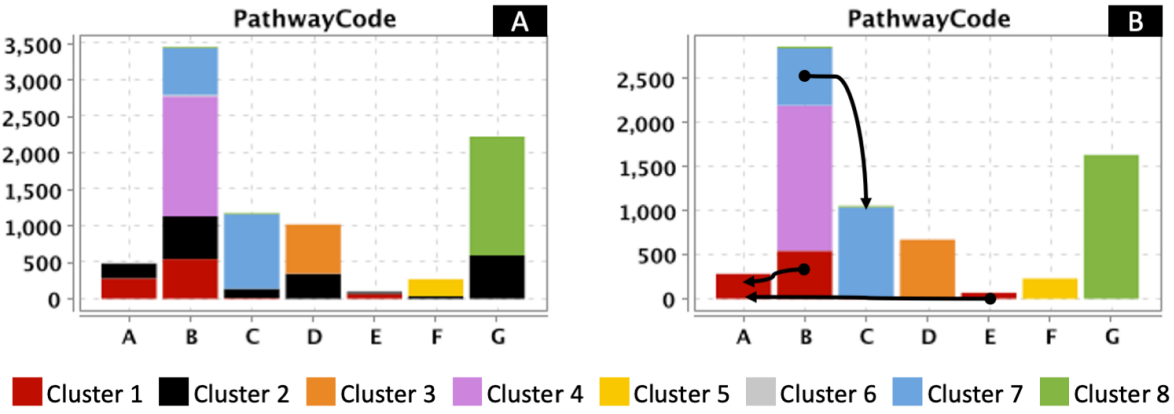


Figure 4.13: *A) Stacked bar chart showing the distribution of clusters across PathwayCode values. B) Updated chart after the series for Cluster 2 and Cluster 6 are removed. Pathway D, F, and G coincide with the suggested clustering; the arrows indicate that sections of Pathway B could be re distributed to Pathway C and Pathway A, and that Pathway E be merged with Pathway A.*

Clusters C2 and C6 are removed from the charts shown in the Attribute Analysis tab. As observed in the bar chart in Figure 4.13-B: pathway G is present only in Cluster C8 and at the same time Cluster C8 only contains pathway G (see Fig. 4.10). The same applies for pathway F and cluster C5, and pathway D and cluster C3. This indicates compliance of the suggested clustering with the pathways D, F, and G.

Pathway B is present in clusters C1, C4, and C7 and it is the Pathway code with the highest number of records. The clustering suggests that a section of Pathway B (Cluster C1) could be moved to Pathway A and another section (Cluster C7) to Pathway C. Moreover, Pathway E is used very infrequently and it only appears in patients with the sequence $NotYetArrived \rightarrow PatientIdentifiedByKiosk \rightarrow InConsultation \rightarrow Complete$ (see Fig. 4.14). This sequence could be integrated in Pathway A and Pathway E could be

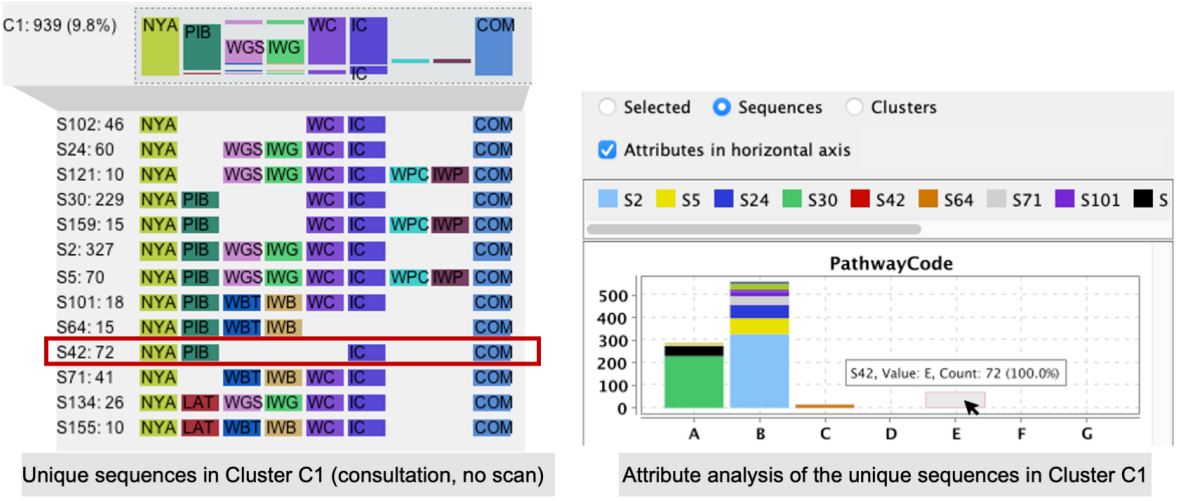


Figure 4.14: *Cluster C1 contains patients that had a consultation but no scan (top left). Unique sequences in cluster C1 (bottom left), the text next to each unique sequence represents its id and frequency, for example the first unique sequence "S102: 46" has the id=102 and frequency=46. The attribute PathwayCode is analysed (right). Each series colour represent the id of a unique sequence in Cluster C1. It is observed that the only sequence with PathwayCode E is the unique sequence S42 (highlighted in red).*

deleted, then patients that would normally be assigned to Pathway E would be assigned to Pathway A.

In summary, Pathways D, F, and G can remain as is. The suggested changes are: 1) to redesign Pathway A, B, and C. Move sections from Pathway B to Pathway A and C. 2) remove Pathway E. In practice, patients that would be normally assigned to Pathway E now would be assigned to Pathway A instead. This means the number of pathway codes is reduced from seven to six. According to this case study, the current pathway design mostly reflects the actual operation of the clinic. However, the suggested changes could be implemented to improve the usability and maintainability of the tracking system. It has been demonstrated how the visual analytics technique allows one to explore a process at different levels of detail and number of clusters, while also characterising the clusters.

4.6.3 Domain expert feedback

The domain experts (E1, E2, E3) that participated in the case studies presented in this thesis, confirmed the plausibility of the findings or considered that they required further investigation. We asked them to provide feedback about the usefulness of Sequen-C and the vertical and horizontal level-of-detail controls. **E1** said that the clustering suggested by the system was useful *"to discover how patient journeys differ beyond what we would expect"*. **E3** stated that this type of analysis *"can help to better understand clinical workflow data to improve services"*. **E2** particularly liked the functionality of exploring the attributes for a selected cluster. Experts found that the vertical level-of-detail control allows one to *"rapidly and intuitively manipulate the granularity [of the visualization]"* (**E3**) and that increasing the number of clusters is useful when looking for outliers **(E1)**.

E1 indicated that *"often the high frequency events [in the cluster view] are the most important"* as these pathways would usually be the ones targeted for interventions to improve outcomes, however sometimes the interest could be in rare scenarios *"that may have led to an adverse event"*. **E1** found useful having the flexibility of adjusting the horizontal level-of-detail to focus on both scenarios. Expert **E2** said that the horizontal slider was useful to *"cluster noise"*. Experts mentioned that it was sometimes confusing to know which subsequences had *collapsed* when the horizontal slider was changed and that they would prefer changing I_τ in smaller steps. Experts **E1** and **E3** mentioned that they would like to continue using the system to link additional data to expand the pathways and explore longer term clinical outcomes data.

4.7 Summary

This chapter has presented a technique to create and explore a multilevel overview of event sequences through hierarchical aggregation, where users can interactively transform the overview from coarse to fine *vertical* or *horizontal level-of-detail*. The overview presents by default k sequence clusters, where the optimal number of clusters is selected using the Average Silhouette Width (ASW) [81] and a data representation of each cluster is produced through the steps Align-Score-Simplify. Users are able to visualise any number of clusters k (vertical level-of-detail), and also transform the level of summarisation applied to clusters through the information score threshold I_τ (horizontal level-of-detail). To facilitate this exploration, the technique obtains a set of optimal number of clusters (k) that represent good overviews.

The visual analytics framework, Sequen-C, implements the technique proposed in this chapter, and allows details-on-demand exploration and the inspection of data attributes at cluster, unique sequence, or individual record level. The purpose of this technique is to provide a flexible overview of temporal event sequences whose overall level-of-detail can be easily transformed and that offers more than one optimal overview. Moreover, the technique aims to allow users to explore different sequence clusterings whilst providing an interpretable cluster representation, and allowing comparisons within and amongst clusters.

Two case studies with two real-world datasets, MIMIC-III and ANC, were presented in this chapter. The case study of the MIMIC-III dataset [45] was made in collaboration with a cardiologist, in which a subset of patients with a primary or secondary diagnosis of Atrial Fibrillation was studied. Sequen-C clustered patients according to their care unit and prescription history, and made it possible to identify clusters with overall higher length of stay. The ANC case study used real-world data from an Antenatal clinic and was made in collaboration with two analysts at Sheffield Teaching Hospitals. The analysis of clusters in the dataset resulted in a proposed re design of the existing workflows in the clinic. The domain experts that collaborated in the case studies found it very interesting to be able to explore sequence clusterings and instantly inspect further details of interesting patterns.

4.8 Limitations

Although the case studies presented in this thesis include a relatively high number of event types (MIMIC-III, 448 event types) and a high number of individual sequences (CUREd, 21,805 individual sequences), the visualisation presents scalability limitations. Events are

color-coded, making it difficult to distinguish more than a certain number of event types via color (e.g. 12), and navigating long lists of event types (e.g. > 20) might be complicated. In the latter case, a hierarchy of event types could facilitate this interaction.

An advantage of the cluster representation is that the event types in the simplified sub-sequences are explicitly encoded, which allows to understand variability and in some cases derive the original sequences. However, the Align-Score-Simplify approach presents three main limitations:

- Future work is needed to determine the optimal value of the information score I_τ.
- With increased number of event types, the representation of merged sub-sequences suffer from visual clutter. Future work is needed to provide alternative designs that better summarise simplified sub-sequences and improve the interpretation of information loss.
- The alignment step of the approach is too time consuming, it is affected by the choice of gap and substitution costs, and it would benefit from an event type categorization (e.g. care units, prescriptions), so that events can be aligned based on their meaning rather than the name of the event type. Alternative alignment methods could be proposed (e.g. based on the longest common sub-sequence). Moreover, an increased number of event types or longer event sequences introduce more variability and affect the quality of the alignment result.

A flexible overview that aims to uncover hidden insights has been proposed. However, the current technique still depends, at some level, on the nature of the dataset, and the knowledge and hypotheses of domain experts. Future work is needed to validate the current technique in other domains besides the clinical one. Lines of future research are to consider in the clustering aspects such as event type importance, event ordering (with strategies to avoid noise) and other data attributes, and provide a visual cluster representation that encodes such attributes.

The next chapter extends the proposed data overview to visually encode multivariate attributes via a novel visualisation called EventBox.

Chapter 5

EventBox: analyzing temporal and multivariate attributes

Current overviews of event data focus on the visual encoding of sequential patterns but present limitations when representing temporal (i.e. duration and time of occurrence) and multivariate attributes, as attributes are either not encoded in the overview or if present, they are oversimplified (e.g. using average values). This chapter proposes a technique to build an overview of multivariate, temporal, and sequential patterns - where multivariate and sequential patterns are encoded at the same time in a single overview. First, an overview of sequential patterns is obtained using the technique presented in Chapter 4, and second, the attributes in selected event types of the overview are aggregated via EventBox. The overview allows the comparison of trends in temporal and multivariate attributes within and across sequential patterns, while also allowing the identification of anomalous situations such as uncommon sequences and data points with outlier attribute values. The visual analytics system Sequen-C presented in Chapter 4 is extended to support EventBox. The technique has been applied to a real-world dataset obtained from a Rheumatology outpatient clinic, from which a set of findings have been derived. Moreover, Chapter 6 presents an extended case study where EventBox is demonstrated using a real-world dataset containing calls made to emergency services

This chapter is based on the publication [61]:

Magallanes, J., van Gemeren, L., Wood, S. and Villa-Uriol, M.C., 2019, October. Analyzing Time Attributes in Temporal Event Sequences. In 2019 IEEE Visualization Conference (VIS) (pp. 1-5). IEEE.

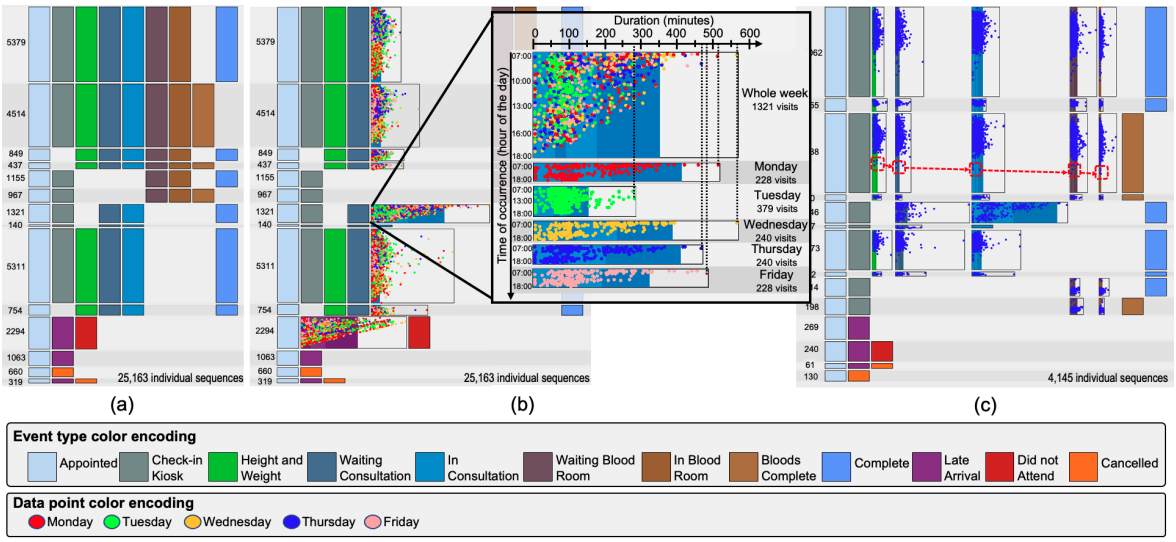


Figure 5.1: *Three screenshots of the system Sequen-C showing different configurations of the overview using the Rheumatology dataset. a) Overview of sequential patterns which shows events as point events. b) The events "In Consultation" and "Late Arrival" are encoded using EventBox. c) A filter is added to only show sequences happening on a Thursday.*

5.1 Introduction

Temporal event data (i.e. event logs) is routinely recorded in a variety of domains such as electronic health records, daily living activities, and web clickstream records. The analysis of event data can provide valuable insights into processes and the behaviour of individuals.

Existing techniques to create a visual overview of event sequences focus on the visual encoding of common pathways or sequential patterns - that is, patterns related to the order of the events in a sequence, but present limitations when representing other temporal and multivariate data attributes. The attributes of an event include its duration and time of occurrence, and other attributes that describe the data (e.g. age, gender, medical condition). These attributes are usually not included in the overview and can only be accessed through secondary views, or if included in the overview, they tend to be oversimplified (e.g. by using average values). For example, in the case of web clickstream analysis, current overviews allow the obtaining of patterns of the form "*the most common sequential pattern is: log in* $\rightarrow$ *purchase product X* $\rightarrow$ *log off*". However, by adding other multivariate data attributes to the overview, more complex patterns could be obtained, such as "*the most common sequential pattern log in* $\rightarrow$ *purchase product X* $\rightarrow$ *log off is performed mostly by young people in the morning*".

This chapter proposes a technique to create an overview of multivariate, temporal, and sequential patterns in event sequences. Firstly, a set of representative sequential patterns are obtained via hierarchical clustering and multiple sequence alignment as shown in Chapter 4. Secondly, the temporal and multivariate attributes of the records in each sequential pattern are aggregated at event level using the *EventBox* visualisation. EventBox is a novel visual encoding that allows users to explore trends and outliers with respect to the duration, time

of occurrence, and categorical attributes of a set of event occurrences of the same type.

The system Sequen-C is extended to support new analytic tasks related to EventBox. The technique is demonstrated using a real-world dataset from a Rheumatology outpatient clinic. The technique proved useful to study trends in the data attributes within and across sequential patterns, as well as identifying anomalous situations such as infrequent sequential patterns or data points with outlier attribute values.

The contributions of the present work are:

- *EventBox*: a novel interactive visual encoding that aggregates the duration, time of occurrence, and categorical attributes of a set of events of the same type.
- *Multivariate overview*: a technique to integrate multivariate, temporal, and sequential patterns into a single overview of event sequences.
- *A case study using a real-world dataset*: the case study demonstrates how EventBox allows users to obtain patterns and outliers with respect to multiple data attributes.

5.2 Related work

This chapter proposes a technique to integrate multivariate and temporal attributes into the overview proposed in Chapter 4. The next subsections present a review of existing visualisation techniques for temporal and multivariate attributes in event data.

5.2.1 Overview of temporal attributes

Existing visual analytics methods mainly focus on visualizing the sequential order of the events [65; 76; 101]. Currently, the visual encoding of time attributes is limited. Time of occurrence is always implicit in the sequential ordering of the sequences. However, no explicit time attributes (e.g. 3pm, Monday, May, 2019) are fully visually encoded in the overview.

Generally, a separate secondary view is required to review the time of occurrence for a selected record. ActiviTree [95] visualises the distribution of sequences across the time of the day, using a secondary view. In LifeLines2 [99], the distribution of the frequency of selected records through time is visualised using a histogram (i.e. the number of occasions a particular event happens on a specific date). Events within a time range before and after an alignment point can also be analysed. However, this method does not aggregate sequences, and frequency distribution is only shown for the selected event.

TimeSpan [59] uses stacked bar charts to indicate the duration of events related to a stroke treatment process, a line chart is used to study trends in duration through monthly intervals. However, they assume that sequences do not vary in the ordering of events, meaning sequential patterns are not included.

In methods where the duration of events is visually encoded, the width of an event is scaled proportionally to the average duration. This approach ignores the distribution of the duration and the presence of outliers. Duration outliers can be defined as observations with a duration which appears to be inconsistent with the remainder of the data [9]. Previous literature [95] indicates the importance of identifying infrequent sequences as outliers, but no emphasis has been made to pinpoint duration outliers.

Some techniques offer the possibility of filtering events by their duration. Eventflow [65] allows querying event sequences after specifying a time window (e.g. displaying only events with a duration above thirty minutes), while Eventpad [18] provides a histogram to inspect event attributes separately. However, this information is not encoded in the sequential patterns overview.

5.2.2 Overview of multivariate attributes

Some of the existing visualisation techniques already integrate multivariate attributes in the overview of event sequences, either representing attributes using average values or by grouping events or sequences according to their attribute value.

Outflow [101] and Frequence [76] propose a Sankey-like visualisation technique where edges are coloured according to average output. However, these techniques limit the visualisation to average values and do not show the distribution of the output values. Di Bartolomeo et al. [24] presents an overview of event sequences using a directed acyclic network where nodes are ordered and coloured by an attribute category. This visualisation shows how the value of an attribute changes from event to event, however, it is limited to a single attribute at the time and does not scale well as the number of event types increase. EventPad [18] separates sequential patterns according to the categories of an attribute, while Malik et al. [63] present a technique for cohort comparison where the overview shows an statistical analysis of a given attribute for each sequential pattern. However, the design of these techniques does not allow one to compare the distribution of multiple attributes across sequential patterns.

Liu et al. [57] and Borland et al. [12] visualise multivariate attributes in linked views using treemaps, bar charts, and other plots. These views focus on the distribution of the attributes but are not mapped to any sequential patterns.

A recent review on event sequence data visualisation by Guo et. al [41] mentions that one of the remaining challenges is to provide a visualisation design that is able to show categorical event types and multivariate attributes at the same time, which is exactly one of the challenges this thesis aims to address.

5.3 An overview of multivariate, temporal, and sequential patterns

This thesis proposes a technique to build an overview of multivariate, temporal, and sequential patterns in event sequences, where the distribution of multiple attributes can be compared across sequential patterns. The overview is built in two steps: firstly, a set of sequential patterns are obtained, secondly, the temporal and multivariate attributes of the records in each sequential pattern are aggregated at event level using an EventBox.

Chapter 4 presented a technique to build a multilevel overview of event sequences, where the overview presents k sequence clusters and each cluster is represented by a sequential pattern. A set of k sequence clusters is obtained through agglomerative hierarchical clustering [1] and the q-grams cosine distance [92], the default number of clusters k is determined using the overall average silhouette width [47], and then each sequence cluster is represented in the overview using the steps Align-Score-Simplify. This work is extended by interactively showing an EventBox in selected event types of the overview. Fig. 5.4 shows the overview

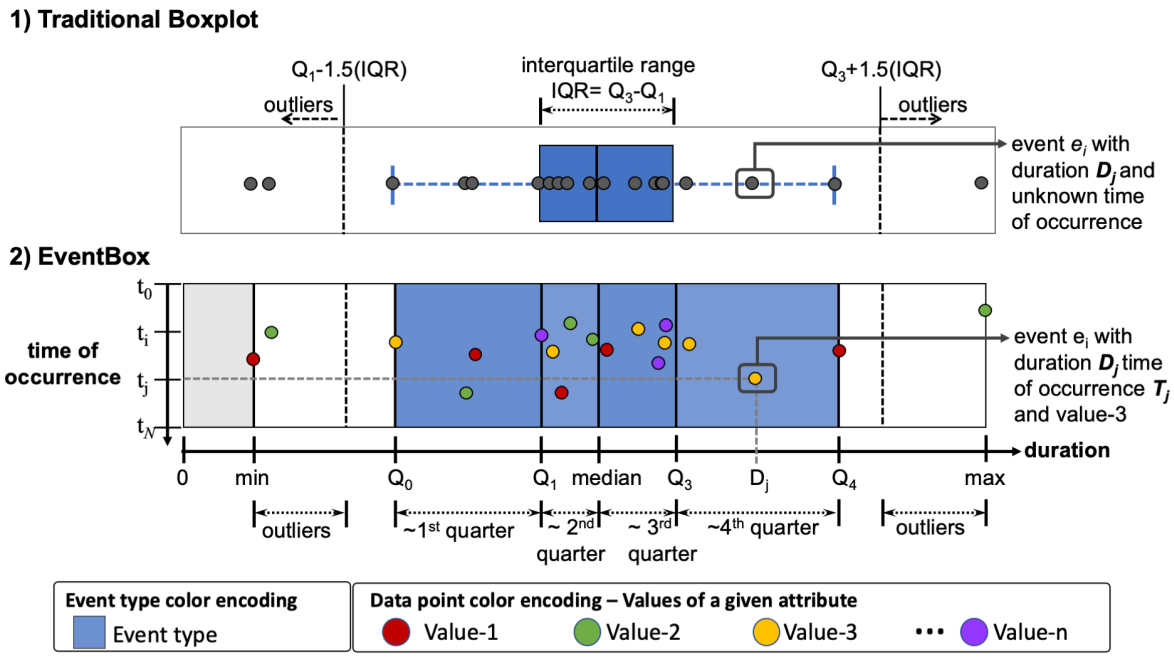


Figure 5.2: *Visual encoding of EventBox compared with the boxplot visualisation. (1) Traditional boxplot visualisation. (2) Visual encoding of EventBox which aggregates a set of events of the same type. Data points represent individual event occurrences. Quantile lines split the box into quarters representing duration ranges, where Q_0 is the minimum duration and Q_4 the maximum. Data points are located in the horizontal axis according to duration and in the vertical axis according to time of occurrence.*

for an example set of sequences, where the event type *WC* is selected to show an EventBox. This makes it possible to extract trends not only with respect to the most common pathways but also with respect to temporal and multivariate attributes.

An *EventBox* is a novel visual encoding that aggregates the temporal and multivariate attributes of a set of events of the same type. Given the definition of an event e, an EventBox $E = \{e_1, e_2, ..., e_n\}$ aggregates a set of events of the same type τ. Figure 5.2 shows the proposed visual encoding, where each data point represents an individual event occurrence e.

5.3.1 EventBox: temporal and multivariate attributes at event level

The EventBox visualisation is inspired in boxplots [91] and scatter plots. This combination allows the better exploration of the distribution of the data points and identify outliers with respect to multiple variables. The visual encoding of an EventBox is broken down using the definition of marks and channels [68]. An EventBox is composed by three marks: container area, quantile lines, and data points whose channels are used to represent temporal and categorical attributes.

- **Container area:** the *width* is proportional to the maximum duration, the *length* is proportional to the frequency, and the *colour hue* of the area represents the event type τ.

- **Quantile lines:** the *horizontal position* of each line mark represents the minimum, 25th percentile, median, 75th percentile, and maximum duration amongst all events in E. Quantile lines split the container area into sub-boxes called *quarters*, each sub-box containing about 25% of the data.
- **Data points:** a data point represents a single event occurrence. The *horizontal position* of the point represents its duration, and the *vertical position* its time of occurrence. The *colour hue* channel is used for categorical attributes from the event itself or the sequence that the event belongs to.

