A NOVEL METHODOLOGICAL FRAMEWORK FOR THE ANALYSIS OF HEALTH TRAJECTORIES AND SURVIVAL OUTCOMES IN HEART FAILURE PATIENTS

Juliette Murris[1], **Tristan Amadei**[2], **Tristan Kirscher**[2], **Antoine Klein**[2],
Anne-Isabelle Tropeano[3] **& Sandrine Katsahian**[1,3]
[1] HeKA, Inserm, Inria, Université Paris Cité, Pierre Fabre R&D [2] ENSAE, IP de Paris,
[3] CIC-1418, HEGP, AP-HP, Paris, France
`juliette.murris@inria.fr`

ABSTRACT

Heart failure (HF) contributes to circa 200,000 annual hospitalizations in France. With the increasing age of HF patients, elucidating the specific causes of inpatient mortality became a public health problematic. We introduce a novel methodological framework designed to identify prevalent health trajectories and investigate their impact on death. The initial step involves applying sequential pattern mining to characterize patients' trajectories, followed by an unsupervised clustering algorithm based on a new metric for measuring the distance between hospitalization diagnoses. Finally, a survival analysis is conducted to assess survival outcomes. The application of this framework to HF patients from a representative sample of the French population demonstrates its methodological significance in enhancing the analysis of healthcare trajectories.

1 MOTIVATING EXAMPLE

Heart failure (HF) is a cardiovascular condition characterized by the heart's inability to pump sufficient blood to meet the body's oxygen and nutrient needs. It is a prevalent disease, affecting 1 to 2% of adults in developed countries, and around 64 million people worldwide (Savarese et al., 2022). Chronic HF often goes with repeated hospitalizations and embodies the condition with highest 30-days re-hospitalization rate (Constantinou et al., 2021). In France alone, over 1.5 million individuals suffer from HF, resulting in approximately 200,000 hospitalizations annually. Thus, understanding primary causes of death in these patients and identifying their most frequent inpatient trajectories holds significant potential for public health impact.

Electronic health records in France aggregate data from all hospitalizations. The EGB (*Echantillon Généraliste des Bénéficiaires*) is a random sample and representative of 1/97th of the population over a two year follow-up (De Roquefeuil et al., 2009). Trajectories are established using primary and associated diagnoses based on the International Classification of Disease (10th edition (ICD-10)). ICD-10 code syntax is text-based, and includes the principal diagnosis category, indication of surgical procedures, a counter, and a severity indicator (example given in Figure 1, '05M092' is the ICD-10 code for HF).

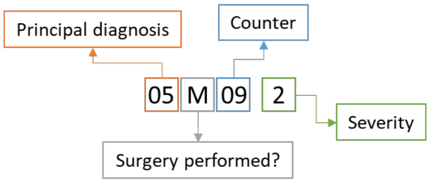


Figure 1: ICD-10 architecture

The primary objective of this work is to identify frequent inpatient health trajectories in HF patients in France. Secondary objectives include the investigation of their associations with mortality.

2 RELATED WORK

The analysis of trajectories in healthcare is a complex task which helps understanding the evolution of patients pathway over time. Many approaches deal clustering (Chouaïd et al., 2022; Lambert et al., 2023; Zhong et al., 2021), and others with times series, like Markov chains (MacDonald & Zucchini, 1997) or neural networks (Hewamalage et al., 2021). More recently the advent of text-based input enabled to collect more and more information (Nguyen et al., 2018; Zhu et al., 2021).

However, none of these methodologies combine pattern mining, clustering and survival analysis all together, even though these concepts are individually well-established and active research fields (Kang et al., 2020; Leis et al., 2023; Murris et al., 2023). This work is a demonstration of the assembling of existing tools to answer concrete clinical need and represents a ”bridging fields” contribution.

3 METHODOLOGICAL FRAMEWORK

3.1 SEQUENTIAL PATTERN MINING - THE *PrefixSpan* ALGORITHM

We used sequential pattern mining technique to extract frequent trajectories (Masseglia et al., 2004). *PrefixSpan* algorithm uses sequence patterns as tuples (P, sup), where P is a sequential pattern and sup is the number of sequences in the database that contain P (Pei et al., 2001). The sequence patterns are used to efficiently compute the support of candidate patterns and avoid unnecessary database scans. The algorithm efficiently identifies frequent sequential patterns in the database while maintaining a concise representation of the patterns (see Algorithm 1).