The channels that encode quantitative values are used for the time of occurrence and duration. However, the EventBox visualisation is not restricted to encoding exclusively time of occurrence and duration, these attributes are exchangeable for other quantitative attributes of interest. For example, the width, quantile lines, and horizontal position of data points, could be used to represent any other value besides duration - such as the results of medical lab tests or the age distribution of patients.

5.3.1.1 Container area

This area is the background that delimits and contains the quantile lines and data points. The frequency of an EventBox is the number of events being aggregated (i.e. $|E|$), this frequency is encoded using the **height** of the container area. The maximum duration amongst all event occurrences is encoded using the **width**. As mentioned, an EventBox aggregates only events of the same type τ; the **colour hue** of the area represents this event type.

5.3.1.2 Quantile lines

The distribution of the duration of all events in an EventBox E is represented on the horizontal axis of the container area using a boxplot-like visualisation.

A boxplot visualises a five-number summary of the data: minimum, lower quartile (Q_1), median, upper quartile (Q_3), and maximum; where Q_1 and Q_3 are the 25^{th} and 75^{th} percentile respectively. These five numbers (i.e. quantiles) divide the data into four parts, each part consisting of about a quarter of the total data [22]. A traditional boxplot visualisation is composed by a box and two whiskers, which are built by drawing a vertical line at each of these five quantiles. For simplicity, the boxplot is explained as if it was drawn on the horizontal axis. As observed in Fig.5.2-1 the box goes from the lower quartile to the upper quartile. Therefore, the size of the box is the difference between the upper and lower quartile. This difference is called the interquartile range (IQR) defined as $IQR = Q_3 - Q_1$. A vertical line is drawn in the middle of the box to represent the median. The lower whisker extends from the minimum to the lower quartile. The upper whisker extends from the upper quartile to the maximum. Outliers are drawn as points beyond the whiskers. Using Tukey's definition [91], outliers are identified as data points outside the range:

$$[Q_1 - k(IQR), Q_3 + k(IQR)],$$

where $k = 1.5$ is usually preferred [31]. *Lower outliers* can be defined as observations smaller than $Q_1 - k(IQR)$ and *upper outliers* as observations larger than $Q_3 + k(IQR)$. In the presence of lower outliers, the extreme of the lower whisker (minimum or Q_0) is given

by the smallest observation within the range $[Q_1 - k(IQR), Q_1)$; similarly, in the presence of upper outliers, the extreme of the upper whisker (maximum or Q_4) is given by the largest observation within the range $(Q_3, Q_3 + k(IQR)]$ [22].

The five quantiles used in a traditional boxplot are used in this work to divide the container area of the EventBox and indicate the distribution of the duration. Given the duration value of each event occurrence e: the minimum, 25^{th} percentile, median, 50^{th} percentile, and maximum duration are obtained. These quantiles are encoded using line marks and their **horizontal position** channel (see Fig.5.2-2). The scale of the horizontal axis goes from left to right, a gray area is drawn to indicate the gap between zero and the minimum value. Quarters (without outliers) are represented as sub boxes, the **colour saturation** of the container area is exchanged between quarters to visually separate them.

5.3.1.3 Data points

A point mark is used to represent an individual event occurrence e . The **horizontal position** of a data point represents its duration; and the **vertical position** its time of occurrence. The scale of the vertical axis is determined by the range $[T_0, T_N]$, where T_0 is the minimum and T_N the maximum time of occurrence being visualised; this scale is proportional to the available length of the container area. The scale $[T_0, T_N]$ can be adjusted to different time formats: as either hours of the day, days of the week, months, or years.

Categorical attributes from the event or sequence are encoded using the **colour hue** of the point. There can be event or sequence attributes - the event attributes provide context about the event occurrence, whereas a sequence attribute applies to all the events in the sequence (e.g. patient name).

Multiple aspects of the time of occurrence can be encoded using two channels simultaneously (vertical position and colour hue). For example, the vertical axis can represent the hour of the day, with a range starting from $T_0 = 8am$ to $T_N = 5pm$ at 1-hour intervals; whilst the colour hue could represent the day of the week (e.g. Monday, Tuesday).

5.3.2 Levels of detail of EventBox

In terms of duration, events are typically categorised into either point or interval events [67]. The proposed visualisation can be customised to show, hide, or modify the detail of the marks of an EventBox. This allows to produce the following levels of detail (see Fig. 5.3):

- *Point event*: the EventBox is fully *collapsed* and it is represented using a square area with arbitrary width - maintaining the length proportional to the frequency and colour hue representing the event type.
- *Interval event without duration outliers*: outlier data points (if any) are hidden.
- *Interval event with duration outliers*: quarters and data points including outliers are shown.
- *Interval event, detailed quarters*: data points inside quarters are shown.
- *Interval event, no detailed quarters*: data points inside quarters are hidden.
- *Interval event not coloured by contextual categorical attribute*: Data points are coloured using transparency, so that the volume of the points is observed.

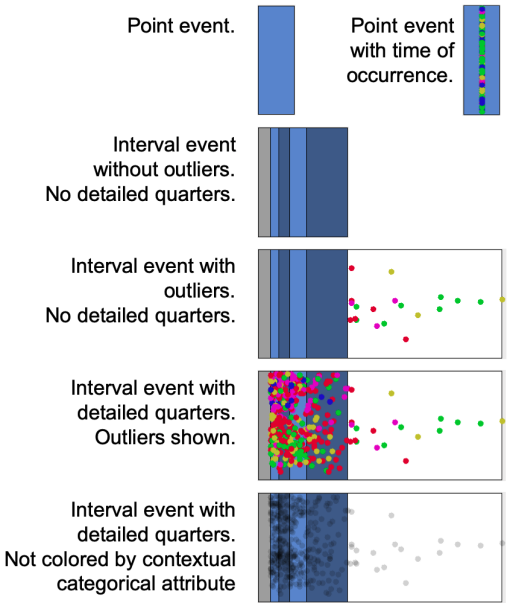


Figure 5.3: *Example of an EventBox at five different levels of detail. Hiding or showing quartile and outlier data points, and changing the colour of the data points, result in different levels of detail.*

- *Point event with time of occurrence*: regardless of point events having null duration, they do have a start time (time of occurrence). The EventBox is collapsed and data points are positioned over the vertical axis as per their time of occurrence.

These levels of detail help in reducing the visual clutter in the overview. Transforming EventBoxes into point or interval allow users to focus the analysis on selected events. Sometimes the user might be interested in the occurrence of a type of event without being interested in its attributes - a point event will provide context without having to remove it from display. When the EventBox is integrated in the overview of sequential patterns, it allows users to study and compare the distribution of attributes within and across sequential patterns.

5.4 The system

The system Sequen-C, presented in the Chapter 4, is extended to include new features that support EventBox according to the analytic tasks defined in Chapter 3. The present chapter focuses on all tasks (**T1-T8**) except for **T2**, which is addressed in Chapter 4. Fig. 5.4 shows the current system, composed by three coordinated views: overview, unique sequence view, and individual sequence view - and a panel containing controls that allow users to transform these views.

5.4.1 The overview: multivariate, temporal, and sequential patterns

Multivariate, temporal, and sequential patterns are integrated in a single overview, according to the technique presented in section 5.3. The system shows k sequence clusters ordered by

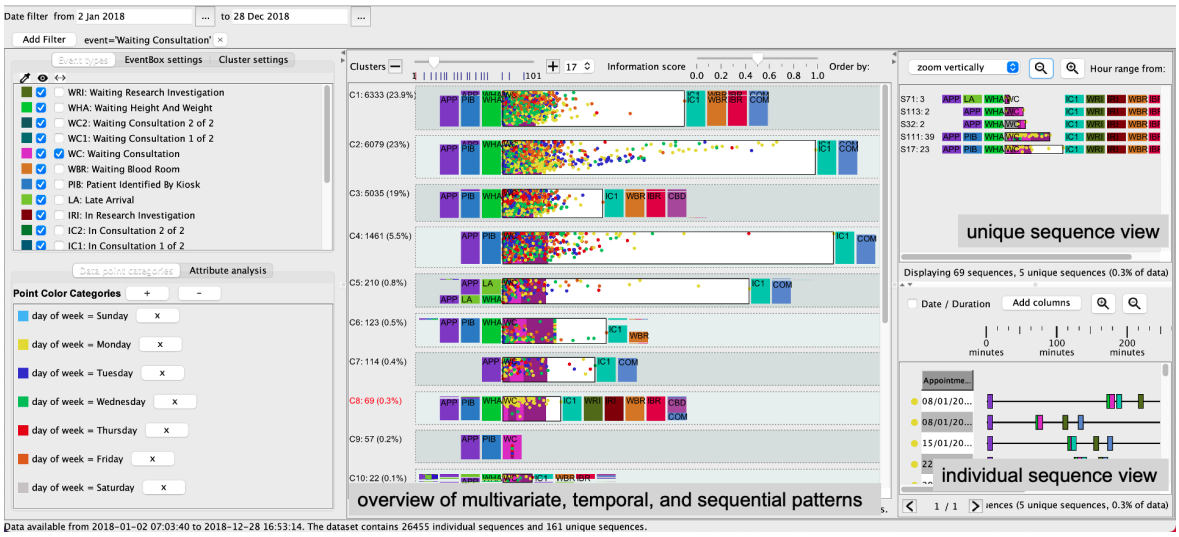


Figure 5.4: *Coordinated views of the system Sequen-C. Left upper panel shows the event types in the dataset, where the event type waiting consultation (WC) is selected to show EventBox and clusters are aligned by WC. The left bottom panel shows the colour categories for the attribute day of week. The overview shows 17 sequential patterns numbered from C1 to C17, the sequential pattern C9 (in red) is selected and shown in the unique sequence view and individual sequence view.*

similarity, where the default k is obtained through the average silhouette width, and clusters are visually encoded using the steps Align-Score-Simplify. The number of clusters can be changed to explore the pathways at different levels of detail, allowing users to discover a set of sequential patterns that describe the dataset **(T1)**, while also identifying pathways that differ from the common ones **(T8)**.

Selected event types in the overview can be expanded to show an aggregation of their temporal and multivariate attributes via EventBox **(T6)**. The panel on the top left shows a list of the event types in the dataset, there are three different controls next to each event type: 1) a control to change the colour of the event type; 2) a checkbox to show or hide the event type in the overview; 3) a checkbox to transform the visual encoding of all the aggregated events of that event type using the EventBox visualisation. Fig. 5.4 shows the event type *WC* visually encoded as an EventBox, data points in an EventBox can be selected by drawing a selection with the mouse; these can include single points such as duration outliers **(T8)** or a group of points shown across the quarters of the box. Selected data points are highlighted in the bar charts of the attribute analysis view, and the sequences containing the selected data points are added to the individual sequence view. This allows users to obtain further detail of interesting data points such as those with an outlier duration **(T4)**.

5.4.2 Unique sequence view

A unique sequence groups individual sequences sharing the exact same event sequence. Unique sequences are presented in this view as ordered lists of events which height is proportional to the number of individual sequences. Selected sequential patterns in the overview are added to this view to inspect the sequences contained in each cluster in more detail **(T1,**

T4). Event types in the unique sequence view can also be expanded and visualised using EventBox. The difference with the overview is that this view will aggregate events at unique sequence level rather than at cluster level **(T6, T7)**.

5.4.3 Individual sequence view

This view shows individual sequences from specific data points selected in EventBoxes or all the individual sequences corresponding to selected clusters. A gantt chart visualisation shows an individual sequence as a lists of ordered events positioned according to their time of occurrence, and a table of attributes is shown next to the gantt chart, where each row shows the raw attribute values of each individual sequence **(T4)**.

5.4.4 User interaction

The following user interactions are supported in order to transform the overview and the visual encoding of an EventBox.

Colour categories for data points of EventBox: the data points in an EventBox are coloured according to their sequence or event attribute values, *colour categories* can be created to assign a colour to the values or range of values of an attribute **(T7)**. A set of colour categories can be created automatically by assigning a different colour to each unique value of an attribute (e.g. yellow for male, blue for female, red for unknown).

Alternatively, users can create customised colour categories by assigning a colour to data points that are true for a query. A query is specified using three controls: 1) attribute, any multivariate attribute in the dataset; 2) operator, including $<, \leq, >, \geq, =, ! =$, is one of, starts with; 3) open text value or list of values obtained from the unique values of the attribute. This allows users to discretise attributes with too many possible values and to create queries that are reflected in the colour of data points **(T3)**. For example the colour category: {colour: blue, query: age $\geq$ 30} means data points of individuals 30 years old or more are coloured in blue. After adding a colour category, the EventBoxes shown in the overview are transformed accordingly. If a data point matches more than one category then the last one added is taken.

Using colour to represent data point categories in addition to event types, can result in over-use of colour and make it complicated for users to differentiate individual visual items (e.g. Fig. 5.5-b). Future work is needed to identify colour schemes that address this challenge. Moreover, shapes (e.g. x, *, o), rather than colour, could be used to encode data point categories; and event types could be encoded in a gray scale where only a subset of event types are coloured. Alternatively, an interaction could be added so a selected data point category is highlighted from the rest.

Break down an EventBox into colour categories: As observed in Fig. 5.5-6, users can visualise the break down of an EventBox into sub-boxes, as many as the number of colour categories, by double clicking over an EventBox. This can help one to compare the distribution of data points depending on their colour category **(T6, T7)**. Note that the height of each sub-box will be proportional to the number of records containing that colour category. Whilst this helps to compare frequency amongst categories, this also results in different vertical axis scales for each sub-box, which could complicate comparing the time of occurrence of data points across sub-boxes. To solve this, there could be an option to give the

same height to all sub-boxes regardless of their frequency, and then indicate this frequency with a number next to each sub-box.

Transform EventBox to reduce visual clutter: users can transform an EventBox according to the levels-of-detail defined on subsection 5.3.2. This helps to focus on specific components of the EventBox and also reduce visual clutter **(T3, T4)**.

Align sequential patterns by a selected event type: alignment by a selected event is a common task in event data visualisation to identify precursor or aftereffect events [98]. In this system, users can select an event type by which all sequential patterns are aligned. Alignment improves the comparison of attributes across sequential patterns, for example, as shown in Fig. 5.4, event type *WC* is shown as EventBox and selected as alignment point. This allows users to see how the distribution of the duration and other attributes in event *WC* change depending on the sequential pattern **(T6)**.

Order by similarity or frequency: sequential patterns can be ordered either by similarity (according to the hierarchical clustering) or by frequency (according to the number of records they contain). Ordering by similarity can help identify the sequential patterns that are the most different from the rest **(T8)**.

Filters: records can be filtered according to the attributes in the dataset, including event and sequence attributes, or filters by a date range or by a specific day of week or month.

5.5 Example scenario: rheumatology clinic

To demonstrate the capabilities of the technique presented in this chapter, more specifically of EventBox, a real-world dataset from a Rheumatology outpatient clinic was used. The dataset corresponds to one year of data and was provided by Sheffield Teaching Hospitals - NHS Foundation Trust, in the United Kingdom. The details of the sequences in this dataset can be found in Table 5.1.

Table 5.1: *Characteristics of Rheumatology dataset.*

Dataset	Event types	Avg. / Max length	Individual sequences	Unique sequences
Rheumatology	18	8/13	26,455	161

The understanding of patient flow is an area where good data analysis is critical [8]. Waiting times, lengths of stay and clinical pathways are key aspects to the study of patient flow. On average, the Rheumatology department has an approximate annual load of 9,000 patients and 25,000 appointments. Patient visits at this clinic are routinely tracked using an in-house workflow tracking system, where the clinical staff (e.g nurses, receptionist, consultants) input the current state of a patient according to the service being provided. The produced event logs are used by the hospital to obtain basic statistics about the quality of care being delivered, particularly focusing on the study of waiting times and lengths of visit.

The current analysis offers the possibility of delving into the raw event logs to extract key insights about patient flow within the clinic. After loading the data in the system Sequen-C and after interacting with it as per the analytic tasks presented in Chapter 3, the findings shown in 5.5 were obtained.

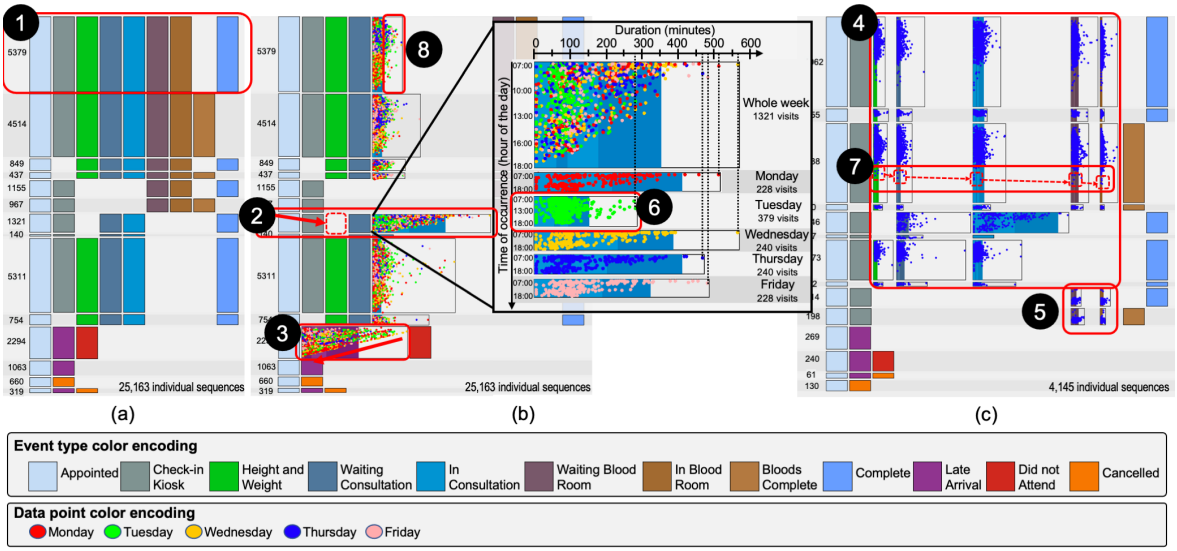


Figure 5.5: *Findings obtained from the Rheumatology dataset. a) Overview of sequential patterns which shows events as point events. b) The events "In Consultation" and "Late Arrival" are encoded using EventBox. c) A filter is added to only show sequences happening on a Thursday. The highlighted numbers represent findings which are explained in the example scenario.*

Discovering main sequential patterns: The most frequent unique sequence is the clinical pathway Appointed → Check-in Kiosk → Height and Weight → Waiting Consultation → In Consultation → Waiting Blood Room → In Blood Room → Complete (Finding 1 in 5.5-a).

Impact of an event in the duration of subsequent events: The time of the event In Consultation is considerably longer when the event Height and Weight does not occur in that sequence (Finding 2 in 5.5-b). In an interview with the clinical staff, they indicated that those patients, without Height and Weight, visit the clinic to undergo longer day-case procedures rather than conventional consultations. A day-case procedure will likely include a consultation plus extra activities such as intravenous infusion of drugs, X rays and other imaging tests; hence the long duration of this event.

The proposed technique allows users to identify scenarios where there is variation in the duration of specific events across sequences, which might be related to the presence or absence of certain events. For example, the sequences ABCDE and BCDFG, share the subsequence BCD. This class of finding suggests that if the duration of BCD varies between the two sequences, that variation might be associated to the occurrence of the event A.

Trends in duration with regard to time of occurrence: In Finding 3 in 5.5-b, the duration of the late arrival event decreases as the day goes by; meaning that the amount of minutes that patients are late are higher in the morning than towards the end of the day. The highest durations (4th quartile and outliers) are concentrated in the morning (between 9:00 and 12:00). This requires a further investigation of morning appointments, revising the reasons for morning late arrivals.

Temporal distribution of time attributes: The proposed visualisation helps to identify what is the "normal" duration of an event, as well as identifying the distribution of points

through time of occurrence. The following was found:

- Distribution of time of occurrence: On Thursdays, the visits to the clinic feature a higher concentration before noon (Finding 4 in 5.5-c). However, the visits for which the purpose is exclusively a blood test occur in the afternoon (Finding 5 in 5.5-c).
- Distribution of duration: On Tuesdays, consultation times are significantly shorter, which suggests that the clinic running on that day might be dealing with less complex pathologies (Finding 6 in 5.5-b).
- Unusual times of occurrence: On Thursdays, the majority of patients visit the clinic in the mornings. Nevertheless, a reduced number of patients are seen in the afternoons (Finding 7 in 5.5-c). Investigating that cohort of patients would be of interest.
- Unusual duration: The present method allows for the identification of patients that stay in an event for an unusual amount of time. Patients that have stayed in an event for an unusual time are represented as outlier data points in the proposed visualisation, these are visually identified as points outside the coloured sub boxes (e.g. see highlighted outliers in 5.5-b, Finding 8).

5.6 Summary

This chapter has proposed a generic methodology to analyse sequential patterns along with their temporal and multivariate attributes using a single visual overview. Building on the work presented in the previous chapter, the multilevel overview, which already presents an overview of sequential patterns, integrates multivariate attributes for selected event types via EventBox. This produces a multilevel and multivariate overview able to represent complex patterns related to multiple variables. An EventBox is a novel visual encoding, inspired by box plots and scatter plots, that aggregates the duration, time of occurrence, and categorical attributes (e.g. age, gender, country) of a set of event occurrences of the same type, and allows users to identify trends and outliers with respect to these attributes. The chapter has proposed different levels of detail for an EventBox to reduce visual clutter and focus the analysis on specific areas (e.g. outliers).

The system Sequen-C was extended to support EventBox and to take into account the analytic tasks, which in general, reflect the requirement of obtaining the main sequential patterns in the dataset and then identify trends involving data attributes within and across sequential patterns. The technique, specifically EventBox, has been demonstrated through a real-world dataset from a Rheumatology outpatient clinic, for which a set of findings was obtained.

The next chapter presents an extended case study performed using a dataset provided by the Centre for Urgent and Emergency Care Research (CURE), the case study evaluates the full methodology presented in chapters 4 and 5.

5.7 Limitations

Depending on the application domain, EventBox could present limitations in terms of scalability: as the ranges of duration $Q_4 - Q_0$ and time of occurrence $T_N - T_0$ become larger, or as the volume of the data points in an EventBox increase. To solve these potential limitations:

time windows, zoom and scaling functions could be used. One of the main challenges of the EventBox visualisation is the overlapping of data points (i.e. over-plotting) as the number of records increase, which is also a common issue in traditional scatter plots and visualisation in general. This over-plotting can *hide* patterns, and make it difficult to distinguish individual visual items or identify dense areas containing high number of overlapping data points. These issues can be addressed by sampling the data and presenting only a proportion of data points, or by using a density chart such as a 2D histogram where the number of data points are represented by a color gradient [104]. Other alternatives include amongst others: stacking overlapping cases in a 3D plane [21], representing overlapping data points with a circle proportional to the number of cases (i.e. bubble plot), or using pixel based mappings [78].

Chapter 6

Case study: CUREd dataset

This chapter presents a case study using the CUREd dataset [48], conducted along with 3 members of the Centre for Urgent and Emergency Care Research (CURE). In contrast with the case studies presented in Chapter 4, the CUREd case study demonstrates the full technique presented in both Chapter 4 and 5.

The CUREd dataset contains calls made to the emergency service, which can lead to different pathways, including ambulance conveyance to the Emergency department (ED) and admissions to inpatient facilities. This case study shows the ability of Sequen-C to obtain sequential and multivariate patterns from temporal event sequences. Firstly, section 6.2 presents a coarse overview of the four main pathways that patients usually follow after a call is made to the emergency service, including patients whose calls are closed and do not lead to any ambulance service. Secondly, section 6.3 focuses on calls that lead to a visit to the emergency department, obtaining a fine detail overview of the variations of this pathway. The analysis of attributes allows to characterise pathways of interest, obtaining findings such as "calls coming from area code 8C3 or from people in their 20s can usually be handled without an ambulance service resource attending."

6.1 Background

The CUREd research database [48] is a unique resource that contains timestamped events and demographic data related to telephone calls made to the emergency service (calls to 999 or 111) throughout Yorkshire and the Humber region. Calls can lead to different pathways, including ambulance conveyance to the Emergency department (ED) and admissions to inpatient facilities. A three-month subset of the dataset was used, containing 25,243 calls relating to 21,805 unique patients, and 57 data attributes. A table with the description of the data attributes used in this case study can be found at Appendix A.

The data were processed so that an individual sequence represents all the events of multiple calls and incidents for the same patient. The events in the provided dataset only contained a start date timestamp but they did not contain an end date. The data was processed so the start date of an event is the same as the end date of the previous event in the sequence. Therefore in this case study, the duration of an event represents the time elapsed between that event and the next one. The details of the sequences in this dataset can be found in Table 6.1.

Table 6.1: *Characteristics of CURE dataset*

Dataset	Event types	Avg. / Max length	Individual sequences	Unique sequences
CUREd	11	22.7 / 177	21,805	962

Analysis sessions

Multiple analysis sessions were conducted along with 3 members of the Centre for Urgent and Emergency Care Research (i.e. CURE team). The CURE team were not only *problem owners*, but they were also *data owners*, who were interested in obtaining a summary of the pathways that originate from the calls as well as the characteristics of the patients following such pathways. This case study was developed following an iterative process, through several analysis sessions via video call during the COVID pandemic. The researcher, her supervisor, and the CURE team were present in these video calls. An analysis session would typically consist of the following steps:

1. The system was shown to the CURE team, where the researcher would interact with the system to demonstrate its main features and the visualisations produced with the CUREd dataset.

2. Based on the visualisations, either one of the three domain experts would request to obtain more information of a specific pattern, or propose a real-world scenario that could be worth exploring. The researcher would interact with the system according to the experts requests or questions. All relevant findings obtained during these meetings were added to a Google Document shared amongst the researcher, her supervisor, and the CURE team.

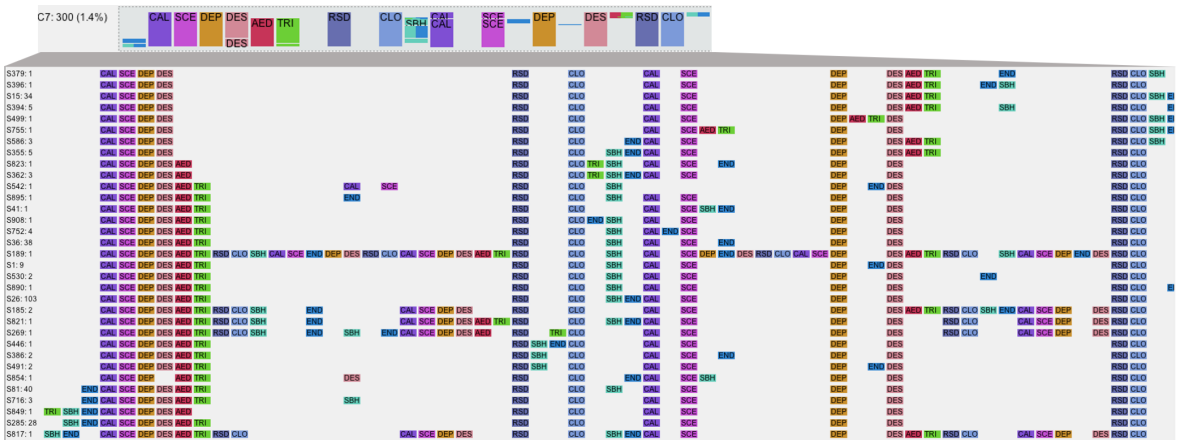


Figure 6.1: *Unique sequences in cluster C7 (from the overview k=200), showing an example of a cluster with small variability.*

3. The questions that could not be answered due to a limitation or a non existing feature of the system would then be turned into a new requirement. For example, during one of these sessions the need of having the attribute analysis view was identified, to explore the demographics of patients.

4. During the days after the analysis session and the next scheduled meeting: the researcher would develop any new features resulting from the meeting, and the CUREd team would comment the Google Document containing the findings to request additional information or post new questions. A new meeting would take place to review the new features, questions and findings posted in the Google Document. The process was repeated iteratively starting again from step 1.

6.2 Overview of main pathways

After loading the data, by default, the overview shows **200 clusters** ordered by frequency. Clusters are well separated and, as observed in Fig. 6.1, present small intra-cluster variability (i.e. sequences in the same cluster are very similar). The first four clusters represent 85% of the data and the remaining patterns are repetitions or variations of the main four. To get a coarser overview, the **number of clusters is changed to 4** (see Fig. 6.2-A). These clusters cover the pathways:

- **Cluster C1** (34.9%): Ambulance service and attendance to the emergency department.
- **Cluster C2** (24.7%): Ambulance service and conveyance to a hospital.
- **Cluster C3** (23.6%): Ambulance arrives at the destination but no conveyance is made.
- **Cluster C4** (16.8%): Call closed and no ambulance service.

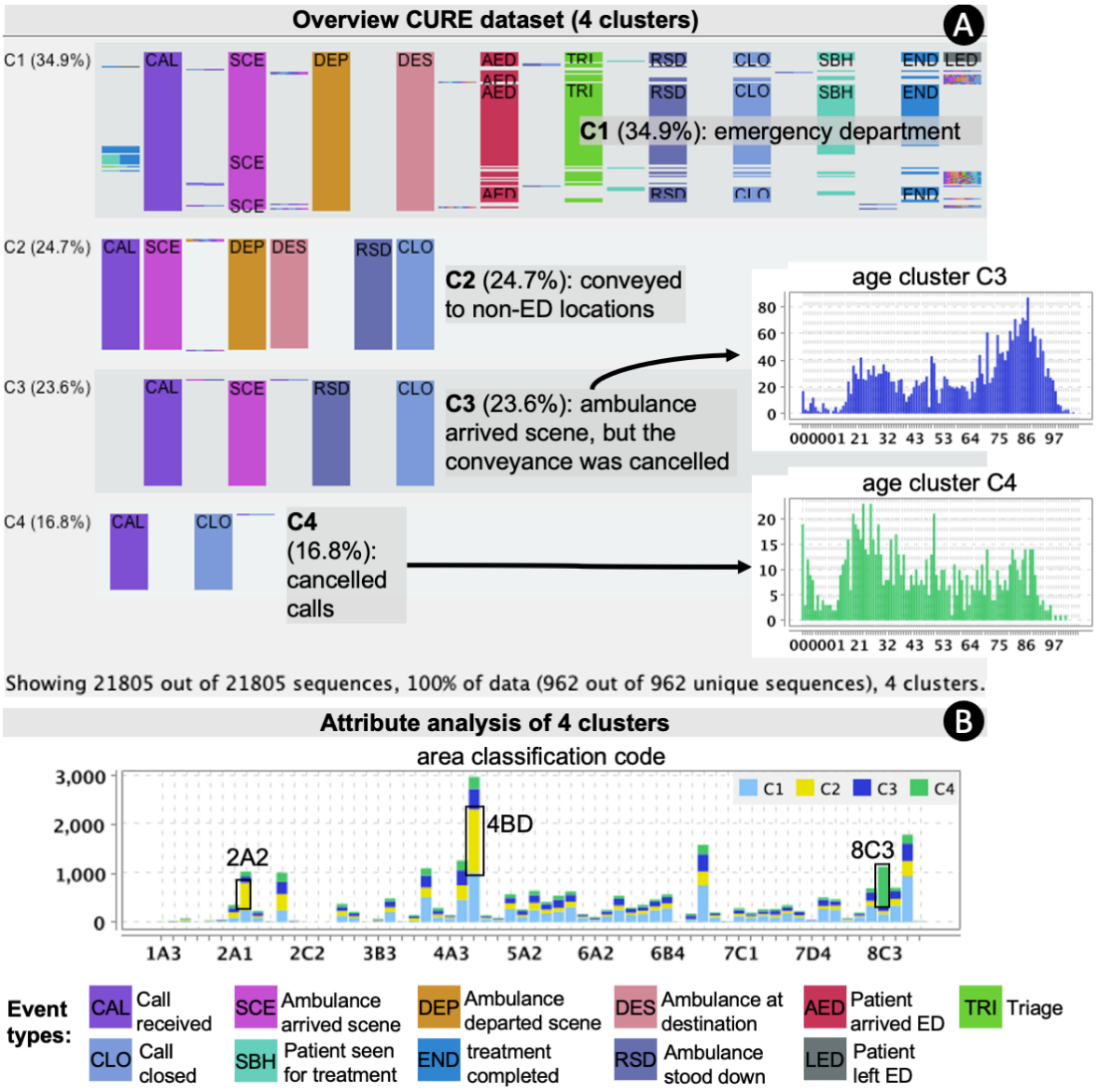


Figure 6.2: *21,805 patients who made calls to emergency services, obtained from the CUREd dataset, partitioned in 4 clusters. (A) Coarse overview showing main pathways (clusters C1 to C4). (B) Attribute analysis for area classification code and age (for clusters C1 to C4).*

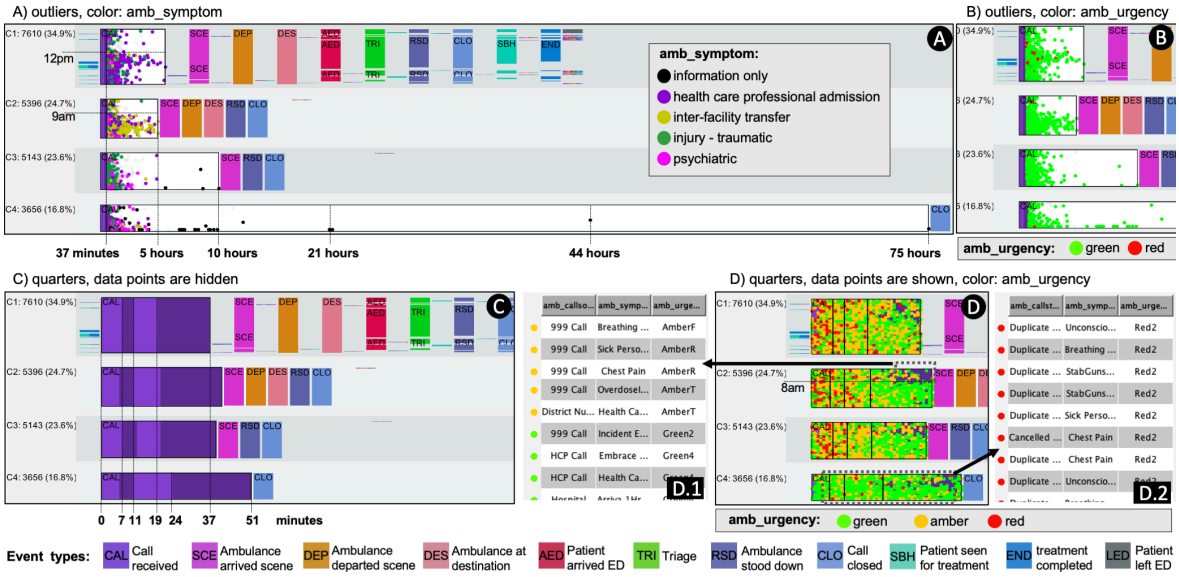


Figure 6.3: *Screenshots of an overview of 21,805 patient pathways originated from calls made to the ambulance service, patient pathways are divided into 4 clusters. The EventBox in each cluster represents the duration from the first call in the sequence to the next event. A) Calls with an outlier duration are shown and coloured by the attribute symptom (amb_symptom). B) Calls with an outlier duration are coloured by their urgency level (amb_urgency). C) Outliers are hidden and EventBoxes are scaled up to review quartiles. D) Data points in the quarters are shown, coloured by their urgency level (amb_urgency). D.1) Further details of anomalous calls in the fourth quarter of cluster C2 happening before 8am. D.2) Further details of anomalous red data points in cluster C4.*

6.2.1 Characteristics of main pathways

The attribute ***area classification code*** is a classification based on socio-economic and demographic information derived from the postcode of the incident. Comparing the area classification code of the four clusters (see Fig. 6.2-B), some clusters predominate more in certain area codes than others. Hospital transfers (cluster C2) happen for 43.5% of calls coming from area code 4BD and 53.1% of calls coming from area 2A2, while 72.3% of calls coming from area 8C3 are in cluster C4. The **age attribute** (*amb_callage*) indicates that cluster C4 is more common amongst younger people, meaning that calls from area 8C3 or people in their 20s can usually be handled without an ambulance service resource attending. Conversely, cluster C3 is more common for people in their 80s (see Fig. 6.2-B). According to the attribute ***symptom***: 59.7% of calls due to chest pain end in an attendance to the emergency department (cluster C1) whereas 45.3% of calls due to a psychiatric incident are in cluster C4.

6.2.2 Duration of calls according to characteristics such as urgency level

The analyst wanted to compare the duration, time of occurrence, and characteristics of calls across the 4 clusters. The event type **call (CAL)** is selected to be visually encoded as an

EventBox and clusters are aligned by the event CAL. The EventBox in clusters C1 to C3 represent the time elapsed from the start of the call to the time in which the ambulance arrives at the scene (i.e. CAL → SCE), whereas the EventBox in cluster C4 represents the duration or time elapsed from the start of the call to its closure (i.e. CAL → CLO).

By default **duration outliers are shown** and quarter data points are hidden. Fig. 6.3-A shows that the longest duration outliers in cluster C2 happen after 9am, whilst the longest duration outliers in C1 usually happen around the afternoon. Outlier data points across the clusters have a duration between 37 minutes and 5 hours. However, cluster C3 and C4 contain some outliers with duration greater than 5 hours. Adding the **attribute amb_symptom** as the colour category of the data points, allows one to see that the outliers with duration greater than 5 hours have the symptom *information only* (Fig. 6.3-A). The symptom attribute indicates that the duration outliers in the 4 clusters are usually due to: health care professional admissions, inter-facility transfers, injuries, or psychiatric incidents. The analyst mentioned that these outliers have such high duration because they might have been assigned a low urgency level at the moment of the call. To validate this, the **attribute amb_urgency** is added as a colour category; Fig. 6.3-B shows that effectively, most of duration outliers correspond to green or low level urgency calls.

Outlier data points are removed from the EventBoxes by deselecting the option "show outliers" in the upper left panel, then the width of the EventBoxes was scaled up to show more details. Fig. 6.3-C shows the quartile values for the EventBoxes across the 4 clusters, which indicate that the median elapsed time from a call to the next event is around 11 minutes.

The analyst wanted to study the response time after a call has been made, depending on the urgency level of the call. The option **"show data points in quarters"** was enabled - data points are coloured according to the current **colour category amb_urgency**, comparing low (green), medium (amber), and high (red) urgency level. Fig. 6.3-D shows that calls with a high urgency level are mostly in the first three quarters of the EventBoxes (i.e. their duration is below 24 minutes); with the exception of few red data points in the fourth quarter. Moreover, there are some red data points in cluster C4, meaning that some calls categorised as urgent ended up being closed without ambulance service. These red data points are considered anomalous scenarios and can be selected to find out more details. For example, the red data points in cluster C4 are selected and added to the individual sequence view; Fig. 6.3-D.2 shows the **amb_callstop** attribute for each of the selected red data points in cluster C4, which indicates that these urgent calls are usually closed due to duplication or cancellation by the caller. For the 4 clusters and specially in cluster C2, fewer data points in the fourth quarter occur before 8am. The **amb_callsorc** column in the individual sequence view of these data points, indicates that the source of most of these calls is 999, HCP call, or hospital (Fig. 6.3-D.2); and that usually 999 calls are assigned an amber urgency level while calls coming from the hospital are assigned a green level urgency.

6.3 Calls leading to the emergency department

To further explore calls in cluster C1, a filter is applied to show only sequences containing at least one emergency department event. The analyst chooses **11 as the number of clusters**, suggested by the system as one of the optimal number of clusters, and orders clusters by

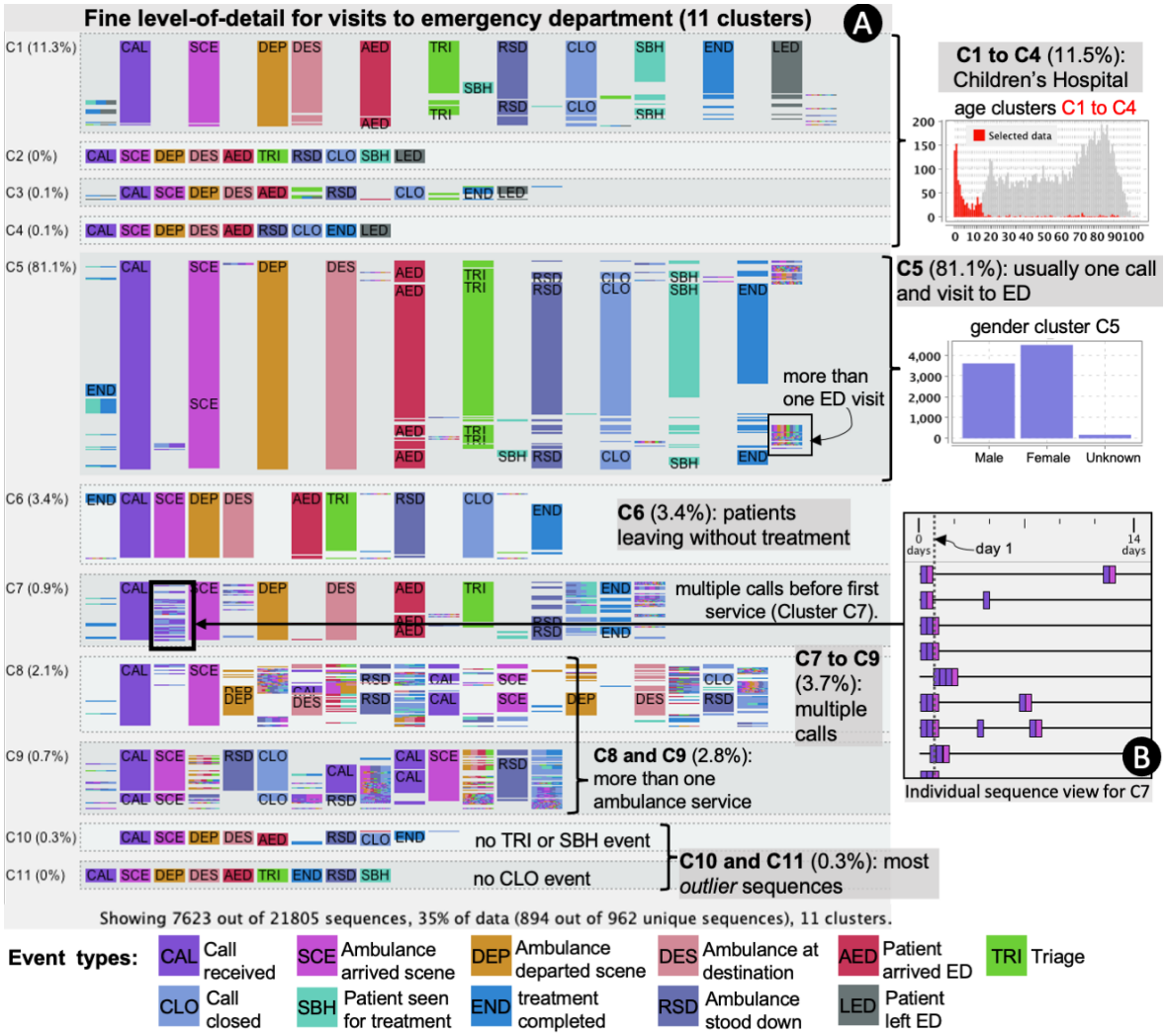


Figure 6.4: *(A) Overview of sequences containing at least one visit to the emergency department (AED), partitioned in 11 clusters. (B) Gantt chart showing events CAL and SCE (for cluster C7), where the majority of repeated calls before the first service happened on the same day.*

similarity. Fig. 6.4-A shows that the first six clusters (C1 to C6) contain about 96% of the filtered data and represent patients with usually only one call to the emergency service. These six clusters categorise visits depending on whether the visit to the emergency department is either followed by a triage event (TRI), seen by a health professional (SBH) to arrange treatment, or both. According to the analysis of the attribute ***attendance disposal*** (i.e. how the visit was concluded), in most cases where a triage event is not followed by an SBH event (clusters C3 and C6) is because the patient *left the department before being treated.* Cluster C5 contains the highest percentage of data, grouping 6180 individual sequences (81.1% of data); the attendance disposal attribute for this cluster indicates that 52% of patients in this cluster were admitted to a hospital bed, 17.4% were discharged with a follow up treatment, and 23.2% were discharged without follow up. According to the **sex attribute**, cluster C5 is slightly more common amongst women (Fig. 6.4).

Interestingly, the event "patient left emergency department" (LED) is only present in clusters C1 to C4. To find out more, the attributes of these clusters are analyzed (see Fig. 6.4), the age of the patients go from 0 to 15 years old and the hospital attribute shows the Children's hospital as the only ambulance destination. The absence of the **event LED** in other hospitals might be due to a different configuration in the event log capturing system.

Clusters C7 to C9 suggest that about 3.7% of patients have called the emergency service twice or more. In order to find out how many calls were made before the first ambulance service resource attended: clusters C7 to C9 are aligned by the first occurrence of the events CAL (call) and SCE (ambulance arrived scene) (see Fig. 6.4-A). Sequences in cluster C7 contain many more calls before the first SCE event, which suggests that these patients had to "try" more times to get an ambulance service for the first time compared to clusters C8 and C9. To investigate whether these multiple calls were made on the same day the ambulance service was provided, cluster C7 was added to the individual sequence view. Fig. 6.4-B shows that most of these **multiple calls were made on the same day** and therefore relate to the same incident. Individual scenarios in this cluster need further exploration.

6.3.1 Waiting times

A patient's pathway through the emergency department can contain several waiting time periods. The analyst was interested in the **duration of the event DES** (ambulance arrived destination), which represents the wait of the ambulance arriving at the hospital before the patient is transferred to the hospital's care. Fig. 6.5-1 shows that the duration of the event type DES across all clusters has a median duration of around 6 minutes and maximum duration of about 14 minutes (without outliers). However, the duration range of the fourth quarter of the EventBox in cluster C8 is considerably larger ($Q_3 = 18$ to $Q_4 = 37$ minutes). Compared to other clusters, cluster C8 contains **many additional events** between the event DES and an attendance at the emergency department (AED) - see Fig. 6.5-2. In order to find out if these extra events are related to the larger duration of DES, the data points in the fourth quarter of the EventBox are selected. The unique sequence view (Fig. 6.5-3) allows one to see that effectively, for the selected data points, after the first DES occurred the call was closed (CLO) and the ambulance was stood down (RSD), but then one or several new calls were made before attending the emergency department. According to this, the ambulance might have arrived the hospital (DES) but these patients were not admitted to the emergency department until a later call and second transfer was made. In the individual sequence view,

the **amb_callstop** column allows to see that most of these calls have a value NA or Vehicle not required. These anomalous scenarios require further investigation.

The **duration of the event type AED** (patient arrived emergency department) represents the waiting time in the emergency department before the patient is either triaged (i.e. initially assessed to determine the priority for treatment) or receives treatment. The event type AED is expanded to show EventBox. By default, data points are coloured by the **day of week**, something that caught the analyst's attention immediately were the yellow outlier data points in cluster C5 which occur on a Monday (see Fig. 6.6). These outlier data points lasted more than hour and a half, and happened after 6pm. When selected to see further details, it is observed that most of them occurred on the 4 of April of the same year; also most of these patients are 60 or above. What happened that day needs further investigation. These types of findings could help the stakeholders in studying specific anomalous scenarios and improve the health care delivery service.

6.4 Summary

The case study has shown the ability of Sequen-C to obtain sequential and multivariate patterns from temporal event sequences. It showcases the impact of the method when applied to real-world datasets. This case study firstly presented a coarse overview of the four main pathways usually followed by patients after a call is made to the emergency service, including: 1) patients visiting the emergency department, 2) patients conveyed to a hospital, 3) ambulance service provided but no conveyance made, and 4) calls closed without an ambulance service being provided. The second part of the analysis focused on patients visiting the emergency department, obtaining finer details of the variations of this pathway and their characteristics. The analysis of attributes allowed to characterise patients following pathways of interest. The analysts mentioned that this type of analysis is really useful as it provides "insight into which calls are likely to need a hospital transfer and which may benefit from a different response" and that "knowing these patterns might help assist decision making for call handlers".

The CUREd dataset is a wide and rich source that contains many patient and call level attributes. However, it was found that in some cases the attributes values are incomplete and do not seem to be captured consistently. For example, the attribute *diagnosis* contains the value *not classifiable* for most of the records. The type of analysis presented in this thesis could serve as a motivation to gather more complete and consistent data throughout all attributes, as this could lead to further, more complex insights.