3.2 UNSUPERVISED LEARNING WITH PATIENT CLUSTERING

The clustering of patients brings the interpretability layer to the approach (Pinaire, 2017). Clusters should fit closely patients' trajectories based on their successive hospitalization sequences (see Algorithm 2). K-medoids algorithm was used to deal with string data and aims to partition input into k clusters (Kaur et al., 2014). Each cluster is represented by a single data point called medoid.

Levenshtein distance $\mathcal{D}_L$ was used to measure the distance between two hospitalization sequences (Yujian & Bo, 2007). Traditionally, the Levenshtein distance calculates the minimum number of edits to perform on single-characters to transform the word a into the word b:

$$\mathcal{D}_L(a,b) = \begin{cases} \max(|a|,|b|) & \text{if } \min(|a|,|b|) = 0, \\ lev(a_{1:}, b_{1:}) & \text{if } a[0] = b[0], \\ 1 + \min \begin{cases} lev(a_{1:}, b) \\ lev(a, b_{1:}) \\ lev(a_{1:}, b_{1:}) \end{cases} & \text{otherwise.} \end{cases}$$

with $|\cdot|$ the number of letters in the word, $\cdot_{1:}$ the word without its first letter and $\cdot[0]$ the first letter of the word. Throughout our study, we will only use the Levenshtein ratio, being the normalized value with $lev(a,b) = \frac{\mathcal{D}_L(a,b)}{\max(|a|,|b|)}$ with words of the same number of characters, hence $\max(|a|,|b|) = |a| = |b|$. We used this string-distance to implement a new metric $\mathcal{D}_{ICD10}$ to compute the distance between two ICD-10 codes A and B:

$$\begin{aligned} \mathcal{D}_{ICD10}(A,B) = & \omega_1 \times lev(A_{0:2}, B_{0:2}) \\ & + \omega_2 \times lev(A[2], B[2]) \\ & + \omega_3 \times lev(A_{3:5}, B_{3:5}) \\ & + \omega_4 \times lev(A[5], B[5]) \end{aligned}$$

with $\Omega = (\omega_i)_{i \in [1,4]} \in \mathbb{N}^+$. This weighted distance thus leverages the information from each ICD-10 component (as per Figure 1). Then, we compare the i^{th} ICD-10 code of a patient with the $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ ICD-10 codes of another patient, compute the distances and keep the minimum in order to get the distance between two patient sequences (Figure 2).

The distance $\mathcal{D}_P$ between two patient sequences is the sum of all ICD-10 codes to one another. This distance respects the symmetry assumption and $\mathcal{D}_P(patient_i, patient_j) = \mathcal{D}_P(patient_j, patient_i)$. Further details on the distance matrix are given in Appendix.

Two hyperparameters require settings and are under constraint: Ω the weights of the distance metric, with $0 \leq \omega_4 \leq \omega_3 \leq \omega_2 \leq \omega_1 \leq 100$, and $k \in [2, 20]$ the number of clusters. Cross-validation was used to find optimal hyperparameters using Optuna (Akiba et al., 2019). The score S to be maximised was defined as follows, for each cluster k:

$$s_{p,k} = \sum_{p=1}^{N_p} (P, sup)[p]_k - (P, sup)[p]$$

And $S = \frac{1}{N_p \times N} \sum_{k=1}^{N} \sum_{p}^{N_p} s_{p,k}$. We set $N_p = 3$. Based on outputs from the *PrefixSpan* algorithm, the idea is to determine the most frequently occurring ICD-10 code patterns of lengths 1, 2, 3 within each cluster, along with their respective frequencies. Subsequently, we calculate the frequency of these patterns across the entire dataset and compute the difference between the two. We then get the mean of these differences separately for patterns of length 1, 2, and 3. The clustering score S is then established by averaging these means.

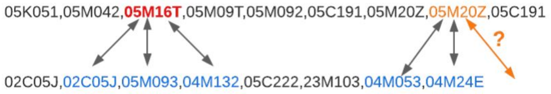

05K051,05M042,05M16T,05M09T,05M092,05C191,05M20Z,05M20Z,05C191
?
02C05J,02C05J,05M093,04M132,05C222,23M103,04M053,04M24E

Figure 2: Measuring distances for two patient sequences

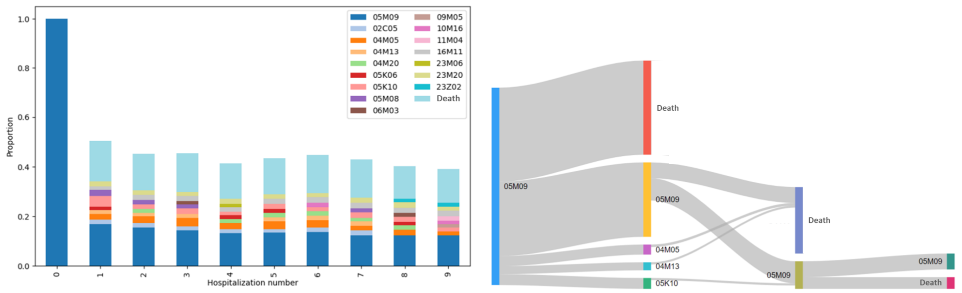
3.3 Survival analysis

For each patient, survival time and status are available. We implemented two ensemble methods, namely random survival forests (RSF) and survival gradient boosting. These methods are survival analysis counterparts of random forests and gradient boosting algorithms tailored for censored data (Ishwaran et al., 2008; Hothorn et al., 2006). Cross-validation was performed for hyperparameter optimization.

Two metrics were used for evaluation. We assessed the goodness of fit using the Akaike information criterion (AIC), with lower values indicating better fit (Hu, 2007). Besides, the concordance index (C-index) is a generalization of the ROC-AUC that can effectively handle censored data, providing a reliable ranking of survival times based on individual risk scores (Harrell et al., 1982; 1984).

4 Results

This study included adult HF patients who experienced their initial HF hospitalization between 2010 and 2016. The cohort comprised 10,051 patients, accounting for a total of 85,594 hospitalizations. Five clusters were identified with optimal hyperparameters ($\Omega^T = [85, 75, 55, 40]$).



(a) Frequent ICD-10 codes

(b) Sankey diagram

Figure 3: Key figures for Cluster 1

When we examined the ten most frequent ICD-10 codes for the n^{th} hospitalization after HF, we observed that these ICD-10 codes collectively accounted for approximately 50% from the first to the tenth occurrence. This indicates significant similarity in the care sequences of these patients (Figure 3). Furthermore, by visualizing the frequent trajectories for the entire HF patients population, we can identify several common sequences leading to death (e.g., '05M09' for HF hospitalization, '04M05' for pleurisy, or '04M13' for pulmonary edema and respiratory distress).

We also observed a high proportion of deaths during the initial hospitalizations in clusters 2 and 4, consistent with the older age and shorter hospitalization sequences of these patients (see Appendix). Additionally, HF patients in cluster 2 who experienced mortality had no more than six hospitalizations following their first HF episode.

Figure 4 displays the survival trajectories for the most and least optimistic scenarios within each cluster. Since clusters have varying numbers of individuals, prediction accuracy varies considerably across models. Specifically, cluster 3 yields more uncertain predictions compared to cluster 5. Aging consistently emerged as a significant factor contributing to mortality across all clusters, as well as being male. Prolonged hospital stays are also associated with a more pessimistic trajectory. We obtained similar results using RSF and survival gradient boosting, yielding a mean C-index of 0.68 (details in Appendix).

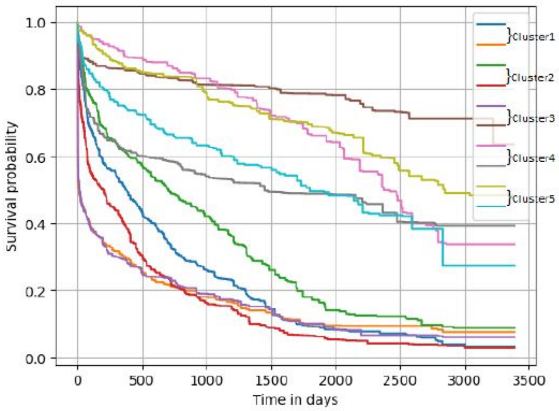


Figure 4: Survival predictions in clusters with best and worst scenarios

5 Discussion

By systematically identifying and analyzing frequent hospitalization patterns, we gained valuable insights into the care sequences of heart failure patients. This comprehensive analysis enabled to not only discern the most common reasons for hospitalization but also to trace the trajectories that often culminate in patient mortality.