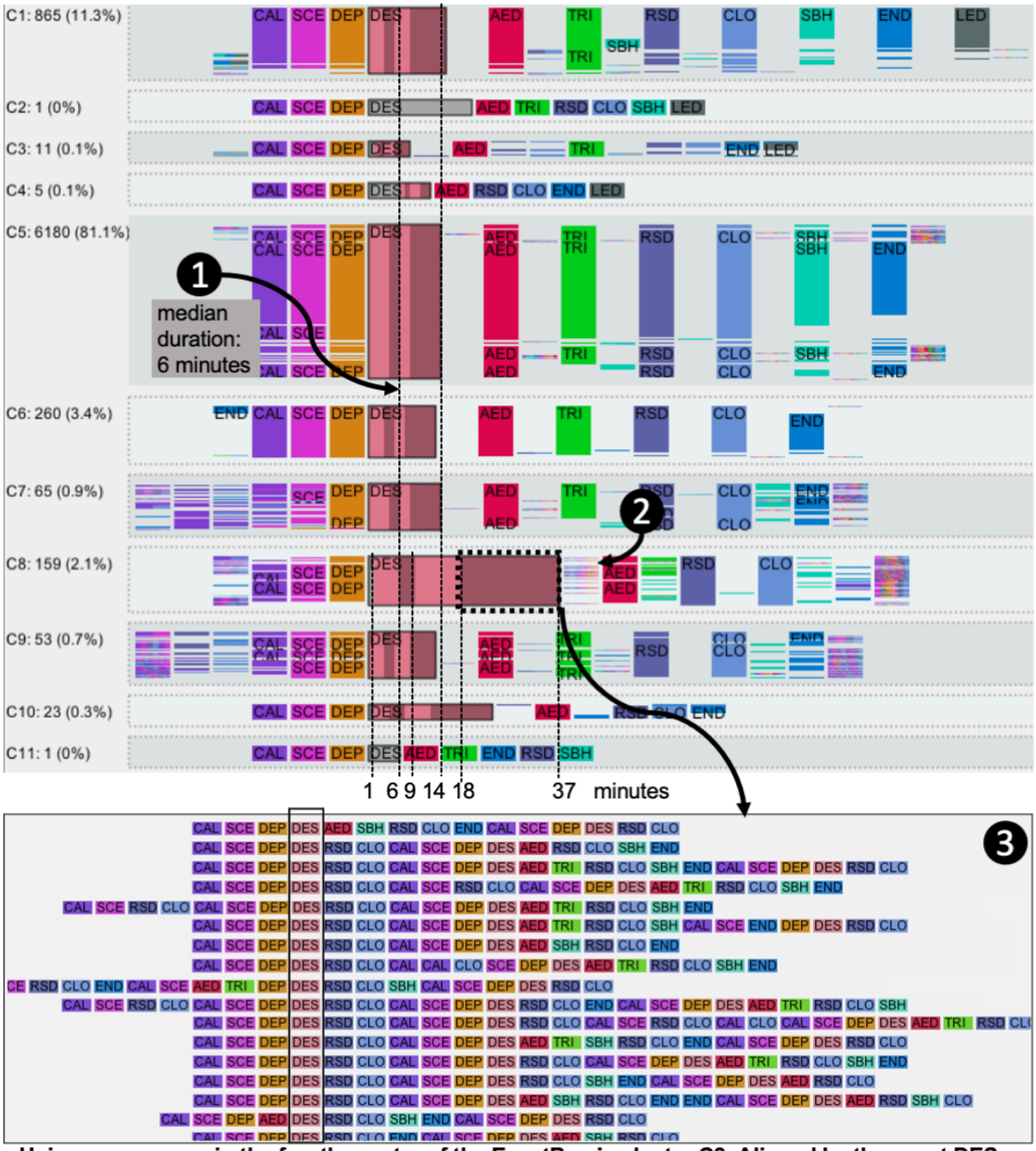


Figure 6.5: *Visits to the emergency department partitioned in 11 clusters. The event type DES (ambulance arrived destination) is encoded using EventBox. (1) DES has a median duration of 6 minutes, however, the duration in cluster C8 is considerably longer compared to other clusters. (2) The longer duration could be related to the extra events between the events DES and AED, note these extra events are not present in other clusters. (3) The data points in the fourth quarter belong to the sequences that have such extra events.*

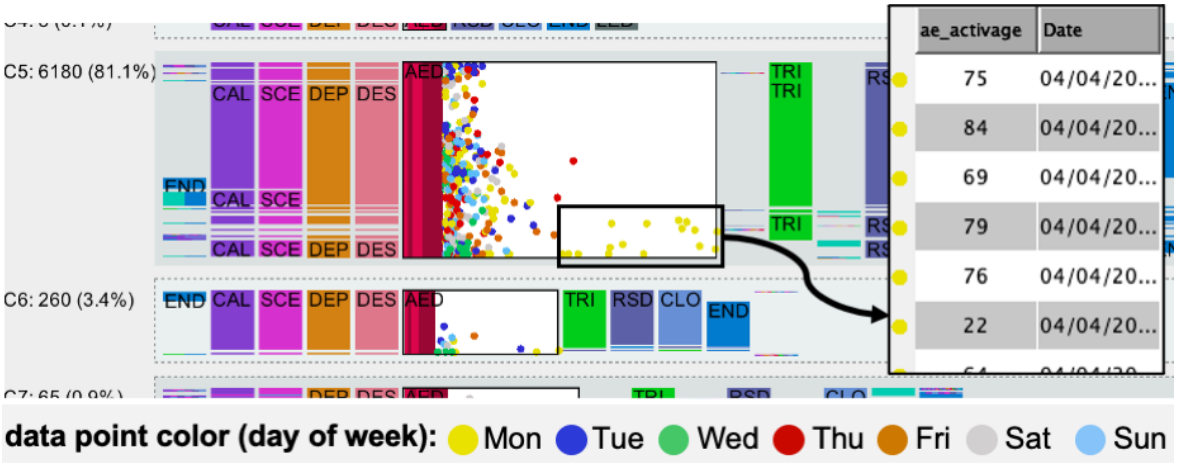


Figure 6.6: *The event type AED (patient attended Emergency Department) is encoded using EventBox, the event represents the waiting time to be triaged or receive treatment. The highest anomalous waiting times on cluster C5 occurred on a Monday on the exact same date.*

Chapter 7

Evaluation

The present chapter evaluates the adoption of the system by a novice user that interacts with the system for the first time. The performance of novice users is then compared with that of two expert users. The purpose of this evaluation is to quantify and qualify: 1) the interpretability of EventBox and the proposed cluster representation; 2) the usability of the system Sequen-C; and 3) the ability of the system to allow users to achieve the analytic tasks presented in Chapter 3. This evaluation is measured ***quantitatively*** through a set of tasks, using completion time and accuracy as the metric, and through the System Usability Scale (SUS) questionnaire [16]. Additionally, ***qualitative*** feedback is gathered from participants in written form and verbally while conducting the evaluation.

Firstly, section 7.1 describes the details of the evaluation, outlines the tasks presented to participants (section 7.1.1) and describes the SUS questionnaire (section 7.1.2). Secondly, section 7.2 presents the results for completion time and task accuracy, comparing the performance of the novice users versus the expert users. Thirdly, section 7.3 analyses the answers to the SUS questionnaire. Lastly, section 7.4 presents the qualitative feedback provided by participants.

7.1 Design

This thesis has presented case studies using four real-world datasets in different domains. These are an evaluation of the generalisability and potential of the technique to obtain insights in real-world scenarios. The case studies were co-developed with a number of stakeholders with professional backgrounds ranging from medicine to computer science. However, the PhD researcher, based on the questions of the stakeholders, was the one effectively interacting with Sequen-C. The intended user of this system is any person, with any professional background, who wants to analyse and obtain insights from temporal event sequences describing a process of her/his interest. Therefore, this evaluation validates the adoption of the system by a novice user that interacts with the system for the first time, and then compares it with that of two expert users.

Eligible novice users to participate in this evaluation are adults that have never interacted with the software Sequen-C before, but are familiar with basic data visualisations such as bar charts or scatter plots. Novice participants were recruited through social media and messages to colleagues and friends. The target user of Sequen-C is expected to be familiarised with the context of the data being explored (i.e. domain expert). However, as all our domain experts had already been involved in the design phase and case studies, it was decided to do the evaluation with novice users. Evaluating the system with novice users might not reflect the exact process in which findings are normally obtained, mainly because the user needs to be familiarised with the data to formulate queries. Note that this evaluation does not aim to validate how findings are obtained, but rather to measure how easily the proposed visualisations are interpreted by new users.

As this evaluation involves human participants, ethics approval was sought and obtained from the University of Sheffield. Appendix B includes the relevant documents. The evaluation was carried out through a video call where only the participant and the researcher were present. The participant interacted with the software by remotely accessing the researcher's desktop through the application TeamViewer. The dataset from the Antenatal Care Unit case study was used in this evaluation.

The evaluation consisted of four main steps. Firstly, the researcher presented a demo of Sequen-C to the participant, who was introduced to the concepts of event sequences, clustering, and alignment - as well as how to interact with the system and interpret the visualisations. Secondly, participants were asked nine test questions to make sure they had learnt the basics of the system and they were free to ask questions to the researcher to clarify concepts. Thirdly, the participant performed a list of thirteen tasks (quantitative evaluation), one at the time. Each task was in the form of a multiple choice question that could be answered by interacting with the system or interpreting the visualisations. Even though an effort was made so that more than one answer seemed to be potentially correct (e.g. avoid participants solving tasks by simply discarding choices), multiple choice answers might still provide hints to participants about how to solve a task and might not reflect how users discover findings in real-world scenarios. However, multiple choice questions were chosen over open ended questions, as the first make it easier to measure correctness and present results in a quantitative way. In the fourth and last step, the participant provided answers to the SUS questionnaire presented in a Google form (usability evaluation). Each evaluation took in total around 2 hours per participant, including the four steps and time to clarify questions.

7.1.1 Quantitative evaluation: analytic tasks

The tasks performed by the participant were designed based on the following analytic tasks, presented in Chapter 3:

T1. Explore common and deviating pathways.

T2. Interpret the sequences that constitute a cluster.

T3. Focus the analysis on a selected set of records.

T4. Obtain details on demand.

T5. Aggregate and compare context information for selected groups of records.

T6. Compare the distribution of attributes within and across sequential patterns.

T7. Identify trends involving multiple variables such as duration, time of occurrence, and categorical attributes.

T8. Identify anomalous scenarios.

For each of the eight analytic tasks, one or more question-based task was designed, resulting in the following thirteen questions and tasks:

Q1. Which one of the following pathways is the most frequent? **(T1)**

Q2. Which one of the following pathways is the least frequent? **(T1,T2,T8)**

Q3. Which one of the following statements is FALSE? **(T2)**

- Not all patients in cluster C4 complete (COM) their visit.
- Across all clusters, there are patients that have three diabetes consultations (IDC) in the same visit.
- In cluster C4, some patients that waited for a consultation (WC) ended up NOT having a consultation (IC).
- Across all clusters, all the patients that have a scan event (IS) also have a consultation event (IC).

Q4. Which of the following events occurs in cluster C2 but does not occur in cluster C3? **(T2)**

Q5. In cluster C7, how many patients had three diabetes consultations (IDC)? **(T2, T4)**

Q6. Filter records with a Waiting Consultation event. **(T3)**

Q7. Change the number of clusters to 9. **(T1)**

Q8. Align clusters by the event Waiting Consultation (WC). **(T1, T2)**

Q9. What is the ID of the patient that waited the longest for a consultation (WC)? **(T4, T8)**

Q10. What is the most popular day of the week where patients in Cluster C3 visited the clinic? **(T5)**

Q11. What cluster has the highest number of patients that received a scan (IS) in room number 5? (see ScanRoom attribute) **(T5)**

Q12. Overall, which of the following clusters has the shortest waiting time for a consultation (WC)? **(T6)**

Q13. A patient will visit the clinic on Wednesday. First, she will have a scan (IS), then she will be referred to the booking team (WBT and IWB), and then she will have a consultation (IC). When is it more likely that she waits more time for a consultation (WC)? **(T7)**

As mentioned, participants had to perform a specific task in order to answer each of the questions above. At the start of each task, the participant had time to read and understand what was required, then the researcher would start a timer to measure the completion time; the timer would be stopped once the participant provided an answer. While the timer was ticking, if the participant did not remember where to find a specific button, tab, or menu, the researcher would remind the participant where to find such option. However, the type of task or steps required to answer a question had to be figured out by the participant alone. For example, for question number 9 (What is the ID of the patient that waited the longest for a consultation?), the participant had to figure out that it was necessary to first click the option to show EventBox in the event type "Waiting Consultation", then she/he had to select the data point with the highest duration and review the ID of that patient in the individual sequence view.

7.1.2 Usability evaluation: System Usability Scale

The System Usability Scale questionnaire [16] is a popular and standardised set of questions used to assess perceived usability [53]. The questionnaire consists of the following 10 questions, where each question is answered with a number from one to five, one meaning *strongly disagree* and five meaning *strongly agree*:

1. I think that I would like to use this system frequently (If I needed to analyse event data).
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.

9. I felt very confident using the system.

10. I needed to learn a lot of things before I could get going with this system.

After the evaluation finished, the novice participants provided answers to this questionnaire using a Google form.

7.2 Results: task completion time and accuracy

Fifteen users took part in this evaluation, seven females and eight males, out of which, thirteen were novice users (P1 to P13) and two were expert users (E1 and E2). This section presents the completion time and accuracy of tasks for the novice users, then these results are compared with the accuracy and time of expert users.

Task completion time and task accuracy are one of the most common metrics to evaluate user performance in information visualisation [50]. The completion time was measured from the time the question was read and understood by the participant, to the time where the task was performed and an answer to the question was provided. The task accuracy refers to whether the answer to the question was correct or not, and in the case of questions Q6 to Q8 whether the task was successfully completed or not.

7.2.1 Task accuracy

Table 7.1 shows the results of task accuracy. The table shows an overall high percentage of accuracy, which reflects that the eight analytic tasks presented in Chapter 3 can be successfully achieved through the system by novice participants. This means that users that have never used Sequen-C before can relatively fast become familiarised with concepts such as event sequences, clustering, and alignment - and then use such concepts to interact with, customise, and interpret the proposed visualisations to successfully obtain insights from the data.

Most questions and tasks were correctly answered except for 4, 1, 1, and 2 participants that provided an incorrect answer for questions Q2, Q8, Q12, and Q13 respectively. The following paragraphs present an analysis of the incorrect answers provided by participants in these four questions.

Question Q2

Question Q2 resulted in an accuracy percentage of 69.2% (four incorrect answers), the lowest accuracy in comparison with the rest of the questions. The question contained the following possible answers:

Q2. Which one of the following pathways is the least frequent?

(a) NYA → LFS → CAN (in Cluster C1)
(b) NYA → PIB → WAO → WS → CTR → IS → COM (in Cluster C3)
(c) NYA → LAT → WBR → IBR → MGW → D2B → IBR → COM (in Cluster C6)
(d) NYA → PIB → WGS → IWG → WC → IC → COM (in Cluster C5)

	count incorrect	count correct	accuracy %
Q1	0	13	100.0
Q2	4	9	**69.2**
Q3	0	13	100.0
Q4	0	13	100.0
Q5	0	13	100.0
Q6	0	13	100.0
Q7	0	13	100.0
Q8	1	12	**92.3**
Q9	0	13	100.0
Q10	0	13	100.0
Q11	0	13	100.0
Q12	1	12	**92.3**
Q13	2	11	**84.6**

Table 7.1: *Number of correct and incorrect answers provided by the thirteen novice participants to the thirteen questions (Q1 to Q13). The last column shows the percentage of correct answers per question across all participants.*

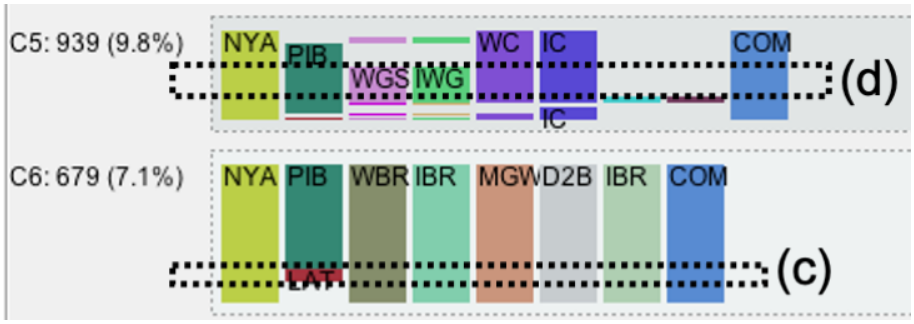


Figure 7.1: *Cluster view for the Antenatal Care Unit (ANC) dataset showing clusters C5 and C6. Participants had to choose the least frequent pathway amongst the options in question Q2. The pathways in options (d) and (c) are highlighted in this figure with a dotted square. Note that in the actual evaluation the dotted lines were not shown and users had to determine the pathway visually. The least frequent pathway is option (c).*

As shown in Fig. 7.1, the correct answer is (c). This question implied that participants first had to visually locate the pathways inside the corresponding cluster, according to the options provided. It was evident that options (a) and (b) were the most frequent pathways as these were noticeably higher. Some participants struggled to choose the least frequent pathway between option (c) and (d). The participants that mistakenly chose (d) as the least frequent stated that because the height of cluster C5 is shorter than cluster C6 that led them to think the pathway in option (d) was less frequent. However, the question refers to the frequency of the pathway inside the cluster rather than the frequency of the whole cluster. Another participant said that she/he misread the option (c) and did not realise that the pathway had to include the event LAT. Comparing options (c) and (d) might be difficult as the height of both pathways is similar. To improve the interpretability of pathways with similar height in the clusters view, an option could be added so when users hover the mouse over an event, the pathway or pathways that contain that event are highlighted and a label with the number of records is shown.

Question Q8

The task presented in question Q8 resulted in an accuracy percentage of 92.3% (one incorrect answer):

Q8. Align clusters by the event Waiting Consultation (WC).

It was explained that alignment by a selected event allows the understanding of the events happening before and after it. The only participant that got this task wrong said she/he did not know what it meant to *align clusters by an event* so she/he did not know what action was required. A hint was provided to guide the participant to the option where the alignment event can be selected, but the participant mistakenly chose the event Waiting Scan as the alignment event. An alternative option to define the alignment event might be necessary in Sequen-C, for example, to align clusters by an event when double clicking on the event of interest. On the other hand, twelve out of the thirteen participants quickly familiarised themselves with the concept of alignment and were able to successfully align by the event Waiting Consultation.

Question Q12

This question resulted in an accuracy percentage of 92.3% (one incorrect answer):

Q12. Overall, which of the following clusters has the shortest waiting time for a consultation?

(a) Cluster C1
(b) Cluster C2
(c) Cluster C3
(d) Cluster C4

This question can be answered by clicking the corresponding checkbox to enable EventBox in the event Waiting Consultation (see Fig. 7.2-1). Fig. 7.2-2 shows that the correct answer

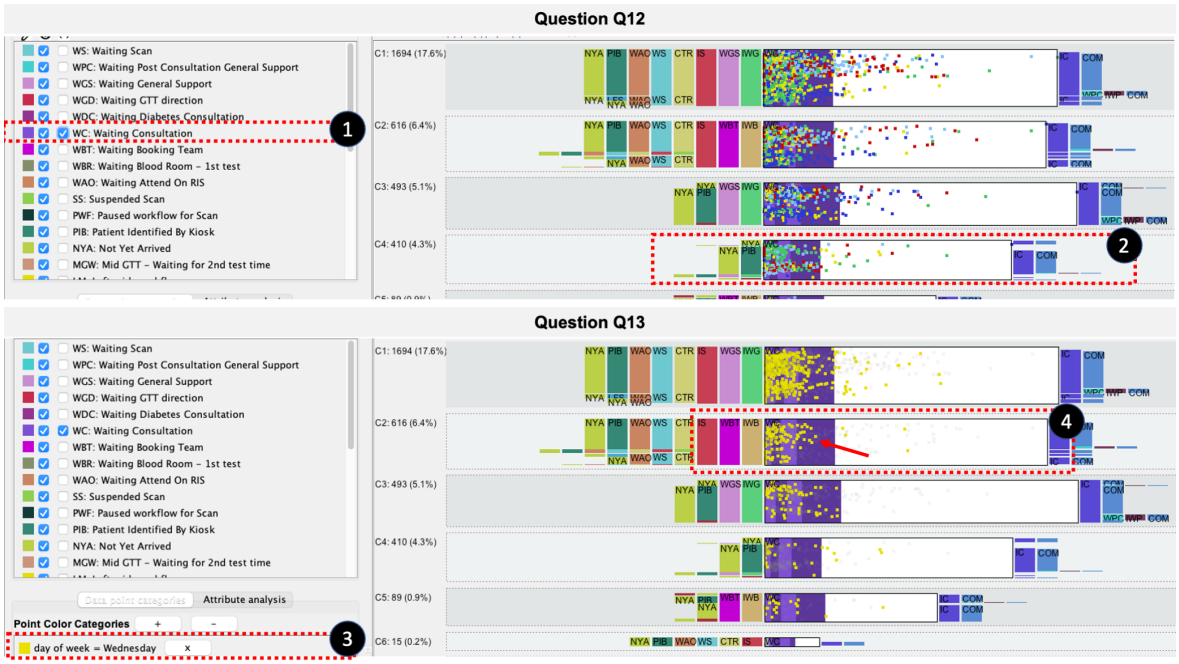


Figure 7.2: *Visualisations used for questions Q12 and Q13. 1) To answer question Q12, participants had to select the checkbox to enable EventBox in the event type Waiting Consultation. 2) The cluster with the overall shortest waiting time is cluster C4. 3) To answer question Q13, participants had to add a colour category to highlight data points occurring on a Wednesday. 4) A highest waiting time is observed on the first part of the day.*

to this question is option (d), as cluster C4 has the overall smaller width compared with clusters C1, C2, and C3. Participant P9, who provided the incorrect answer to this question, got confused because she/he thought that the duration was represented in the vertical axis of the EventBox, rather than on the horizontal axis.

Question Q13

This question resulted in an accuracy percentage of 84.6% (two incorrect answers):

Q13. A patient will visit the clinic on Wednesday. First, she will have a scan (IS), then she will be referred to the booking team (WBT and IWB), and then she will have a consultation (IC). When is it more likely that she waits more time for a consultation (WC)?

(a) First half of the day
(b) Second half of the day

As observed in Fig. 7.2-4, the correct answer is (a). The waiting time is likely to be higher on the first half of the day, as there are more data points at the top half of the fourth quarter of the highlighted EventBox. The purpose of this question was to validate the understanding of complex patterns that involve a sequential pattern and multiple variables such as duration, time of occurrence, and categorical attributes (analytic task T7). The results show that most of participants quickly familiarised with these complex patterns and were able to interpret the visualisation to answer the question correctly.

Participant P2, one of the participants who incorrectly answered this question, stated that she/he thought that the time of occurrence was represented on the horizontal axis of the EventBox, rather than the vertical axis. This type of confusion occurred as well with participant P9 in question Q12. This might imply that placing the time of occurrence on the horizontal axis would have been more intuitive for users. A future study with a larger number of participants is needed, to evaluate how switching the attributes in the axes of the EventBox improve its interpretation. Moreover, Sequen-C could include an interaction option to easily see what variable is presented in the horizontal or vertical axis of an EventBox.

7.2.2 Task completion time

The completion time can be analysed at participant or task level. A table containing the exact timings for each participant can be found in Appendix C, including minimum, median and maximum time per question.

Fig. 7.3 shows stacked bar charts comparing the total **time spent by each participant** in all the tasks. Participant P8 was the participant with the lowest cumulative completion time, with 07 minutes and 17 seconds; whereas Participant P10 took the longest to solve all questions, with a total time of 21 minutes and 36 seconds. Participants could be divided into three groups according to their completion time:

- **High proficiency:** the first group includes participants P8, P12, and P13 who finished the tasks in under 8.5 minutes.
- **Medium proficiency:** The second group includes P1, P5, P6, P7, and P9 who finished under 14.3 minutes.

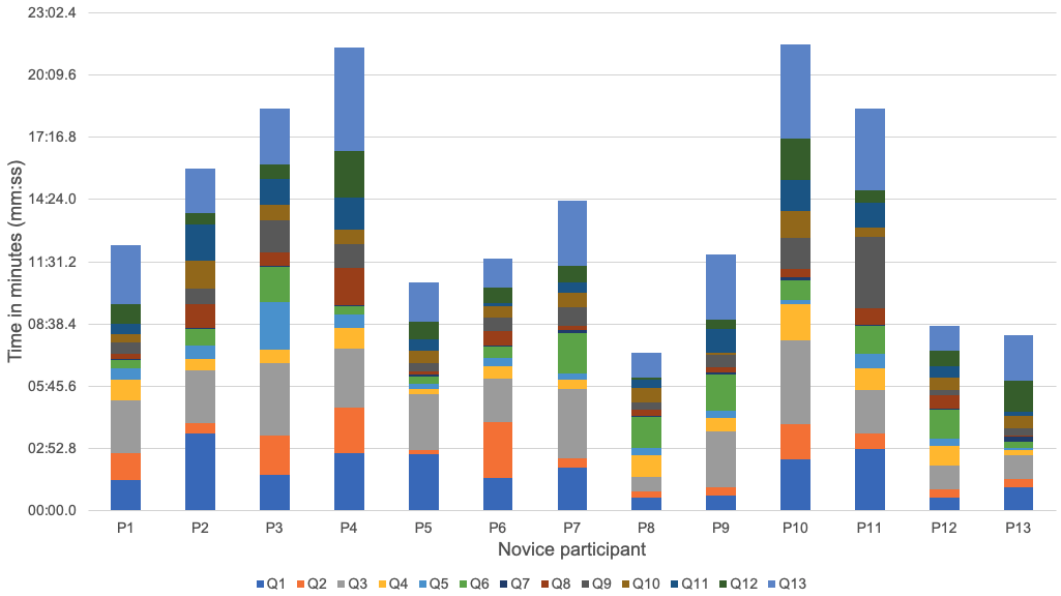


Figure 7.3: *Completion time per participant for all the questions. Each coloured series represents a question (Q1 to Q13).*

- **Low proficiency:** The last group includes P2, P3, P4, P10, and P11 with a total completion time between 15.8 minutes and 21.6 minutes.