Our study benefits from a large dataset, which provides a solid foundation for our analyses. A significant highlight of our methodology is the use of unsupervised clustering, not least with the introduction of a novel distance metric. More recently, methodological research has been conducted to create similarity measures for clustering purposes, either based on prevalence (Mannino et al., 2017), or Bayesian framework (Wu & Hao, 2020). However, our approach enables to keep into account the numbered order of hospitalizations, and to assess the impact on death using the survival analysis (Pinaire et al., 2017). Our methodology also allows us to avoid making any *a priori* assumptions about the patient population, thereby reducing potential biases associated with specific patient characteristics (clinical discussion is available in Appendix A.4).

Health trajectories is a burning topic in clinical research. Within our methodological framework, we introduce an innovative approach to comprehending hospitalization sequences and their implications for survival outcomes. While our focus has been on the HF patients, our approach is adaptable and can be extended to address more intricate populations, and this way to meet a variety of clinical challenges.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Mohammed Al-Maolegi and Bassam Arkok. An Improved Apriori Algorithm For Association Rules. *International Journal on Natural Language Computing*, 3(1):21–29, February 2014. ISSN 23194111, 22781307. doi: 10.5121/ijnlc.2014.3103. URL http://www.airccse.org/journal/ijnlc/papers/3114ijnlc03.pdf.

Christos Chouaïd, Valentine Grumberg, Alexandre Batisse, Romain Corre, Matteo Giaj Levra, Anne-Françoise Gaudin, Martin Prodel, Joannie Lortet-Tieulent, Jean-Baptiste Assié, and Francois-Emery Cotté. Machine learning–based analysis of treatment sequences typology in advanced non–small-cell lung cancer long-term survivors treated with nivolumab. *JCO Clinical Cancer Informatics*, 6:e2100108, 2022.

Panayotis Constantinou, Nathalie Pelletier-Fleury, Valérie Olié, Christelle Gastaldi-Ménager, Yves JuillÈre, and Philippe Tuppin. Patient stratification for risk of readmission due to heart failure by using nationwide administrative data. *Journal of Cardiac Failure*, 27(3):266–276, 2021.

David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

Laurence De Roquefeuil, A Studer, A. Neumann, and Y. Merlière. L'échantillon généraliste de bénéficiaires : représentativité, portée et limites. *Pratiques et Organisation des Soins*, 40(3): 213–223, 2009. ISSN 1952-9201. doi: 10.3917/pos.403.0213. URL https://www.cairn.info/revue-pratiques-et-organisation-des-soins-2009-3-page-213.htm.

Núria Farré, Emili Vela, Montse Clèries, Montse Bustins, Miguel Cainzos-Achirica, Cristina Enjuanes, Pedro Moliner, Sonia Ruiz, José María Verdú-Rotellar, and Josep Comín-Colet. Real world heart failure epidemiology and outcome: A population-based analysis of 88,195 patients. *PLOS ONE*, 12(2):e0172745, February 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0172745. URL https://dx.plos.org/10.1371/journal.pone.0172745.

Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982.

Frank E Harrell, Kerry L Lee, Robert M Califf, David B Pryor, and Robert A Rosati. Regression modelling strategies for improved prognostic prediction. *Statistics in medicine*, 3(2):143–152, 1984.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37 (1):388–427, 2021.

Torsten Hothorn, Peter Bühlmann, Sandrine Dudoit, Annette Molinaro, and Mark J Van Der Laan. Survival ensembles. *Biostatistics*, 7(3):355–373, 2006.

Shuhua Hu. Akaike information criterion. *Center for Research in Scientific Computation*, 93:42, 2007.

Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860, September 2008. ISSN 1932-6157, 1941-7330. doi: 10.1214/08-AOAS169. URL https://projecteuclid.org/journals/annals-of-applied-statistics/volume-2/issue-3/Random-survival-forests/10.1214/08-AOAS169.full.

Jian Pei, Jiawei Han, B. Mortazavi-Asl, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu. PrefixSpan,: mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings 17th International Conference on Data Engineering*, pp. 215–224, Heidelberg, Germany, 2001. IEEE Comput. Soc. ISBN 978-0-7695-1001-9. doi: 10.1109/ICDE.2001.914830. URL http://ieeexplore.ieee.org/document/914830/.