The above classification means that not all users have familiarised with the system at the same speed, probably depending on their background or previous experience interacting with and interpreting data visualisations. It is clear that the graphical interface of Sequen-C could improve in terms of usability to allow different types of users, with any proficiency level or background, to familiarise with the system quicker and close the gap in the time of completing a task.

The average completion **time per task** was one minute five seconds. However, not all tasks had the same level of complexity or required the same amount of steps. Fig. 7.4 shows boxplots of the distribution of the completion time for each task amongst all participants. The tasks can be divided into *simple* and *advanced* according to their median completion time:

- **Simple tasks**: these tasks had a median completion time under 45 seconds and include questions Q4, Q5, and questions from Q7 to Q12.
- **Advanced tasks**: advanced tasks include questions Q1, Q2, Q3, Q6, and Q13. Table 7.2 shows the minimum, median, and maximum completion times for these tasks. Question Q2 is considered advanced as it has a high variation across participants.

Simple tasks

These tasks have in common that they required few clicks (or none), and that they required visual comparison of two items that had clearly evident differences, which might explain why

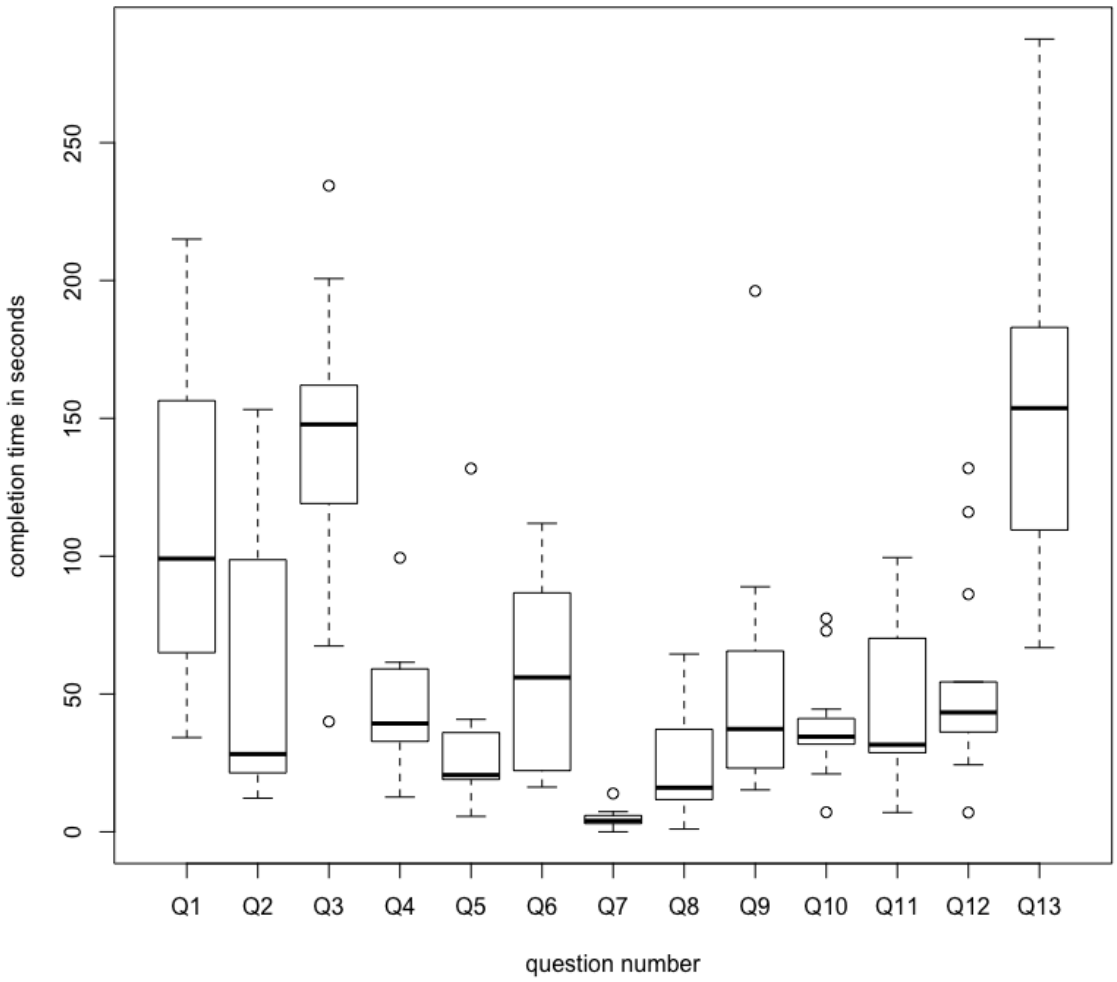


Figure 7.4: *Boxplots showing the distribution of the completion time of each task-based question.*

	average	**min**	**median**	**maximum**
Q1	01:48.5	00:34.2	01:39.1	03:35.0
Q2	00:58.3	00:12.2	00:28.2	02:33.2
Q3	02:19.3	00:40.0	02:27.8	03:54.4
Q6	00:59.1	00:16.2	00:56.0	01:51.9
Q13	02:36.3	01:06.8	02:33.7	04:47.6

Table 7.2: *Minimum, median, and maximum completion times in minutes (mm:ss) for the advanced tasks (Q1, Q2, Q3, Q6, and Q13)*

these tasks took a shorter time than the rest. For example, **question Q4** was relatively quick to answer as it only required participants to visually compare the events between two clusters and identify the event that did not occur on the second cluster. Similarly, **question Q5** required participants to select a cluster from the overview to the unique sequence view, and then identify the frequency of the sequence containing three IDC (In Diabetes Consultation) events. Moreover, **question Q12** required participants to compare the EventBox across four clusters and identify the one with the overall shortest duration. **Question Q7** was the task with the shortest time (under 7 seconds) as it only required participants to change the number of clusters to nine clusters - this indicates that the system allows participants to understand the concept of different sequence clusterings.

Questions Q10 and Q11 required the participant to study a given attribute using the bar charts in the attribute analysis panel. As per Table 7.1, all participants successfully interpreted the stacked bar charts and provided the correct answer to both questions. However, regarding question Q11:

Q11. What cluster has the highest number of patients that received a scan (IS) in room number 5?,

The researcher observed that some participants were confused when interpreting this question as they did not know if it required to add a filter, or maybe explore the EventBox of the event In Scan, or whether the scan room was an attribute or a type of event. This type of confusion would not happen where users are familiarised with the context of the dataset being explored, in this case the Antenatal Care Unit outpatient clinic.

Advanced tasks

These tasks might have taken longer, either because they required a larger number of clicks or involved more complicated visual comparisons. For example, it was observed that in **questions Q1, Q2, and Q3**, most of the task time was spent on a *visual scan* of the clusters view to find the pathway related to the question. This issue calls for an option in the system Sequen-C that allows users to find and highlight a given event or sequence within the clusters view.

Question Q6 required users to add a filter to only show sequences containing the event type Waiting Consultation. Some participants did not immediately remember the place where the "Add Filter" button was located, which increased the task time. This issue could be avoided by moving the filter controls to the main panel.

Question Q13 was the one that, overall, took the longest (see Table 7.2). As mentioned, the purpose of this question was to test the interpretation of complex multivariate patterns and the EventBox visualisation. It was observed that the more time consuming action was to understand the steps required to produce the visualisation, not the interpretation of the visualisation itself. This required the participant to first locate the cluster containing the sequential pattern IS $\rightarrow$ WBT $\rightarrow$ IWB $\rightarrow$ WC $\rightarrow$ IC, then the participant had to add a colour category to highlight data points occurring on a Wednesday (see Fig. 7.2-3). After obtaining the visualisation shown in Fig. 7.2-4, 11 out of 13 participants successfully interpreted the EventBox and provided the correct answer.

	E1	E2	P8	P13	P12	average novice
Q1	00:47.3	00:43.9	00:34.2	01:05.0	00:36.0	01:39.4
Q2	00:56.0	00:28.5	00:18.5	00:21.0	00:21.4	00:54.5
Q3	01:05.8	01:42.5	00:40.0	01:07.4	01:07.7	02:20.6
Q4	00:45.6	00:30.3	01:00.4	00:15.2	00:53.7	00:43.8
Q5	00:07.4	00:13.3	00:20.6	00:05.6	00:19.0	00:38.5
Q6	00:14.1	00:26.8	01:26.7	00:16.2	01:21.4	01:11.3
Q7	00:02.5	00:04.4	00:03.0	00:13.9	00:04.3	00:05.7
Q8	00:06.8	00:28.3	00:16.0	00:03.8	00:35.9	00:38.8
Q9	00:11.6	00:17.8	00:20.3	00:19.5	00:15.2	00:54.8
Q10	00:08.6	00:42.4	00:40.3	00:34.0	00:34.5	00:38.8
Q11	00:18.7	00:38.1	00:23.3	00:12.2	00:30.8	00:48.7
Q12	00:29.8	00:27.6	00:07.0	01:26.2	00:42.6	00:58.6
Q13	00:53.2	01:16.4	01:06.8	02:06.0	01:09.6	02:38.4
Total	06:07.3	08:00.2	07:17.0	08:06.1	08:32.1	14:11.9

Table 7.3: *Comparison of task completion time in minutes (mm:ss) for expert users (E1 and E2), the three novice participants that completed the task in the shortest time (P8, P12, P13), and the average completion time of all novice participants.*

7.2.3 Novice users versus expert users

Expert participants (E1 and E2) are considered *experts* as they have used the system Sequen-C before. Expert E1 has used the current and previous versions of system from the start of the project. Expert E2 has occasionally used the system over the last 6 months.

Figure 7.5 compares the average completion time of all the novice participants versus the two experts. It is observed that expert users significantly outperform the majority of novice participants. This makes sense as there is a natural learning curve when interacting with the system for the first time. However, as observed in Table 7.3, the completion times of novice participants P8, P12, and P13 are very close to the time of experts, meaning that the learning curve is not the same for every user.

Expert E1 completed all the tasks in 6 minutes and 7 seconds, expert E2 had a total time of 8 minutes, whilst the novice participant P8 had a total time of 7 minutes and 17 seconds. Expert E1 was the participant with the shortest time, but at the same time, participant P8 outperformed expert E2. As observed in Figure 7.6, the time difference amongst the three users (P8, E1, E2) changes in each question - this might be caused by external variables such as nervousness, attention span, or the personal interpretation of the questions. However, this suggests that certain users that interact with Sequen-C for the very first time can potentially become experts in a very short period of time.

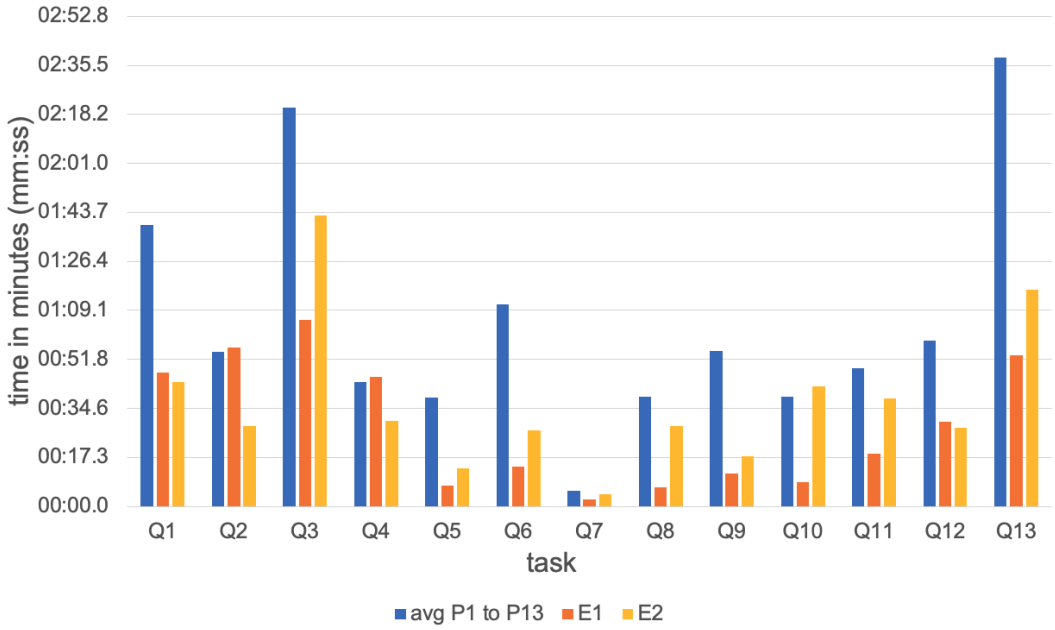


Figure 7.5: *Comparison of the average completion time of novice participants versus the two experts.*

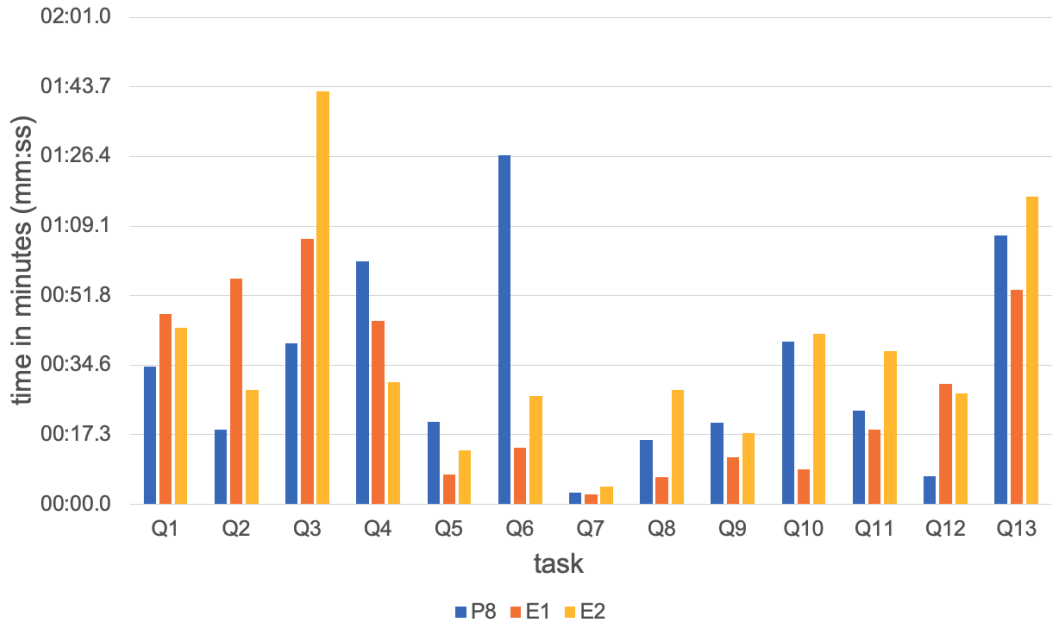


Figure 7.6: *Comparison of the novice participant with the shortest completion time (P8) and the two experts.*

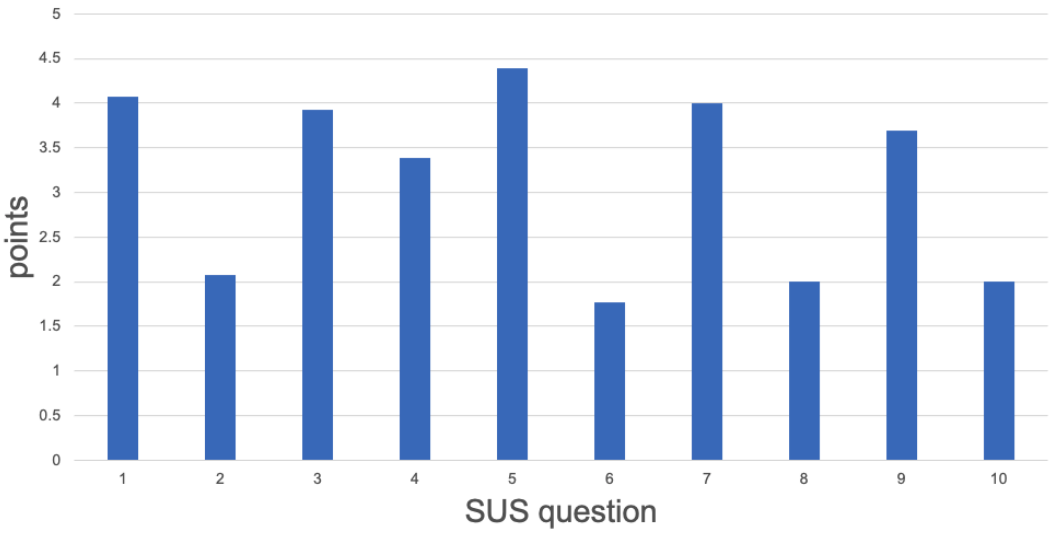


Figure 7.7: *Average number of points assigned to each question in the SUS questionnaire. A high number in an odd numbered question and a low number in an even numbered question is considered to be positive and indicates higher usability.*

7.3 Results: System Usability Scale

The thirteen novice participants provided answers to the System Usability Scale (SUS) [16] questionnaire based on their experience using the system. A link to a Google form containing this questionnaire was sent to the participants, who answered the questions after finishing the video call of the evaluation.

The SUS questionnaire is made up of ten questions that are each answered with a five point scale (see Section 7.1.2). Fig. 7.7 shows the average number of points assigned to each question, a high number in an odd numbered question and a low number in an even numbered question is considered to be *positive* and indicates higher usability. The SUS score of one participant can be computed as $score = (x - y + 20) * 2.5$, where x is the sum of the odd numbered questions and y is the sum of the even numbered questions. This score can go from 0 to 100, according to Bangor et. al [6] a score closer to 100 indicates a better perceived usability.

Fig. 7.8 shows the SUS score assigned by each of the thirteen novice participants. The average SUS score of the system Sequen-C is 72.1, which according to Bangor's scale (see Table 7.4), corresponds to a **"Good" usability**. Note that the outlier participant P12, who assigned a SUS score of 10, typed in the comments section of the SUS form: "very intuitive and easy to use. I would use it if I worked in something related", however, she/he assigned the lowest score compared to the rest of the participants. It might be that this participant misunderstood the five point scale of the answers. The score of this participant is included in the reported average score (72.1), however, if the score of this participant was removed, the average score would be 77.2.

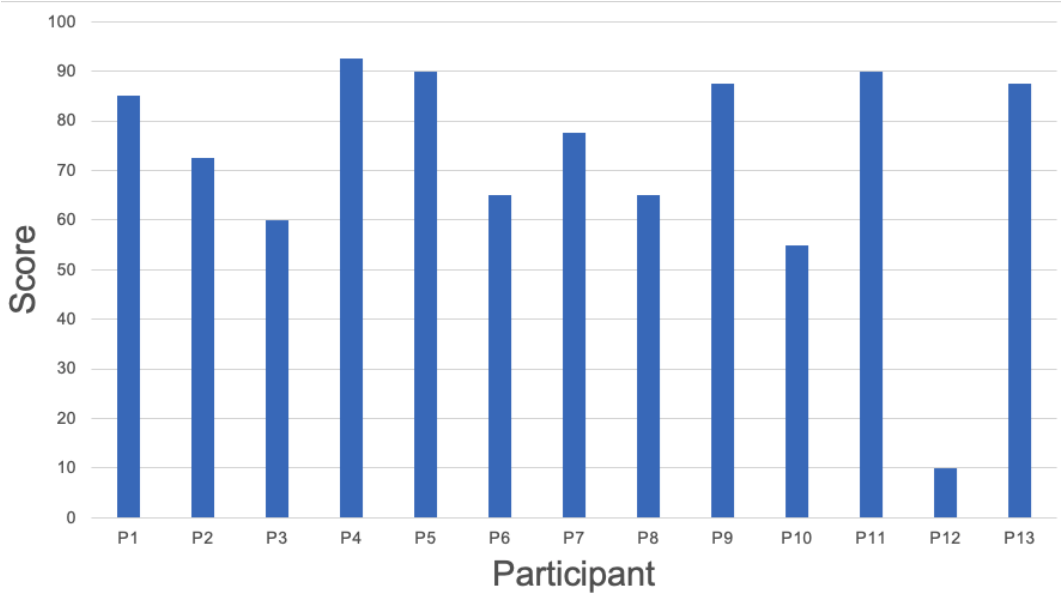


Figure 7.8: *Results of the SUS score assigned by each participant. A score closer to 100 means the perceived usability is higher.*

Adjective	**Mean SUS score**
Worst imaginable	12.5
Awful	20.3
Poor	35.7
OK	50.9
Good	71.4
Excellent	85.5
Best imaginable	90

Table 7.4: *Adjectives assigned to the SUS score. Table extracted from Bangor et. al [6]*

7.4 Results: qualitative feedback

The SUS questionnaire provided to participants through Google Form, contained an open text field for *optional feedback or comments*. Seven out of the thirteen novice participants filled this field with the following:

- "The smaller data in the clusters were harder to read and analyse."
- "Great SW for data visualization, I like the practical interface and useful possibility to get depth into the information."
- "It would be very useful if the input/search options were located in one place, instead of trying to remember which module of the system I should use to make the necessary inputs and find the information I am looking for more quickly."
- "It is very complete and easy to use."
- "I think this is a very useful system and very well built, with proper documentation and training I think users will get valuable insights from these visualizations. I found it a bit tricky to understand at first, but as the question progressed I found myself to feel more confident when using the system. Overall I think people might feel a little overwhelmed by the number of visualizations shown and with all the cool features this system has, but once they learn the basics and get more comfortable withe the system, they will find it very easy to use and to manipulate. Thank you!"
- "Very intuitive and easy to use. I would use it if I work in something related. Thanks."
- "fast to manage the tool once learned. Learning to manage is relatively easy."

Additionally, the researcher took the following notes during the evaluation, about areas of improvement that either the participant explicitly mentioned or that the researcher noticed when observing the participant's interaction with the system.

- It would be good to have a control to highlight events in the overview with a specific event type, so it does not have to be *scanned visually*.
- It would be good to be able to zoom in or magnify a selected area in the overview (e.g. to zoom in tiny data points in an EventBox).
- Colour is used to visually encode event types and attribute values in the data points of an EventBox, however, there is a limitation of how many different colours can be distinguished in the visualisation. This caused confusion for some participants, in cases where the colour of a data point is too similar to the event type containing it. A participant mentioned that rather than using colour in data points, attribute values could be encoded using shapes (e.g. +, *, o), thus avoiding the confusion caused by over-using colour.

- In some cases, it is difficult for participants to see the outlier with the highest duration, specially when the color of the data point is too clear or when the color is similar to the color of the next event.
- In some cases, it was difficult for participants to know what the available attributes are or if the question should be solved by looking at an event rather than an attribute. Other participants struggled to know if the question should be solved with a filter or by EventBox. For example Participant P5, when asked about the waiting for consultation time, she/he was looking in the bar charts rather than expanding the EventBox visualisation. This might be related to the fact that the novice participants were not data owners and therefore not familiarised with the real-world context of the data.
- Participant P7 mentioned that the font size of words in the GUI was too small to read.
- Some participants initially interpreted the gap in the cluster visualisation as two events not being sequential or consecutive (i.e. not occurring immediately after). Once the researcher clarified the concept of sequence alignment, participants understood and there was no further confusion.
- Participant P4 was momentarily confused distinguishing two event types represented with very similar colors (e.g. two shades of red).
- It was difficult for some participants to understand the meaning of the word "frequency", they understood faster if the researcher used the term "number of patients" or "number of sequences".
- Participant P8 struggled to find the button to add a filter, which is located at the upper left corner of the GUI.
- Participant P11 struggled to remember the location of the control to enable EventBox for a given event type. Similarly, another participant was clicking an event in the overview expecting it to show EventBox. It might be a better approach to visualise EventBox by clicking an event rather than using a checkbox control.

The above observations suggest improvements that could be made to the GUI, such as rearranging the location of controls and changing the type of interactions with the system (e.g. click, hover, drag and drop), to achieve a more intuitive interaction.

7.5 Summary

This chapter evaluated the adoption of the system by 13 novice users and compared it with that of two expert users. Firstly, the interpretability of EventBox and the proposed cluster representation, and the ability of the system to allow users to achieve the analytic tasks, were evaluated through a set of tasks - using completion time and accuracy as the metric. Secondly, the usability of the system Sequen-C was evaluated using the System Usability Scale (SUS) questionnaire [16].

The results indicate that the system Sequen-C can allow novice users to quickly familiarise with the proposed visualisations and successfully obtain insights from the data according to

the analytic tasks. The comparison of novice versus expert users indicate that certain users can become *experts* of the system in a relatively short time. However, there is a considerable gap in the task completion time between participants; the analysis of the interaction of novice users with the system, allowed the identification of areas of improvement in the graphical interface of Sequen-C. The results of the SUS questionnaire indicate that Sequen-C has a *good* usability level.

Chapter 8

Conclusions

8.1 Conclusions

This thesis has made **four main contributions**: first, a technique to build and explore a multilevel and multivariate overview of temporal event sequences via hierarchical aggregation, a novel cluster representation *Align-Score-Simplify*, and the novel *EventBox* visualisation; second, Sequen-C, a visual analytics system that implements the proposed technique; third, four case studies using four real-world datasets in the healthcare domain; and fourth, an evaluation of the technique in terms of user performance and usability.

A technique to build and explore a *multilevel and multivariate* overview of temporal event sequences was presented in this thesis. The proposed overview is a visual summary of event sequences and multivariate data attributes that allows users to explore different sequence clusterings via hierarchical aggregation, where sequence clusters are visually represented by their most relevant events using the procedure *Align-Score-Simplify*. The ***multilevel*** component refers to the ability of transforming the overview from a coarse to fine level-of-detail (Chapter 4); whilst the ***multivariate*** component means that, in addition to sequential patterns, temporal and categorical data attributes are visually encoded in the overview via EventBox (Chapter 5). The methodology was designed to allow the exploration of sequential patterns and the identification of trends involving multivariate attributes across and within sequential patterns, while also allowing the identification of anomalous scenarios such as infrequent patterns or individuals with outlier values.

The technique was implemented into a visual analytics system called Sequence Cluster Explorer (Sequen-C), which allows detail-on-demand exploration through three coordinated views and the analysis of data attributes at cluster, sequence, or individual record level. Four case studies were performed in collaboration with domain experts using real-world datasets: the MIMIC-III and Antenatal Care Unit (ANC) case studies evaluated the multilevel component of the overview (Chapter 4), the Rheumatology case study presented examples of findings obtained with the novel visualisation EventBox (Chapter 5), and lastly, the CUREd case study demonstrated the full technique and presented various findings (Chapter 6). The case studies show the potential of Sequen-C to support domain experts in their decision making process. Furthermore, an evaluation with fifteen participants shows how Sequen-C can allow novice users to quickly familiarise with the proposed visualisations and obtain insights from the data according to the analytic tasks (Chapter 7).

Chapter 2 presented a review of the current visual analytic techniques for event data. Despite the great advances, two main gaps were identified in the current literature. **First limitation:** current visual overviews have a static level-of-detail, users are limited to a single overview as the starting point rather than being able to seamlessly transform the level-of-detail and explore alternative optimal overviews. Further, it was identified that there is no existing technique to explore different sequence clusterings that at the same time provides a clear interpretable representation of each cluster. The most similar techniques to Sequen-C are Vasabi [72] and Sequence Synopsis [19]. However the cluster representation in Vasabi does not allow users to easily derive the original sequences, and the overall level-of-detail of the overview presented by Sequence Synopsis is static as it does not allow one to change the number of clusters or the level of simplification applied to cluster representations. **Second limitation:** multivariate event attributes such as duration, time of occurrence, and categorical attributes (e.g. age, gender, country) are usually not included in the overview and can only be accessed through secondary views, or if attributes are included in the overview, they are oversimplified using average values. This thesis approaches the first and second limitations by proposing a multilevel (to address the first limitation) and multivariate (to address the second limitation) overview of event sequences.

Chapter 3 presented the analytic tasks that guided the development of the methodology and system proposed in this thesis.

Chapter 4 presented the multilevel component of the proposed visual overview. The multilevel overview presents a given number of sequence clusters k retrieved from a hierarchical aggregation of the input event sequences, built using a bottom-up approach [1]. A data representation of the sequences in each cluster is obtained using the novel procedure Align-Score-Simplify. First, Multiple Sequence Alignment (MSA) [30] is used to align the sequences in each cluster (Align). Secondly, the information score of each column in the alignment matrix is computed (Score). And thirdly, the alignment is simplified by merging consecutive columns with a low information score (Simplify), where the merged columns are specially encoded to provide a summary of the simplified events. The cluster representation allows users to identify permutations in the order of events and it highlights the most relevant events in the sequences of a cluster. By default the overview presents the best clustering according to the Average Silhouette Width metric (ASW) [47], but users can interactively change and explore any other overviews with any number of clusters. To facilitate that exploration, a subset of alternative optimal values of k is obtained from the ASW curve.

The overview can be transformed from coarse to fine *vertical* or *horizontal level-of-detail*. The vertical level-of-detail of the overview changes along with the number of clusters k, whilst the horizontal level-of-detail refers to the level of summarisation applied to each cluster representation. Two case studies with two real-world datasets, MIMIC-III and ANC, were presented in this chapter. The case study of the publicly available MIMIC-III dataset [45] was made in collaboration with a cardiologist, where a subset of patients with a primary or secondary diagnosis of Atrial Fibrillation was studied. Sequen-C clustered patients according to their care unit and prescription history, and made it possible to identify clusters with overall higher length of stay. The ANC case study used real-world data from an Antenatal clinic and was made in collaboration with two analysts at Sheffield Teaching Hospitals. The analysis of clusters in the dataset resulted in a proposed re design of the existing workflows in the clinic.

Chapter 5 presented the multivariate component of the proposed visual overview. Building on the work presented in Chapter 4, the multilevel overview, which already presents an overview of sequential patterns, integrates multivariate attributes for selected event types via EventBox. Resulting in a multilevel and multivariate overview able to represent complex patterns related to multiple variables in a single overview. An EventBox is a novel visual encoding, inspired by Box plots and Scatter plots, that aggregates the duration, time of occurrence, and categorical attributes (e.g. age, gender, country) of a set of event occurrences of the same type, and allows the identification of trends and outliers with respect to these attributes. The chapter proposes different levels of detail for an EventBox to reduce visual clutter and focus the analysis on specific areas (e.g. outliers). A set of findings was obtained for a real-world dataset from a Rheumatology outpatient clinic, which shows how Sequen-C allows the comparison of the distribution of attributes within and across sequential patterns using a single overview, supporting findings such as "the duration of the event In Consultation is considerably longer when the event Height and Weight does not occur in that sequence".

Chapter 6 demonstrated the full technique (multilevel and multivariate component) using the CUREd dataset [48], which contains timestamped events and demographic data related to telephone calls made to the emergency service (calls to 999 or 111), throughout Yorkshire and the Humber region in the UK. This case study was made in collaboration with three members of the Centre for Urgent and Emergency Care Research (CURE). The case study has demonstrated the ability of the technique to support users in obtaining a set of distinct clusters that best describe the data, and at the same time describe and compare clusters by their multivariate attributes. This means, through exploratory analysis, users can determine the most appropriate clustering for patients - for example, a cluster of patients that attended the emergency department versus a cluster of patients that did not. Then, the analysis of multivariate attributes can allow the characterisation of clusters and to obtain findings such as "72.3% of calls coming from area 8C3 are usually closed without any ambulance service being provided" or "the longest calls in Cluster 1 usually happen around the afternoon". These findings demonstrate the ability of the technique to obtain relevant insights from the data, opening up the possibility of supporting decisions towards improving services in real-world scenarios.

Chapter 7 evaluates the technique with fifteen participants, thirteen novice and two expert users, in terms of user performance and usability. User performance is evaluated through a set of tasks, where completion time and accuracy are measured. Usability was measured with the System Usability Scale [16] questionnaire. The results indicate that the system Sequen-C allows novice users to quickly familiarise with the proposed visualisations and successfully obtain insights from the data according to the analytic tasks. The comparison of novice versus expert users indicate that three out of the thirteen novice users outperformed or got close to the total time of an expert user. On the other hand, the analysis of time and accuracy of certain tasks, allowed the identification of areas of improvement in the graphical interface of Sequen-C. The results of the SUS questionnaire indicate that Sequen-C has a *good* usability level.

8.2 Discussion and future work

Feedback from domain experts

Experts that collaborated in the case studies (MIMIC-III, Antenatal Care Unit, Rheumatology, and CUREd) found it very interesting to be able to explore sequence clusterings and instantly inspect further details of interesting patterns.

According to one of the domain experts from the Rheumatology and the Antenatal case studies, the type of findings obtained in this thesis would traditionally take several meetings between analysts and policymakers. The analysts would require time to go back and answer certain questions based on the data; with Sequen-C, some of those questions can be answered instantly. Moreover, a domain expert from the CUREd case study mentioned that this type of analysis is really useful as it provides "insight into which calls are likely to need a hospital transfer and which may benefit from a different response" and that "knowing these patterns might help assist decision making for call handlers". The domain expert from the MIMIC-III case study stated that "such observations are helpful in informing healthcare planning" and that outpatient dosing could be targeted to specific clusters with the purpose to "reduce length of stay and free up valuable hospital beds".

Generalisability

The proposed technique is intended to be used with any dataset from any domain as long as it contains a collection of timestamped events. Event sequences, time of occurrence, duration, and data attributes are terms that can be translated to any domain where timestamped events are collected (e.g. in the form of event logs). The case studies demonstrate the generalisability of this technique, even though the four case studies are in the healthcare domain, the datasets have different dimensions, structure, and variability.

Scalability and Multiple Sequence Alignment

The case studies have shown that the technique can handle variability caused by relatively high number of event types (e.g. 448 event types in the MIMIC-III case study) and volume caused by a relative high number of sequences (e.g. 21,805 individual sequences and 962 unique sequences in the CUREd case study). However, the implementation of the Multiple Sequence Alignment (MSA) algorithm [30] might perform too slowly when aligning thousands of unique sequences or thousands of event types. Moreover, the scalability of the cluster representation should be tested with this type of really high volume datasets, to ensure that the resulting alignment is still interpretable. Another limitation is that the cluster representation (Align-Score-Simplify) can produce different results depending on the chosen values of substitution score and gap penalty (see section. 2.4.2), so these costs might need to be adjusted for certain datasets. The system Sequen-C could include controls to interactively adjust the substitution and gap costs. The alignment of very different sequences, with not many events in common, is challenging - especially when building the alignment of the full dataset (overview with $k = 1$). MSA could benefit from an event type categorisation (e.g. care units, prescriptions), so that events can be aligned based on their meaning rather than the name of the event type.

Optimal overviews of temporal event sequences

One of the purposes of the present technique is to provide a *flexible* visual overview of temporal event sequences, whose overall level-of-detail can be easily transformed and that offers more than one optimal overview. Users can change the number of clusters k shown in such overview, and also transform the level of summarisation applied to clusters through the information score threshold I_τ. The technique already obtains a set of optimal values of k that represent good overviews. However, future work is needed to determine the optimal value of the information score I_τ, either at individual cluster level or for the whole dataset.

Cluster representation design

In the proposed cluster representation (Align-Score-Simplify), the event types in the simplified sub-sequences are explicitly encoded, which allows users to understand the variability of the cluster and in some cases derive the original sequences. However, with an increased number of event types, the representation of merged sub-sequences may become difficult to interpret due to visual clutter. Future work is needed to provide alternative designs to explicitly encode merged event types that increase information content while handling visual clutter.

EventBox visualisation

A line of future work is to encode other numeric and categorical attributes in the horizontal and vertical axis of the EventBox, besides duration and time of occurrence. As mentioned in Chapter 5, similar to a scatter plot visualisation, each data point is located in an EventBox according to its duration (horizontal axis) and time of occurrence (vertical axis). Duration and time of occurrence are numerical values. A challenge would be to create an intuitive design that allows to position data points according to categorical attributes, especially with large numbers of categories. Moreover, the EventBox visualisation may present scalability limitations when visualising a high volume of records, due to the overlap in the position of the data points. Another line of future research is to propose designs that further summarise or group the data points presented in an EventBox.

Automatic findings

A flexible overview that allows users to uncover hidden insights was presented in this thesis. However, the current technique still depends, at some level, on the knowledge and hypothesis of domain experts to drill down on interesting items of the overview or ask the right questions. It is still a challenge to provide a default overview with the *ideal* level-of-detail that shows exactly the insights that the user is looking for, hence the importance of providing intuitive interaction controls for exploratory data analysis. A line of future research would be to integrate machine learning algorithms in the visualisation that can learn from the user interaction, types of queries, and types of findings obtained; then use such knowledge to produce better visual overviews that highlight similar findings to the ones previously discovered by the user.

Explainable clustering

Currently, the occurrence of events is taken to measure similarity between sequences. A line of future research is to consider other data attributes in the clustering algorithm, for example the duration and time of occurrence of the events, as well as data attributes such as patient personal data, in the case of clinical records. This means that rather than using a string edit distance such as Levenshtein or q-grams, the distance metric should integrate other weighted categorical and numerical features. Another line of future research is to adapt the system Sequen-C to be *plugged in* to other clustering algorithms for event sequences, besides hierarchical clustering. The visualisations presented in this thesis could allow data scientists to interpret clusters and understand which features are more prevalent in each cluster. For example, preliminary experiments on a side project, where different sequence clusterings were obtained using auto-encoders, k-means, and Hidden Markov Models, showed that Sequen-C can provide insights on the criteria used by the clustering algorithms to form groups.

Closing remark

The four main contributions made by this thesis aim to advance the science of visual analytics of temporal event sequences, specially in the topics of sequence clustering and visual overviews. Ideally, the system Sequen-C should become publicly available, either commercially or as an open source project. So more users are able to explore datasets from different domains including healthcare, supply chain, public transport, amongst many others.

Bibliography

[1] Aggarwal, C. C. and Reddy, C. K. [2014], *Data clustering: algorithms and applications*, CRC Press.

[2] Aherne, J. and Whelton, J. [2010], *Applying lean in healthcare: a collection of international case studies*, CRC Press.

[3] Aigner, W., Federico, P., Gschwandtner, T., Miksch, S. and Rind, A. [2012], Challenges of time-oriented data in visual analytics for healthcare, *in* 'IEEE VisWeek Workshop on Visual Analytics in Healthcare', Vol. 4, IEEE.

[4] Ayres, J., Flannick, J., Gehrke, J. and Yiu, T. [2002], Sequential pattern mining using a bitmap representation, *in* 'Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining', pp. 429–435.

[5] Backhouse, A. and Ogunlayi, F. [2020], 'Quality improvement into practice', *British Medical Journal Publishing Group* **368**.

[6] Bangor, A., Kortum, P. and Miller, J. [2009], 'Determining what individual sus scores mean: Adding an adjective rating scale', *Journal of usability studies* **4**(3), 114–123.

[7] Bardsley, M. [2016], 'Understanding analytical capability in health care', *Do we have more data than insight?. London: The Health Foundation* .
URL: *https://www.health.org.uk/publications/understanding-analytical-capability-in-health-care*

[8] Bardsley, M., Steventon, A. and Fothergill, G. [2019], 'Untapped potential: Investing in health and care data analytics', *The Health Foundation* .
URL: *https://www.health.org.uk/publications/reports/untapped-potential-investing-in-health-and-care-data-analytics*

[9] Barnett, V. and Lewis, T. [1994], *Outliers in statistical data.*, 3rd edition edn, John Wiley & Sons Ltd.

[10] Bellet, A., Habrard, A. and Sebban, M. [2015], 'Metric learning', *Synthesis Lectures on Artificial Intelligence and Machine Learning* **9**(1), 1–151.

[11] Bolyen, E., Rideout, J. R., Chase, J., Pitman, T. A., Shiffer, A., Mercurio, W., Dillon, M. R. and Caporaso, J. G. [2018], 'An introduction to applied bioinformatics: a free, open, and interactive text.', *The Journal of open source education* **1**(5).

[12] Borland, D., West, V. L. and Hammond, W. E. [2016], Multivariate visualization of longitudinal clinical data, *in* 'Proceedings 2016 IEEE VIS Workshop on Visual Analytics in Healthcare, Chicago, IL'.

[13] Bose, R. J. C. and van der Aalst, W. [2010], Trace alignment in process mining: opportunities for process diagnostics, *in* 'International Conference on Business Process Management', Springer, pp. 227–242.

[14] Bose, R. J. C. and Van der Aalst, W. M. [2009], Context aware trace clustering: Towards improving process mining results, *in* 'proceedings of the 2009 SIAM International Conference on Data Mining', SIAM, pp. 401–412.

[15] Bouarfa, L. and Dankelman, J. [2012], 'Workflow mining and outlier detection from clinical activity logs', *Journal of biomedical informatics* **45**(6), 1185–1190.

[16] Brooke, J. [1996], 'Sus: a "quick and dirty'usability', *Usability evaluation in industry* **189**.

[17] Cadez, I., Heckerman, D., Meek, C., Smyth, P. and White, S. [2003], 'Model-based clustering and visualization of navigation patterns on a web site', *Data mining and knowledge discovery* **7**(4), 399–424.

[18] Cappers, B. C. and van Wijk, J. J. [2018], 'Exploring multivariate event sequences using rules, aggregations, and selections', *IEEE Transactions on Visualization and Computer Graphics* (1), 532–541.

[19] Chen, Y., Xu, P. and Ren, L. [2018], 'Sequence synopsis: Optimize visual summary of temporal event data', *IEEE Transactions on Visualization and Computer Graphics* **24**(1), 45–55.

[20] Courtlandt, C. D., Noonan, L. and Feld, L. G. [2009], 'Model for improvement-part 1: A framework for health care quality', *Pediatric Clinics of North America* **56**(4), 757–778.

[21] Dang, T. N., Wilkinson, L. and Anand, A. [2010], 'Stacking graphic elements to avoid over-plotting', *IEEE Transactions on Visualization and Computer Graphics* **16**(6), 1044–1052.

[22] Dekking, F. M., Kraaikamp, C., Lopuhaä, H. P. and Meester, L. E. [2005], *A Modern Introduction to Probability and Statistics: Understanding why and how*, Springer Science & Business Media.

[23] Demirkan, H., Spohrer, J. C. and Welser, J. J. [2016], 'Digital innovation and strategic transformation', *IT Professional* **18**(6), 14–18.

[24] Di Bartolomeo, S., Zhang, Y., Sheng, F. and Dunne, C. [2020], 'Sequence braiding: Visual overviews of temporal event sequences and attributes', *IEEE Transactions on Visualization and Computer Graphics* **27**(2), 1353–1363.

[25] Du, F., Shneiderman, B., Plaisant, C., Malik, S. and Perer, A. [2016], 'Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus', *IEEE Transactions on Visualization and Computer Graphics* **23**(6), 1636–1649.

[26] Edgar, R. C. and Batzoglou, S. [2006], 'Multiple sequence alignment', *Current opinion in structural biology* **16**(3), 368–373.

[27] Elmqvist, N. and Fekete, J.-D. [2009], 'Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines', *IEEE Transactions on Visualization and Computer Graphics* **16**(3), 439–454.

[28] England, N. [2019], 'The NHS long term plan', *Department of Health and Social Care* .

[29] Fails, J. A., Karlson, A., Shahamat, L. and Shneiderman, B. [2006], A visual interface for multivariate temporal data: Finding patterns of events across multiple histories, *in* '2006 IEEE Symposium On Visual Analytics Science And Technology', IEEE, pp. 167–174.

[30] Feng, D.-F. and Doolittle, R. F. [1987], 'Progressive sequence alignment as a prerequisite to correct phylogenetic trees', *Journal of molecular evolution* **25**(4), 351–360.

[31] Frigge, M., Hoaglin, D. C. and Iglewicz, B. [1989], 'Some implementations of the boxplot', *The American Statistician* **43**(1), 50–54.

[32] Gay, D., Guigourès, R., Boullé, M. and Clérot, F. [2015], Tess: temporal event sequence summarization, *in* '2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)', IEEE, pp. 1–10.

[33] Goodstadt, L. and Ponting, C. P. [2001], 'CHROMA: consensus-based colouring of multiple alignments for publication', *Bioinformatics* **17**(9), 845–846.

[34] Gotz, D. and Borland, D. [2016], 'Data-driven healthcare: challenges and opportunities for interactive visualization', *IEEE Computer Graphics and Applications* **36**(3), 90–96.

[35] Gotz, D. and Stavropoulos, H. [2014], 'DecisionFlow: Visual analytics for high-dimensional temporal event sequence data', *IEEE Transactions on Visualization and Computer Graphics* **20**(12), 1783–1792.

[36] Gotz, D., Sun, J., Cao, N. and Ebadollahi, S. [2011], Visual cluster analysis in support of clinical decision intelligence, *in* 'AMIA Annual Symposium Proceedings', Vol. 2011, American Medical Informatics Association, p. 481.

[37] Gotz, D., Zhang, J., Wang, W., Shrestha, J. and Borland, D. [2019], 'Visual analysis of high-dimensional event sequence data via dynamic hierarchical aggregation', *IEEE Transactions on Visualization and Computer Graphics* **26**(1), 440–450.

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

[39] Guo, S., Jin, Z., Gotz, D., Du, F., Zha, H. and Cao, N. [2018], 'Visual progression analysis of event sequence data', *IEEE Transactions on Visualization and Computer Graphics* **25**(1), 417–426.

[40] Guo, S., Xu, K., Zhao, R., Gotz, D., Zha, H. and Cao, N. [2018], 'Eventthread: Visual summarization and stage analysis of event sequence data', *IEEE Transactions on Visualization and Computer Graphics* **24**(1), 56–65.

[41] Guo, Y., Guo, S., Jin, Z., Kaul, S., Gotz, D. and Cao, N. [2020], 'Survey on visual analysis of event sequence data', *arXiv preprint arXiv:2006.14291* .

[42] Ham, C., Berwick, D. and Dixon, J. [2016], 'Improving quality in the english NHS', *London: The King's Fund* .

[43] Huesch, M. D. and Mosher, T. J. [2017], 'Using it or losing it? the case for data scientists inside health care', *NEJM Catalyst* **3**(3).

[44] James, G., Witten, D., Hastie, T. and Tibshirani, R. [2013], *An introduction to statistical learning*, Vol. 112, Springer.

[45] Johnson, A. E., Pollard, T. J., Shen, L. and other [2016], 'MIMIC-III, a freely accessible critical care database', *Scientific data* **3**(1), 1–9.

[46] Jones, B., Vaux, E. and Olsson-Brown, A. [2019], 'How to get started in quality improvement', *BMJ* **364**, k5408.

[47] Kaufman, L. and Rousseeuw, P. J. [2009], *Finding groups in data: an introduction to cluster analysis*, Vol. 344, John Wiley & Sons.

[48] Kuczawski, M., Stone, T. and Mason, S. [2019], CUREd: Creating a research database to improve urgent and emergency care system research, *in* 'EUSEM Abstracts. Prague.', p. 512.

[49] Kwon, B. C., Verma, J. and Perer, A. [2016], Peekquence: Visual analytics for event sequence data, *in* 'ACM SIGKDD 2016 Workshop on Interactive Data Exploration and Analytics', Vol. 1.

[50] Lam, H., Bertini, E., Isenberg, P., Plaisant, C. and Carpendale, S. [2011], 'Empirical studies in information visualization: Seven scenarios', *IEEE Transactions on Visualization and Computer Graphics* **18**(9), 1520–1536.

[51] Lee, C. [2003], 'Generating consensus sequences from partial order multiple sequence alignment graphs', *Bioinformatics* **19**(8), 999–1008.

[52] Levenshtein, V. I. [1966], Binary codes capable of correcting deletions, insertions, and reversals, *in* 'Soviet physics doklady', Vol. 10, pp. 707–710.

[53] Lewis, J. R. [2018], 'The system usability scale: past, present, and future', *International Journal of Human–Computer Interaction* **34**(7), 577–590.

[54] Liberatore, M. J. [2013], 'Six sigma in healthcare delivery', *International journal of health care quality assurance* .

[55] Limb, C., Fowler, A., Gundogan, B., Koshy, K. and Agha, R. [2017], 'How to conduct a clinical audit and quality improvement project', *International journal of surgery. Oncology* **2**(6), e24.

[56] Liu, Z., Kerr, B., Dontcheva, M. et al. [2017], CoreFlow: Extracting and visualizing branching patterns from event sequences, *in* 'Computer Graphics Forum', Vol. 36, pp. 527–538.

[57] Liu, Z., Stasko, J. and Sullivan, T. [2009], 'Selltrend: Inter-attribute visual analysis of temporal transaction data', *IEEE Transactions on Visualization and Computer Graphics* **15**(6), 1025–1032.

[58] Liu, Z., Wang, Y., Dontcheva, M., Hoffman, M., Walker, S. and Wilson, A. [2017], 'Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths', *IEEE Transactions on Visualization and Computer Graphics* **23**(1), 321–330.

[59] Loorak, M. H., Perin, C., Kamal, N., Hill, M. and Carpendale, S. [2016], 'Timespan: Using visualization to explore temporal multi-dimensional data of stroke patients', *IEEE Transactions on Visualization and Computer Graphics* **22**(1), 409–418.

[60] Magallanes, J., Stone, T., Morris, P. D., Mason, S., Wood, S. and Villa-Uriol, M.-C. [2021], 'Sequen-C: A Multilevel Overview of Temporal Event Sequences', *IEEE Transactions on Visualization and Computer Graphics* . arXiv preprint: `https://arxiv.org/abs/2108.03043`, In press.

[61] Magallanes, J., van Gemeren, L., Wood, S. and Villa-Uriol, M.-C. [2019], Analyzing time attributes in temporal event sequences, *in* '2019 IEEE Visualization Conference (VIS)', IEEE, pp. 1–5.

[62] Makanju, A., Brooks, S., Zincir-Heywood, A. N. and Milios, E. E. [2008], Logview: Visualizing event log clusters, *in* '2008 Sixth Annual Conference on Privacy, Security and Trust', IEEE, pp. 99–108.

[63] Malik, S., Du, F., Monroe, M., Onukwugha, E., Plaisant, C. and Shneiderman, B. [2015], Cohort comparison of event sequences with balanced integration of visual analytics and statistics, *in* 'Proceedings of the 20th International Conference on Intelligent User Interfaces', pp. 38–49.

[64] Margetts, H. and Dorobantu, C. [2019], 'Rethink government with ai', *Nature Publishing Group* .
URL: *https://doi.org/10.1038/d41586-019-01099-5*

[65] Monroe, M., Lan, R., Lee, H., Plaisant, C. and Shneiderman, B. [2013], 'Temporal event sequence simplification', *IEEE Transactions on Visualization and Computer Graphics* **19**(12), 2227–2236.

[66] Monroe, M., Lan, R., Morales del Olmo, J., Shneiderman, B., Plaisant, C. and Millstein, J. [2013], The challenges of specifying intervals and absences in temporal queries: A graphical language approach, *in* 'Proceedings of the SIGCHI Conference on Human Factors in Computing Systems', pp. 2349–2358.

[67] Monroe, M., Wongsuphasawat, K., Plaisant, C., Shneiderman, B., Millstein, J. and Gold, S. [2012], 'Exploring point and interval event patterns: Display methods and interactive visual query', *University of Maryland Technical Report* .

[68] Munzner, T. [2014], *Visualization Analysis and Design*, CRC Press.

[69] Murphy, K. P. [2012], *Machine learning: a probabilistic perspective*, MIT press.

[70] Navarro, G. [2001], 'A guided tour to approximate string matching', *ACM computing surveys (CSUR)* **33**(1), 31–88.

[71] Needleman, S. B. and Wunsch, C. D. [1970], 'A general method applicable to the search for similarities in the amino acid sequence of two proteins', *Journal of Molecular Biology* **48**(3), 443–453.

[72] Nguyen, P. H., Henkin, R., Chen, S., Andrienko, N., Andrienko, G., Thonnard, O. and Turkay, C. [2019], 'Vasabi: Hierarchical user profiles for interactive visual user behaviour analytics', *IEEE Transactions on Visualization and Computer Graphics* **26**(1), 77–86.

[73] NHS Digital [2021], 'Attendance disposal code list'.
URL: *https://digital.nhs.uk/data-and-information/data-tools-and-services/data-services/hospital-episode-statistics/hospital-episode-statistics-data-dictionary#download-hes-data-dictionaries*

[74] NHS England [2021], 'Quality, service improvement and redesign (qsir) tools'.
URL: *https://www.england.nhs.uk/quality-service-improvement-and-redesign-qsir-tools/*

[75] ONS UK [2021], '2011 area classification code'.
URL: *https://www.ons.gov.uk/methodology/geography/geographicalproducts/areaclassifications/2011areaclassifications*

[76] Perer, A. and Wang, F. [2014], Frequence: interactive mining and visualization of temporal frequent event sequences, *in* 'Proceedings of the 19th international conference on Intelligent User Interfaces', pp. 153–162.

[77] Polack, P. J., Chen, S.-T., Kahng, M., Sharmin, M. and Chau, D. H. [2015], Timestitch: Interactive multi-focus cohort discovery and comparison, *in* '2015 IEEE Conference on Visual Analytics Science and Technology (VAST)', IEEE, pp. 209–210.

[78] Raidou, R. G., Gröller, M. E. and Eisemann, M. [2019], 'Relaxing dense scatter plots with pixel-based mappings', *IEEE Transactions on Visualization and Computer Graphics* **25**(6), 2205–2216.

[79] Riehmann, P., Hanfler, M. and Froehlich, B. [2005], Interactive sankey diagrams, *in* 'IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.', pp. 233–240.

[80] Rostamzadeh, N., Abdullah, S. S. and Sedig, K. [2021], Visual analytics for electronic health records: A review, *in* 'Informatics', Vol. 8, Multidisciplinary Digital Publishing Institute, p. 12.

[81] Rousseeuw, P. J. [1987], 'Silhouettes: a graphical aid to the interpretation and validation of cluster analysis', *Journal of computational and applied mathematics* **20**, 53–65.

[82] Sadowski, J. [2019], 'When data is capital: Datafication, accumulation, and extraction', *Big Data & Society* **6**(1), 2053951718820549.

[83] Schmidt, M. [2008], 'The Sankey diagram in energy and material flow management: part ii: methodology and current applications', *Journal of industrial ecology* **12**(2), 173–185.

[84] Shneiderman, B. [2003], The eyes have it: A task by data type taxonomy for information visualizations, *in* 'The Craft of Information Visualization', Elsevier, pp. 364–371.

[85] Shneiderman, B., Plaisant, C. and Hesse, B. W. [2013], 'Improving healthcare with interactive visualization', *Computer* **46**(5), 58–66.

[86] Song, M., Günther, C. W. and Van der Aalst, W. M. [2008], Trace clustering in process mining, *in* 'International conference on business process management', Springer, pp. 109–120.

[87] Stragier, J., Vandewiele, G., Coppens, P., Ongenae, F., Van den Broeck, W., De Turck, F. and De Marez, L. [2019], 'Data mining in the development of mobile health apps: Assessing in-app navigation through Markov chain analysis', *Journal of medical Internet research* **21**(6), e11934.

[88] Taylor, M. J., McNicholas, C., Nicolay, C., Darzi, A., Bell, D. and Reed, J. E. [2014], 'Systematic review of the application of the plan–do–study–act method to improve quality in healthcare', *BMJ quality & safety* **23**(4), 290–298.

[89] The scikit-bio development team [2020], 'scikit-bio: A bioinformatics library for data scientists, students, and developers'.
URL: *http://scikit-bio.org/docs/0.5.1/alignment.html*

[90] Trümper, J., Telea, A. and Döllner, J. [2012], Viewfusion: Correlating structure and activity views for execution traces., *in* 'TPCG', Citeseer, pp. 45–52.

[91] Tukey, J. W. [1977], *Exploratory Data Analysis*, Addison-Wesley.

[92] Ukkonen, E. [1992], 'Approximate string-matching with q-grams and maximal matches', *Theoretical computer science* **92**(1), 191–211.

[93] UOS CUREd [2021], 'Uos cured data dictionaries version 1'.
URL: *https://docs.google.com/spreadsheets/d/18M3ZulPeDs0sHo2ieGElQ1MrNpwtUdeb23ajSH4qVLM/edit?usp=sharing*

[94] Van der Loo, M. P. [2014], 'The stringdist package for approximate string matching', *The R Journal* **6**(1), 111–122.

[95] Vrotsou, K., Johansson, J. and Cooper, M. [2009], 'Activitree: Interactive visual exploration of sequences in event-based data using graph similarity', *IEEE Transactions on Visualization and Computer Graphics* **15**(6), 945–952.

[96] Wang, G., Zhang, X., Tang, S., Zheng, H. and Zhao, B. Y. [2016], Unsupervised clickstream clustering for user behavior analysis, *in* 'Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems', ACM, pp. 225–236.

[97] Wang, L. and Xu, Y. [2003], 'SEGID: Identifying interesting segments in (multiple) sequence alignments', *Bioinformatics* **19**(2), 297–298.

[98] Wang, T. D., Plaisant, C., Quinn, A. J., Stanchak, R., Murphy, S. and Shneiderman, B. [2008], Aligning temporal data by sentinel events: discovering patterns in electronic health records, *in* 'Proceedings of the SIGCHI conference on Human factors in computing systems', pp. 457–466.

[99] Wang, T. D., Plaisant, C., Shneiderman, B., Spring, N., Roseman, D., Marchand, G., Mukherjee, V. and Smith, M. [2009], 'Temporal summaries: Supporting temporal categorical searching, aggregation and comparison', *IEEE Transactions on Visualization and Computer Graphics* **15**(6), 1049–1056.

[100] Wei, J., Shen, Z., Sundaresan, N. and Ma, K.-L. [2012], Visual cluster exploration of web clickstream data, *in* '2012 IEEE Conference on Visual Analytics Science and Technology (VAST)', IEEE, pp. 3–12.

[101] Wongsuphasawat, K. and Gotz, D. [2012], 'Exploring flow, factors, and outcomes of temporal event sequences with the Outflow visualization', *IEEE Transactions on Visualization and Computer Graphics* **18**(12), 2659–2668.

[102] Wongsuphasawat, K., Guerra Gómez, J. A., Plaisant, C., Wang, T. D., Taieb-Maimon, M. and Shneiderman, B. [2011], LifeFlow: visualizing an overview of event sequences, *in* 'Proceedings of the SIGCHI conference on human factors in computing systems', ACM, pp. 1747–1756.

[103] Wongsuphasawat, K. and Lin, J. [2014], Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at twitter, *in* '2014 IEEE Conference on Visual Analytics Science and Technology (VAST)', IEEE, pp. 113–122.

[104] Yan Holtz for Data-to-Viz [2021], 'How to avoid overplotting'.
URL: *https://www.data-to-viz.com/caveat/overplotting.html*

[105] Zgraggen, E., Drucker, S. M., Fisher, D. and DeLine, R. [2015], (s—qu)eries: Visual regular expressions for querying and exploring event sequences, *in* 'Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems', pp. 2683–2692.

[106] Zhao, J., Liu, Z., Dontcheva, M. et al. [2015], MatrixWave: Visual comparison of event sequence data, *in* 'Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems', pp. 259–268.

[107] Zhou, M., Yang, S., Li, X., Lv, S., Chen, S., Marsic, I., Farneth, R. A. and Burd, R. S. [2017], Evaluation of trace alignment quality and its application in medical process mining, *in* '2017 IEEE International Conference on Healthcare Informatics (ICHI)', IEEE, pp. 258–267.

Activities and publications during PhD

Publications

- Magallanes, J., van Gemeren, L., Wood, S. and Villa-Uriol, M.C., 2019, October. Analyzing Time Attributes in Temporal Event Sequences. In 2019 IEEE Visualization Conference (VIS) (pp. 1-5). IEEE.
- Magallanes, J., Stone, T., Morris, P. D., Mason, S., Wood, S., and Villa-Uriol, M. C., 2021. Sequen-C: A Multilevel Overview of Temporal Event Sequences. IEEE Transactions on Visualization and Computer Graphics, in press.

Conferences and workshops

- Manchester Digital Epidemiology Summer School. July 2018.
- Conference IEEE VIS 2018 at Berlin, Germany. October 2018.
- Conference IEEE VIS 2019 at Vancouver, Canada. October 2019.
- Online conference IEEE VIS 2020. October 2020.
- INSIGNEO Showcase 2017, 2018, and 2019.

Presentations

- Presented poster at the conference IEEE VIS 2018.
- Presented my work to a panel of senior visualization researchers at the IEEE VIS Doctoral Colloquium 2019.
- Presented conference paper "Analyzing Time Attributes in Temporal Event Sequences" at the IEEE VIS 2019.
- Talk at the Early Career Researcher session, INSIGNEO Showcase 2019.
- Talk and poster presentation at the Engineering Research Symposium 2019.
- Presented poster at INSIGNEO Showcase 2017, 2018, and 2019.

Work and supervising experience

- Research job at the Royal Hallamshire Hospital as part of the project funded by The Health Foundation. September 2019 to March 2021.
- Supervision of an undergraduate student project as part of the INSIGNEO Summer Placement 2020. Co-supervision of a second project under the same program. June to September 2020.

- Graduate Teaching Assistant at the modules COM6102, COM1003, COM6515 and BIE103. February 2018 to June 2019.

Awards

- Honorable Mention Poster Award at the IEEE VIS 2018 VAST track.
- Selected as a participant of the IEEE VIS Doctoral Colloquium 2019.
- Grant awarded by The Health Foundation with a project based on my PhD work.

Appendices

Appendix A

Data dictionary CUREd dataset

The following table describes the data attributes used in the CUREd case study presented in Chapter 6. The description was extracted from the public UOS CUREd Data Dictionary [93].

The code list for the attribute *area classification code* can be found at [75].

The code list for the attribute *attendance disposal* can be found at [73].

Field title	**Field name**	**Category**	**Description**
Area classification code	inc_oac11	Geographical	Output Area (oa11) classification. Derived from Postcode of incident.
Age at call	amb_callage	Patient data	Main patient's age in whole completed years, calculated from the call date and date of birth if present.
Symptom group description	amb_symptom	Call data	The symptom group description, usually derived from AMPDS code. This data is not cleaned.
Urgency level	amb_urgency	Call data	The urgency level classification of the call. This data is not cleaned.
Call stop reason	amb_callstop	Call data	Details the reason the call was stopped if not conveyed. This data is not cleaned.
Source of call	amb_callsorc	Call data	A categorisation of the profession/service from which the call originated. This data is not cleaned.
Attendance disposal	ae_attenddisp	Attendances	Classification of how the attendance was concluded.
Sex of patient	amb_sex	Patient data	Defines the sex of the patient. The classification is phenotypical rather than genotypical.

Table A.1: *Description of data attributes used in the CUREd case study.*

Appendix B

Evaluation: ethics application

This appendix includes the documents related to the ethics application sought to perform the evaluation presented in chapter 7:

- Approval letter for the ethics application provided by the Department of Computer Science at The University of Sheffield.
- Submitted ethics application.
- Consent form.
- Information sheet.

Jessica Magallanes Castaneda
Registration number: 160263776
Computer Science
Programme: PhD

Dear Jessica

PROJECT TITLE: Evaluation of the visual analytics system Sequen-C
APPLICATION: Reference Number 039940

On behalf of the University ethics reviewers who reviewed your project, I am pleased to inform you that on 19/05/2021 the above-named project was **approved** on ethics grounds, on the basis that you will adhere to the following documentation that you submitted for ethics review:

- University research ethics application form 039940 (form submission date: 18/05/2021); (expected project end date: 30/12/2021).
- Participant information sheet 1091349 version 3 (18/05/2021).
- Participant consent form 1091350 version 2 (18/05/2021).

If during the course of the project you need to deviate significantly from the above-approved documentation please inform me since written approval will be required.

Your responsibilities in delivering this research project are set out at the end of this letter.

Yours sincerely

Com Ethics
Ethics Administrator
Computer Science

Please note the following responsibilities of the researcher in delivering the research project:

- The project must abide by the University's Research Ethics Policy: https://www.sheffield.ac.uk/rs/ethicsandintegrity/ethicspolicy/approval-procedure
- The project must abide by the University's Good Research & Innovation Practices Policy: https://www.sheffield.ac.uk/polopoly_fs/1.671066!/file/GRIPPolicy.pdf
- The researcher must inform their supervisor (in the case of a student) or Ethics Administrator (in the case of a member of staff) of any significant changes to the project or the approved documentation.
- The researcher must comply with the requirements of the law and relevant guidelines relating to security and confidentiality of personal data.
- The researcher is responsible for effectively managing the data collected both during and after the end of the project in line with best practice, and any relevant legislative, regulatory or contractual requirements.

Application 039940

Section A: Applicant details

Date application started:
Wed 5 May 2021 at 12:05

First name:
Jessica

Last name:
Magallanes Castaneda

Email:
jgmagallanescastaneda1@sheffield.ac.uk

Programme name:
PhD

Module name:
Computer Science
Last updated:
19/05/2021

Department:
Computer Science

Applying as:
Postgraduate research

Research project title:
Evaluation of the visual analytics system Sequen-C

Has your research project undergone academic review, in accordance with the appropriate process?
Yes

Similar applications:
(032412) Analysis of pathways and temporal patterns in the CUREd research database

Section B: Basic information

Supervisor

Name	Email
Maria-Cruz Villa-Uriol	m.villa-uriol@sheffield.ac.uk

Proposed project duration

Start date (of data collection):
Wed 5 May 2021

Anticipated end date (of project)
Thu 30 December 2021

3: Project code (where applicable)

Project externally funded?
No

Project code
- not entered -

Suitability

Takes place outside UK?
No

Involves NHS?
No

Health and/or social care human-interventional study?
No

ESRC funded?
No

Likely to lead to publication in a peer-reviewed journal?
Yes

Led by another UK institution?
No

Involves human tissue?
No

Clinical trial or a medical device study?
No

Involves social care services provided by a local authority?
No

Is social care research requiring review via the University Research Ethics Procedure
No

Involves adults who lack the capacity to consent?
No

Involves research on groups that are on the Home Office list of 'Proscribed terrorist groups or organisations?
No

Indicators of risk

Involves potentially vulnerable participants?
No
Involves potentially highly sensitive topics?
No

Section C: Summary of research

1. Aims & Objectives

The objective is to carry out an evaluation of a software called Sequen-C, a visual analytics system for temporal event sequences. The software Sequen-C has been developed as part of my PhD and implements the methodologies proposed in my research. The purpose of the software is to obtain patterns and insights from temporal event sequences (e.g. event logs). Given a dataset containing the status of individuals (e.g. patients, customers) at specific points in time, the system can visualise the most frequent patterns and the characteristics of the individuals following such patterns. For example, when analysing an online shopping dataset, the system Sequen-C allows to obtain patterns such as "customers in their 20s usually buy product B after purchasing product A". The system Sequen-C will be evaluated in terms of usability and its ability to enable users to obtain insights from the data. The participants of this evaluation will interact with the software Sequen-C to produce visualizations and obtain insights from the data. The type of data analysed by the participant will be anonymised operational data from two clinical domains: data representing the flow of patients in an outpatient clinic and data representing calls to the emergency department. The data analysed by the participant through the system Sequen-C cannot be downloaded or accessed by the participant, as all the interaction with the software will be through a virtual machine.

2. Methodology

The evaluation consists of three main steps. First, the participant will watch a training video that explains how to interact with the system Sequen-C-2 and how to interpret the visualizations. Second, participants will answer a list of questions to make sure they have learnt the basic concepts. Participants cannot move to the evaluation until the questions are answered correctly, they will be free to ask questions to clarify any concept or solve issues. Third, the evaluation starts, where the participant will provide answers to two questionnaires: 1) a quantitative questionnaire, and 2) the System Usability Scale (SUS) questionnaire. The participant will carry out a set of tasks in order to answer these questions.

The evaluation and order of questions or tasks will be guided by a Google Form.

Firstly, the participant will answer the quantitative questionnaire by interacting with Sequen-C. An example of a quantitative question would be “According to the visualisation, what is the pattern that contains the highest number of female individuals?”; after reading the question, the participant is expected to interact with the system to obtain the answer and then type such answer in the Google Form. Quantitative questions will be multiple choice or open text. The output of this part of the evaluation is completion time and accuracy, this means, the time that the participant takes in answering each question and whether the answer is correct or not.

Secondly, the participant will answer the SUS questionnaire, based on his/her experience using the system. The SUS questionnaire is a popular and standardized set of questions used to assess perceived usability. The SUS questionnaire consists of 10 questions, where each question is answered with a number from one to five, one meaning “strongly disagree” and five meaning “strongly agree”. An example of a question from the SUS questionnaire is “I thought there was too much inconsistency in this system” or “I would imagine that most people would learn to use this system very quickly”.

The evaluation will finish when the participant answers all the questions presented in the Google Form.

Due to current restrictions, the evaluation will be conducted online by video call. Only the participant and the researcher will be present. Participants will get access to a virtual machine that will have Sequen-C installed and ready to be used. The participant will share her/his screen with the researcher during the video call. The researcher will write down the time taken by the participant to answer each question in the Google Form. Additionally, the screen of the call will be recorded as a backup of the timings – however, this video will be deleted as soon as the results are processed. The Google Form will only save the participant number, age, and gender. The name or email of the participant will not be saved, so it will not be possible to link a participant number with a specific person.

The evaluation per participant is expected to last around 30 minutes up to an hour.

- A better wording would be that the participant has to carry out a set of tasks

  Mon 17 May 2021 at 16:46

3. Personal Safety

Have you completed your departmental risk assessment procedures, if appropriate?

Not applicable

Raises personal safety issues?

No

The risk of the researcher conducting the evaluation is minimal as the main activity is to have a video call with the participant and observe the interaction with the system. There are no concerns of personal safety as the researcher will be working from home, which has been the usual place of work since the pandemic started.

Section D: About the participants

1. Potential Participants

Healthy adults that know how to use a computer, install programs, and start a video call. Two groups of users will be enrolled in the study. Group 1: Novice users. These participants should not have any previous knowledge of the software Sequen-C, but should be familiar with basic data visualizations such as bar charts or pie charts.
Group 2: Expert users. These participants will have had some experience either with Sequen-C (or previous versions).

It is intended to enroll 15 novice participants and 5 expert participants.

2. Recruiting Potential Participants

Novice user group: publication on social media (e.g. Facebook, LinkedIn) to invite friends and colleagues (except those

familiar with the software) to participate voluntarilly. Expert user group: the researcher will contact current research collaborators who have been exposed to Sequen-C (or to a previous version of it) via email, for example, collaborators that have used Sequen-C before to analyse their own data (therefore they are considered expert users). They will be asked to participate voluntarily. A special care will be put in not coercing participants to conduct the evaluation. They will be invited only once and they can come back to the researcher if they wish to.

- have the expert users consented previously to be re-contacted?
  Mon 17 May 2021 at 16:48

2.1. Advertising methods

Will the study be advertised using the volunteer lists for staff or students maintained by CiCS? No

- not entered -

3. Consent

Will informed consent be obtained from the participants? (i.e. the proposed process) Yes

Prior to the participation, a consent form will be shared with participants via email. The participant will provide consent by replying via email saying that they have read and agree to each and every point in the consent form. To avoid storage of the email in the consent form, uppon arrival of the email, an screenshot of the email consenting to participate will be taken. However, the email and name will be cropped from the image and the participant ID will be inserted instead, the image will be saved as proof of consent. The image will be saved in the UniDrive researcher account. All email exchanged between the researcher and the participants will be deleted from the researcher's university email account. This is the most practical way of obtaining consent, given the social distancing restrictions it would be more difficult to obtain physical signature.

4. Payment

Will financial/in kind payments be offered to participants? No

5. Potential Harm to Participants

What is the potential for physical and/or psychological harm/distress to the participants?

There are no known risks of taking part in this evaluation. The evaluation consists of testing a software in a computer or laptop. The nature of the questions being asked poses minimal risks, as they only deal with the analysis of data. In a extreme case, a potential risk is that the participant feels stressed if the questions are too complicated for her/him.

How will this be managed to ensure appropriate protection and well-being of the participants?

The participant is free to widthdraw from the evaluation at any point, before the evaluation starts or during the evaluation. No explanation needs to be given. As soon as the participant wants to widthdraw from it, the evaluation finishes. Participants can skip questions if they find them too complicated and continue with the next question.

- Can the participant simply skip a question if too complicated or do they need to either continue or withdraw?
  Mon 17 May 2021 at 16:49

6. Potential harm to others who may be affected by the research activities

Which other people, if any, may be affected by the research activities, beyond the participants and the research team?

none

What is the potential for harm to these people?

none

How will this be managed to ensure appropriate safeguarding of these people?

does not apply

7. Reporting of safeguarding concerns or incidents

What arrangements will be in place for participants, and any other people external to the University who are involved in, or affected by, the research, to enable reporting of incidents or concerns?

The consent form will contain the contact details of the researcher, the researcher's supervisor, and the head of department. The participant will be made aware before the evaluation that they are free to contact the supervisor or the

head of department if they wish to report any concern or incident.

Who will be the Designated Safeguarding Contact(s)?

the supervisor

How will reported incidents or concerns be handled and escalated?

The departamental ethics contact will be contacted and he/she will decide if it is necessary to escalate at university level.

Section E: About the data

1. Data Processing

Will you be processing (i.e. collecting, recording, storing, or otherwise using) personal data as part of this project? (Personal data is any information relating to an identified or identifiable living person).
Yes

Which organisation(s) will act as Data Controller?

University of Sheffield only

2. Legal basis for processing of personal data

The University considers that for the vast majority of research, 'a task in the public interest' (6(1)(e)) will be the most appropriate legal basis. If, following discussion with the UREC, you wish to use an alternative legal basis, please provide details of the legal basis, and the reasons for applying it, below:

a task in the public interest (6(1)(e))

Will you be processing (i.e. collecting, recording, storing, or otherwise using) 'Special Category' personal data?
No

3. Data Confidentiality

What measures will be put in place to ensure confidentiality of personal data, where appropriate?

The video recording of the call will be safely stored in the available university storage (UniDrive), no local copies will be stored. The video will only be used to obtain the timings taken to complete each task, as soon as the timings are processed the video will be deleted. The age and gender of the participant will be aggregated to report the age range and number of female/male individuals. Neither the name nor the email of the participant will be stored at all, the participant will be assigned an anonymous participant number which will not allow linkage to any specific person. To avoid storage of the email in the consent form, upon arrival of the email, an screenshot of the email consenting to participate will be taken. However, the email and name (if present in the email) will be cropped from the image and the participant number will be inserted instead, the image will be saved as proof of consent. The image will be saved in the UniDrive researcher account. All email exchanges between the researcher and the participants will be deleted from the researcher university email account.

- By receiving a consent form and exchanging emails/contacts for video calls you are still collecting personal data - you must make sure they are stored separately from the replies.
  Mon 17 May 2021 at 16:50

4. Data Storage and Security

In general terms, who will have access to the data generated at each stage of the research, and in what form

Only the researcher will have access to the recorded videos, an anonymous participant number will be assigned to each participant and will be used to link to the video. The videos will be exclusively stored in the UniDrive of the researcher using her own university account, nobody else will have access to it. After the results have been processed and transcribed to a report, the recorded videos will be immediately deleted. The anonymised questionnaires will be securely stored in The University of Sheffield Research Data Catalogue and Repository (ORDA) for future reference, as it might be a research interest to compare the present study with future versions of Sequen-C.

What steps will be taken to ensure the security of data processed during the project, including any identifiable personal data, other than those already described earlier in this form?

No data will be identifiable. In the case of the audio of the video, which might be identifiable, will be destroyed as soon as the timings of each video are processed. The name of the participant will not be stored at all, the participant will be assigned an anonymous participant number which will not allow linkage to any specific person.

Will all identifiable personal data be destroyed once the project has ended?
Yes

Please outline when this will take place (this should take into account regulatory and funder requirements).

Up to 1 month after the timings/results have been writen in the report.

Section F: Supporting documentation

Information & Consent

Participant information sheets relevant to project?
Yes

Document 1091349 (Version 3) — All versions
information sheet

- Be careful that in the PIS there is still a comment
  Mon 17 May 2021 at 16:54

Consent forms relevant to project?
Yes

Document 1091350 (Version 2) — All versions

Additional Documentation

External Documentation

The System Usability Scale (SUS) questionnaire can be found here: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

Section G: Declaration

Signed by:
Jessica Gisela Magallanes Castaneda
Date signed:
Tue 18 May 2021 at 18:04

Offical notes

- not entered -

The University Of Sheffield.

Participant Consent Form
Software Sequen-C Consent Form

Please tick the appropriate boxes	Yes	No
Taking Part in the Project		
I have read and understood the project information sheet dated [18/05/2021] or the project has been fully explained to me. (If you will answer No to this question please do not proceed with this consent form until you are fully aware of what your participation in the project will mean.)	☐	☐
I have been given the opportunity to ask questions about the project.	☐	☐
I agree to take part in the project. I understand that taking part in the project will include: • Watching a video to get some training in the use of the software Sequen-C. You will be able to ask questions to the researcher. • Performing a list of tasks in the software Sequen-C while I share my screen with the researcher on a video call. I will be able to ask questions during that time. • The video call screen will be recorded (including video and audio). • Complete two questionnaires.	☐	☐
I understand that by choosing to participate as a volunteer in this research, this does not create a legally binding agreement nor is it intended to create an employment relationship with the University of Sheffield.	☐	☐
I understand that my taking part is voluntary and that I can withdraw from the study at any time; I do not have to give any reasons for why I no longer want to take part and there will be no adverse consequences if I choose to withdraw. I can also skip questions if I find them too complicated and continue with the next question.	☐	☐
How my information will be used during and after the project		
I understand my personal details such as name, email, age, and gender will not be revealed to people outside the project.	☐	☐
I understand and agree that my words may be quoted in publications, reports, web pages, and other research outputs. A participant number will be used to refer to a specific participant.	☐	☐
I understand and agree that other authorised researchers will have access to the recorded video and questionnaires only if they agree to preserve the confidentiality of the information as requested in this form.	☐	☐
I understand and agree that other authorised researchers may use the questionnaires in publications, reports, web pages, and other research outputs, only if they agree to preserve the confidentiality of the information as requested in this form.	☐	☐
I understand and agree that after the results have been processed and transcribed to a report, the recorded videos will be immediately deleted. The anonymised questionnaires will be securely stored in The University of Sheffield Research Data Catalogue and Repository (ORDA) for future reference.	☐	☐
So that the information you provide can be used legally by the researchers		
I agree to assign the copyright I hold in any materials generated as part of this project to The University of Sheffield.	☐	☐

Name of participant [printed] Signature Date

Name of Researcher [printed] Signature Date

Project contact details for further information:

Researcher: Jessica Gisela Magallanes Castaneda (jgmagallanescastaneda1@sheffield.ac.uk)
Supervisor: Dr Maria-Cruz Villa-Uriol (m.villa-uriol@sheffield.ac.uk)

Participant Information Sheet (18th May 2021)

Evaluation of the Visual Analytics System Sequen-C

You are being invited to take part in a research project. Before you decide whether or not to participate, it is important for you to understand why the research is being done and what it will involve. Please take time to read the following information carefully and discuss it with others if you wish. Ask us if there is anything that is not clear or if you would like more information. Take time to decide whether or not you wish to take part. Thank you for reading this.

1. What is the project's purpose?

The objective is to carry out an evaluation of a visual analytics system called Sequen-C, which has been developed as part of the researcher's PhD. The purpose of the software is to obtain patterns and insights from temporal event sequences (e.g. event logs). Given a dataset containing the status of individuals (e.g. patients, customers) at specific points in time, the system can visualise the most frequent patterns and their characteristics. For example, when analysing an online shopping dataset, the system Sequen-C allows to obtain patterns such as "customers in their 20s usually buy product B after purchasing product A". The system Sequen-C will be evaluated in terms of usability and its ability to enable users to obtain insights from the data. The participants of this evaluation will interact with the software Sequen-C to produce visualizations and obtain insights from the data.

2. Why have I been chosen?

You have been chosen as the project requires adults with no previous knowledge of the software.

3. Do I have to take part?

Taking part in this research is entirely voluntary. It is up to you to decide whether or not to take part. If you do not wish to take part, there will be no negative consequences. If you decide to take part you will be given this information sheet to keep and you will be asked to sign a consent form, you can still withdraw at any point before or during the evaluation without any negative consequences. You can skip questions if you find them too complicated and continue with the next question. You can withdraw from the research without giving a reason. If you wish to withdraw from the research, please contact Jessica Magallanes or Dr Maria-Cruz Villa-Uriol (see contact details at the end of this document).

Please note that that by choosing to participate in this research, this will not create a legally binding agreement, nor is it intended to create an employment relationship between you and the University of Sheffield.

4. What will happen to me if I take part? What do I have to do?

The evaluation consists of three main steps. First, you will watch a training video that explains how to interact with the system Sequen-C and how to interpret the visualizations. Second, you will answer a list of questions to make sure you have learnt the basic concepts. You cannot start with the evaluation until the questions are answered correctly, of course you are free to ask questions to clarify any concept or solve issues. Third, the evaluation starts, where you will be asked to provide answers to two questionnaires: 1) a quantitative questionnaire, and 2) the System Usability Scale (SUS) questionnaire.

The evaluation and order of questions will be guided by a Google Form.

Firstly, you will answer the quantitative questionnaire by interacting with Sequen-C. An example of a quantitative question would be "According to the visualisation, what is the pattern that contains the highest number of female individuals?"; after reading the question, you are expected to interact with the system to obtain the answer and then type such answer in the Google Form. Quantitative questions will be multiple choice or open text. The output of this part of the evaluation is completion time and accuracy, this means, the time that you take in answering each question and whether the answer is correct or not.

Secondly, you will be asked to answer the SUS questionnaire, based on your experience using the system. The SUS questionnaire is a popular and standardized set of questions used to assess perceived usability. The SUS questionnaire consists of 10 questions, where each question is answered with a number from one to five, one meaning “strongly disagree” and five meaning “strongly agree”. An example of a question from the SUS questionnaire is “I thought there was too much inconsistency in this system” or “I would imagine that most people would learn to use this system very quickly”.

The evaluation will finish when you answer all the questions presented in the Google Form.

Due to current restrictions, the evaluation will be conducted online by video call. Only you and the researcher will be present. You will get access to a virtual machine that will have Sequen-C installed and ready to be used. You will be asked to share your screen with the researcher during the video call. The researcher will write down the time you take to answer each question. Additionally, the screen of the call will be recorded as a backup of the timings – however, this video will be deleted as soon as the results are processed. The Google Form will only save an anonymous participant number, and your age and gender. Your name will not be saved, so it will not be possible to link a participant number with a specific person.

The evaluation per participant is expected to last from 1 to 2 hours.

5. Will I be recorded, and how will the recorded media be used?

The screen of the video call will be recorded. The purpose is to record the software being used and to measure the time it takes to answer each question. The recording will include audio and you are free to turn off your camera if you do not wish to appear in the recording. After the results have been processed and transcribed to a report, the recorded videos will be immediately deleted. Your name and email will not be associated at any time to the video.

6. What are the possible disadvantages and risks of taking part?

There are no known risks of taking part in this evaluation. If before or during the recording you decide that you do not wish to continue, the evaluation and recording will finish at that very moment. You do not have to provide any reason and the existing recording, if any, will be immediately deleted.

7. What are the possible benefits of taking part?

Whilst there are no immediate benefits for those people participating in the project, it is hoped that this work will contribute to the state the art in the visual analytics of temporal event sequences.

8. Will my taking part in this project be kept confidential?

All the information collected in this evaluation (such as video recordings) will be kept strictly confidential in the university storage UniDrive. Your age and gender will be only used to report the age range and number of female/male participants. You will be assigned an anonymous participant number and your name will not be stored at all. After the results have been processed and transcribed to a report, the recorded videos will be immediately deleted. The anonymised questionnaires will be securely stored in The University of Sheffield Research Data Catalogue and Repository (ORDA) for future reference as it might be interesting to assess how Sequen-C has evolved.

9. What is the legal basis for processing my personal data?

According to data protection legislation, we are required to inform you that the legal basis we are applying in order to process your personal data is that ‘processing is necessary for the performance of a task carried out in the public interest’ (Article 6(1)(e)). Further information can be found in the University’s Privacy Notice https://www.sheffield.ac.uk/govern/data-protection/privacy/general.

10. What will happen to the data collected, and the results of the research project?

Video recordings will be deleted immediately after the results have been transcribed in a report. The anonymised questionnaires will be securely stored in The University of Sheffield Research Data Catalogue and Repository (ORDA) for future reference. The results of this evaluation will be reported in the researcher's PhD thesis and probably in future journal publications. Due to the nature of this research, it is likely that other researchers may find the collected questionnaires to be useful in answering future research questions. We will ask for your explicit consent if the collected questionnaires are to be shared with other researchers.

11. Who is organising and funding the research?

This PhD project has been funded by CONACYT and by The Health Foundation.

12. Who is the Data Controller?

The University of Sheffield will act as the Data Controller for this study. This means that the University is responsible for looking after your information and using it properly.

13. Who has ethically reviewed the project?

This project has been ethically approved via the University of Sheffield's Ethics Review Procedure, as administered by the Computer Science department. The University's Research Ethics Committee monitors the application and delivery of the University's Ethics Review Procedure across the University.

14. What if something goes wrong and I wish to complain about the research or report a concern or incident?

It you are dissatisfied with any aspect of the research and wish to make a complaint, please contact Jessica Magallanes or Dr Maria-Cruz Villa-Uriol in the first instance (see contact details at the end of this document). If you feel your complaint has not been handled in a satisfactory way you can contact the Head of the Department of Computer Science, Professor Guy Brown (g.j.brown@sheffield.ac.uk) who will then escalate the complaint through the appropriate channels. If the complaint relates to how your personal data has been handled, you can find information about how to raise a complaint in the University's Privacy Notice: https://www.sheffield.ac.uk/govern/data-protection/privacy/general.

15. Contact for further information

Please contact us if you have any questions.

Researcher: Jessica Gisela Magallanes Castaneda (jgmagallanescastaneda1@sheffield.ac.uk)

Supervisor: Dr Maria-Cruz Villa-Uriol (m.villa-uriol@sheffield.ac.uk)

Thank you very much for your time to read this information sheet.

You will be given a copy of this information sheet and, if appropriate, a signed consent form to keep.

Appendix C

Evaluation: participant completion times

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Total
P1	01:23.7	01:14.9	02:27.8	00:57.8	00:32.1	00:21.8	00:03.1	00:13.9	00:32.8	00:21.0	00:29.4	00:54.4	02:43.7	12:16.4
P2	03:35.0	00:28.2	02:24.9	00:32.8	00:37.0	00:46.2	00:04.0	01:04.5	00:44.2	01:17.4	01:39.5	00:32.6	02:03.5	15:49.8
P3	01:39.1	01:49.0	03:20.7	00:39.3	02:11.8	01:36.5	00:02.8	00:37.2	01:28.9	00:44.5	01:10.2	00:42.9	02:33.7	18:36.7
P4	02:39.0	02:08.0	02:42.0	00:59.1	00:36.0	00:22.2	00:04.0	00:01:44	01:05.6	00:41.1	01:27.3	02:11.9	04:47.6	21:28.0
P5	02:36.4	00:12.2	02:35.4	00:12.6	00:14.4	00:20.8	00:05.4	00:10.2	00:23.1	00:34.3	00:31.6	00:47.2	01:49.5	10:33.2
P6	01:30.8	02:33.2	02:01.5	00:34.5	00:23.0	00:32.7	00:03.9	00:40.1	00:37.3	00:31.8	00:07.0	00:43.3	01:19.6	11:38.8
P7	01:58.0	00:25.6	03:14.8	00:23.8	00:19.6	01:51.9	00:07.1	00:11.7	00:51.0	00:40.7	00:28.7	00:46.9	03:00.9	14:20.9
P8	00:34.2	00:18.5	00:40.0	01:00.4	00:20.6	01:26.7	00:03.0	00:16.0	00:20.3	00:40.3	00:23.3	00:07.0	01:06.8	07:17.0
P9	00:41.6	00:22.4	02:35.0	00:39.0	00:19.3	01:39.3	00:05.9	00:15.2	00:33.4	00:07.1	01:06.5	00:24.3	03:03.0	11:52.0
P10	02:21.0	01:38.7	03:54.4	01:39.4	00:11.2	00:56.0	00:07.3	00:24.4	01:26.0	01:12.9	01:26.6	01:56.0	04:22.0	21:36.0
P11	02:50.6	00:44.2	01:59.1	01:01.5	00:40.8	01:16.2	00:02.6	00:48.2	03:16.2	00:25.8	01:09.8	00:36.2	03:46.5	18:37.6
P12	00:36.0	00:21.4	01:07.7	00:53.7	00:19.0	01:21.4	00:04.3	00:35.9	00:15.2	00:34.5	00:30.8	00:42.6	01:09.6	08:32.1
P13	01:05.0	00:21.0	01:07.4	00:15.2	00:05.6	00:16.2	00:13.9	00:03.8	00:19.5	00:34.0	00:12.2	01:26.2	02:06.0	08:06.1
avg	01:48.5	00:58.3	02:19.3	00:45.3	00:31.6	00:59.1	00:05.2	00:32.7	00:54.9	00:38.9	00:49.5	00:54.7	02:36.3	13:54.2
min	00:34.2	00:12.2	00:40.0	00:12.6	00:05.6	00:16.2	00:02.6	00:03.8	00:15.2	00:07.1	00:07.0	00:07.0	01:06.8	07:17.0
median	01:39.1	00:28.2	02:27.8	00:39.3	00:20.6	00:56.0	00:04.0	00:24.4	00:37.3	00:34.5	00:31.6	00:43.3	02:33.7	12:16.4
max	03:35.0	02:33.2	03:54.4	01:39.4	02:11.8	01:51.9	00:13.9	01:44.3	03:16.2	01:17.4	01:39.5	02:11.9	04:47.6	21:36.0

Table C.1: *Completion time by participant (P1 to P13) per question (Q1 to Q13).*

Appendix D

Time performance of additional subsets of the data

This appendix contains further details of the complexity analysis presented in section 4.4 of chapter 4.

CUREd dataset							
Subset	**No. sequences**		**No. event types**	**Length of sequences**		**Execution time (s)**	
	Individual	**Unique**		**Average**	**Maximum**	`buildAggTree`	**Align**
5%	14232	48	11	7.33	9	0.01	0.38
10%	17869	96	11	8.43	10	0.00	1.23
15%	19065	144	11	8.96	11	0.01	3.17
20%	19696	192	11	9.47	11	0.02	5.70
25%	20124	240	11	9.97	13	0.03	9.27
30%	20200	289	11	10.56	14	0.04	13.49
35%	20417	337	11	11.15	15	0.05	19.29
40%	20738	385	11	11.76	17	0.06	32.71
45%	20794	433	11	12.44	18	0.17	41.27
50%	20901	481	11	13.07	19	0.10	59.11
55%	21006	529	11	13.69	20	0.12	83.32
60%	21327	577	11	14.28	22	0.14	105.54
65%	21400	625	11	14.87	23	0.17	143.82
70%	21461	673	11	15.52	25	0.19	174.42
75%	21520	722	11	16.25	28	0.21	225.50
80%	21583	770	11	17.04	30	0.27	297.74
85%	21657	818	11	17.88	33	0.26	421.03
90%	21707	866	11	18.88	38	0.47	614.18
95%	21757	914	11	20.07	48	0.36	895.94
100%	21805	962	11	22.70	177	0.37	3984.67

Table D.1: *Time performance of the* `buildAggregateTree` *(buildAggTree) and Align functions for additional subsets of the CUREd data.*

MIMIC-III dataset							
Subset	**No. sequences**		**No. event types**	**Length of sequences**		**Execution time (s)**	
	Individual	**Unique**		**Average**	**Maximum**	`buildAggTree`	**Align**
5%	135	66	51	4.42	5.00	0.05	0.19
10%	240	131	88	4.94	6.00	0.01	0.40
15%	310	197	112	5.37	7.00	0.12	1.49
20%	375	262	139	5.77	7.00	0.04	2.31
25%	442	328	158	6.07	8.00	0.07	3.46
30%	507	393	184	6.39	8.00	0.10	6.00
35%	573	459	199	6.62	8.00	0.20	6.44
40%	638	524	221	6.88	9.00	0.17	8.74
45%	704	590	231	7.11	9.00	0.21	10.63
50%	770	656	242	7.31	10.00	0.22	16.91
55%	835	721	261	7.55	10.00	0.35	24.75
60%	901	787	277	7.76	10.00	0.34	33.90
65%	966	852	301	7.99	11.00	0.36	37.57
70%	1032	918	322	8.21	11.00	0.48	48.80
75%	1097	983	338	8.45	12.00	0.51	73.36
80%	1163	1049	355	8.71	13.00	1.30	77.56
85%	1228	1114	373	8.99	14.00	0.80	109.02
90%	1294	1180	388	9.34	16.00	1.22	147.31
95%	1359	1245	412	9.79	20.00	1.37	228.06
100%	1425	1311	448	10.68	96.00	1.46	995.39

Table D.2: *Time performance of the* `buildAggregateTree` *(buildAggTree) and Align functions for additional subsets of the MIMIC-III data.*

Appendix E

Design and development of Sequen-C

The GUI/front-end and most of the back-end of the system Sequen-C were implemented in Java, except for the clustering and alignment algorithms which were implemented in R (see section 4.4.1 in Chapter 4). The Java libraries Swing and Graphics2D were used to build the GUI and visualisations. SQLite was used as the database management system to store the datasets used in the case studies. Fig. E.1 shows a sequence diagram of how the front-end interacts with the back-end components in order to generate the visualisation of the data loaded by the user. As observed, the aggregate tree (including clustering and alignment) is computed using an R script - whilst the score and simplify steps to create the data representation are computed in Java according to the current values of k and I_τ.

Fig. E.2 shows an Entity Relationship diagram of the main classes in Java. The classes in green are populated by the user via the files Events.csv and Attributes.csv, which then are used to compute the objects for the classes in blue. The SQLite database contains the tables Event, EventAttribute, SequenceAttribute, UniqueSequence, and Sequence - which reflect the same structure as the Java classes. These tables store the data uploaded by the user and data structures computed by Java, so when the system is run a second time, the data is loaded from the database rather than being recomputed again.

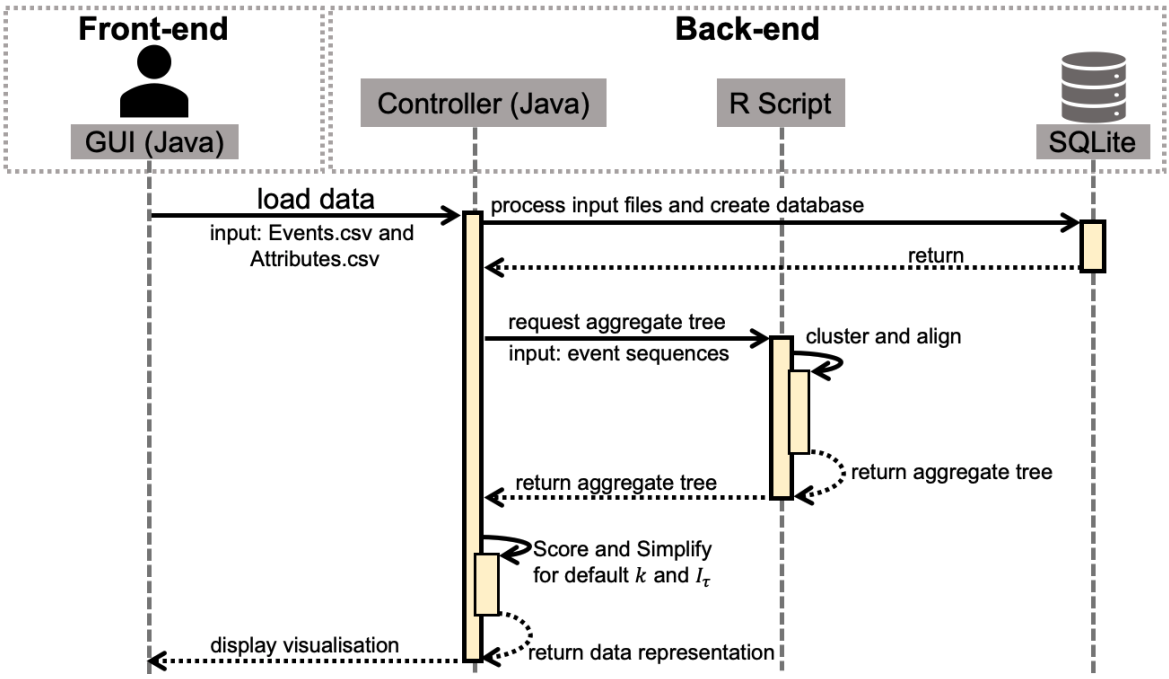


Figure E.1: *Sequence diagram showing the interaction between the front-end and back-end components of the system Sequen-C, to create a visualisation from the data provided by the user.*

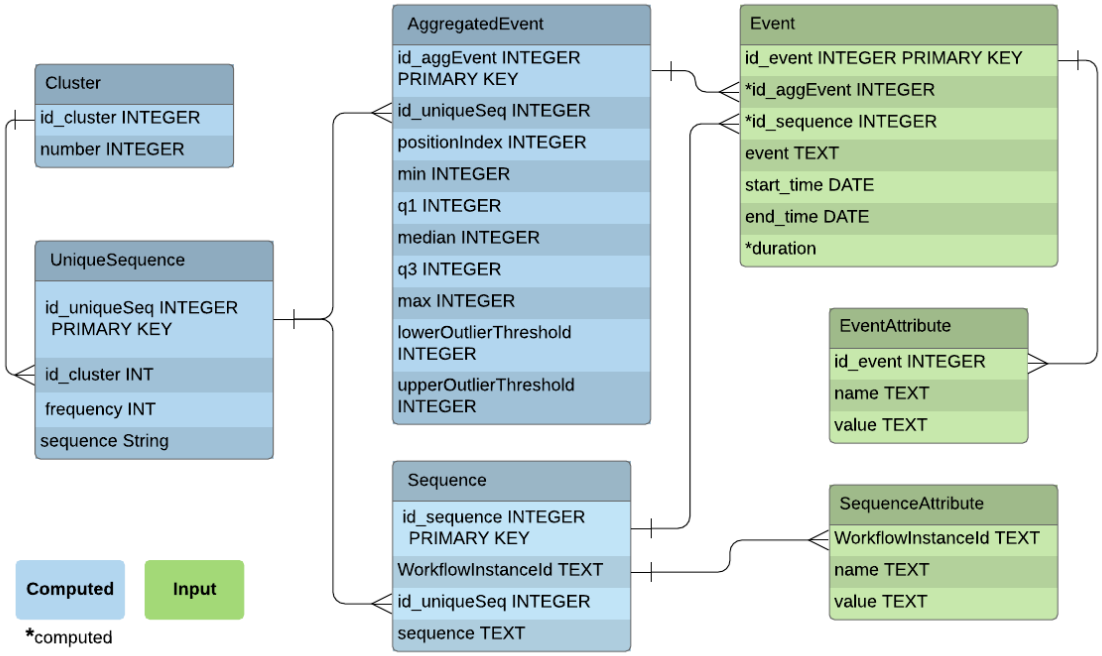


Figure E.2: *Entity Relationship diagram of the main classes in Java for the system Sequen-C. Classes in green colour are populated by the user via csv files, except for the fields marked with an asterisk which are automatically computed. Classes in blue colour are computed based on the green classes. The class AggregatedEvent refers to the data structure used to represent an EventBox.*