Ji-Soo Kang, Ji-Won Baek, and Kyungyong Chung. Prefixspan based pattern mining using time sliding weight from streaming data. *IEEE Access*, 8:124833–124844, 2020.

Noor Kamal Kaur, Usvir Kaur, and Dheerendra Singh. K-medoid clustering algorithm-a review. *Int. J. Comput. Appl. Technol*, 1(1):42–45, 2014.

Judith Lambert, Anne-Louise Leutenegger, Anne-Sophie Jannot, and Anaïs Baudot. Tracking clusters of patients over time enables extracting information from medico-administrative databases. *Journal of Biomedical Informatics*, 139:104309, 2023.

Aleda M Leis, Erin McSpadden, Hannah E Segaloff, Adam S Lauring, Caroline Cheng, Joshua G Petrie, Lois E Lamerato, Manish Patel, Brendan Flannery, Jill Ferdinands, et al. K-medoids clustering of hospital admission characteristics to classify severity of influenza virus infection. *Influenza and Other Respiratory Viruses*, 17(3):e13120, 2023.

Nizar R Mabroukeh and Christie I Ezeife. A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys (CSUR)*, 43(1):1–41, 2010.

Iain L MacDonald and Walter Zucchini. *Hidden Markov and other models for discrete-valued time series*, volume 110. CRC Press, 1997.

Michael Mannino, Joel Fredrickson, Farnoush Banaei-Kashani, Iris Linck, and Raghda Alqurashi Raghda. Development and evaluation of a similarity measure for medical event sequences. *ACM Transactions on Management Information Systems (TMIS)*, 8(2-3):1–26, 2017.

Florent Masseglia, Maguelonne Teisseire, and Pascal Poncelet. Extraction de motifs séquentiels. Problèmes et méthodes. *Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information*, 9(3/4):183, 2004. doi: 10.3166/isi.9.3-4.183-210. URL `https://hal-lirmm.ccsd.cnrs.fr/lirmm-00108563`.

Juliette Murris, Anais Charles-Nelson, Abir Tadmouri Sellier, Audrey Lavenu, and Sandrine Katsahian. Towards filling the gaps around recurrent events in high dimensional framework: a systematic literature review and application. *Biostatistics & Epidemiology*, 7(1):e2283650, 2023.

Dang Nguyen, Wei Luo, Svetha Venkatesh, and Dinh Phung. Effective identification of similar patients through sequential matching over icd code embedding. *Journal of medical systems*, 42: 1–13, 2018.

Jian Pei, Jiawei Han, B. Mortazavi-Asl, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu. PrefixSpan,: mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings 17th International Conference on Data Engineering*, pp. 215–224, April 2001. doi: 10.1109/ICDE.2001.914830. ISSN: 1063-6382.

Jessica Pinaire. *Explorer les trajectoires de patients via les bases médico-économiques : application à l'infarctus du myocarde*. phdthesis, Université Montpellier, October 2017. URL `https://theses.hal.science/tel-01903477`.

Jessica Pinaire, Jérôme Azé, Sandra Bringay, and Paul Landais. Patient healthcare trajectory. an essential monitoring tool: a systematic review. *Health information science and systems*, 5:1–18, 2017.

Gianluigi Savarese, Peter Moritz Becher, Lars H Lund, Petar Seferovic, Giuseppe MC Rosano, and Andrew JS Coats. Global burden of heart failure: a comprehensive and updated review of epidemiology. *Cardiovascular research*, 118(17):3272–3287, 2022.

Joanne Simpson, Pardeep S Jhund, Lars H Lund, Sandosh Padmanabhan, Brian L Claggett, LI Shen, Mark C Petrie, William T Abraham, Akshay S Desai, Kenneth Dickstein, et al. Prognostic models derived in paradigm-hf and validated in atmosphere and the swedish heart failure registry to predict mortality and morbidity in chronic heart failure. *JAMA cardiology*, 5(4):432–441, 2020.

Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, January 1996. ISSN 00359246. doi: 10.1111/j.2517-6161.1996.tb02080.x. URL `https://onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1996.tb02080.x`.

Zhenya Wu and Jianping Hao. A maintenance task similarity-based prior elicitation method for bayesian maintainability demonstration. *Mathematical Problems in Engineering*, 2020:1–19, 2020.

Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.

Haodi Zhong, Grigorios Loukides, and Solon P Pissis. Clustering demographics and sequences of diagnosis codes. *IEEE Journal of Biomedical and Health Informatics*, 26(5):2351–2359, 2021.

Xudong Zhu, Joseph M Plasek, Chunlei Tang, Wasim Al-Assad, Zhikun Zhang, Yun Xiong, Liqin Wang, Sharmitha Yerneni, Carlos Ortega, Min-Jeoung Kang, et al. Embedding, aligning and reconstructing clinical notes to explore sepsis. *BMC Research Notes*, 14:1–6, 2021.

A Appendix

A.1 Sequential pattern mining

A.1.1 Definitions

Itemset. Let $I = i_1, ..., i_{N_p}$ be the set of N_p items. A subset of I is called an itemset. In this study, a pattern or itemset consists in an ICD-10 code.

Event sequence. An event sequence $seq = \{e_1, ..., e_m\}, e_i \subseteq I$ for $1 \leq i \leq m$ is an ordered list of itemsets. The event sequence database is the starting point for sequential pattern mining.

Subsequence. An event sequence $seq_{sub} = \{r_1, ..., r_q\}$ is a subsequence of seq if there exist integers $1 \leq i_1 \leq ... \leq i_q \leq m$, s. t. $r_1 \subseteq e_{i_1}, ..., r_q \subseteq e_{i_q}$.

Support. Let $B = \{seq_1, ..., seq_{N_p}\}$ a set of sequences. The support $Freq_B(seq)$ of a sequence seq is the number of sequences in B that have seq as a subsequence. The higher the support, the more frequently the pattern occurs in the database.

Frequent sequential pattern. An event sequence seq is frequent and called a frequent sequential pattern if its support is greater than or equal to a minimum threshold $\sigma : Freq_B(seq) \geq \sigma$.

A.1.2 Algorithm

The *PrefixSpan* algorithm was retained and uses the concept of ”prefixes” to efficiently search for frequent patterns in a sequence database . The algorithm works by first identifying all frequent single-item sequences, and then iteratively extending these prefixes to form longer sequential patterns.

The algorithm starts with an empty prefix and the entire dataset as the initial projected database. It then recursively explores and extends the prefixes while checking the support of the generated sequences. Frequent sequences above the minimum support threshold are output, and the process continues until no more frequent sequences can be found.

Algorithm 1 The *PrefixSpan* algorithm

```
1. Initialize: Start with an empty set of frequent patterns $I_0 = \emptyset$
2. Frequent Items: For each item $i$ in the first sequence $seq_1$ of the input data, create a singleton pattern
3. Generate Sequences: For each frequent item $i$ found in step 2, extend the current prefix sequence by adding that item to the set of frequent patterns
4. Recursive Search: For each new sequence created in step 3, repeat steps 2 and 3 recursively.
   For each pattern:
    (a) Construct a database of all sequences that contain the pattern as a subsequence
    (b) For each item that appears after the last item of the pattern in the input data, create a new pattern by extending the pattern with the item
    (c) Compute the support of the new pattern by concatenating the support of the item with the support of the database
    (d) If the new pattern is frequent in the database, add it to the set of frequent patterns and continue the recursive search
```

PrefixSpan algorithm is known for its efficiency and scalability, particularly for mining long sequential patterns (Mabroukeh & Ezeife, 2010). Of note, several sequential pattern mining algorithms were experimented, like *APriori* (Al-Maolegi & Arkok, 2014). Similar support for patterns were found but computing time was much higher. This is in line with existing literature in terms of run-time and memory usage (Jian Pei et al., 2001).

Table 1: Most occurring patterns in health care pathway for each patient clusters

Cluster	**Count**	**Freq.**	**Top1 pattern (len1)**	**Count**	**Freq.**	**Top1 pattern (len2)**	**Count**	**Freq.**	**Top1 pattern (len3)**
1	833	0.610	[’Death’]	507	0.371	[’05M09’, ’Death’]	301	0.220	[’05M09’, ’05M09’, ’Death’]
2	3467	0.685	[’Death’]	1542	0.305	[’05M09’, ’Death’]	336	0.066	[’05M09’, ’05M09’, ’Death’]
3	22	0.629	[’23M20’]	15	0.429	[’23M20’, ’23M20’]	8	0.229	[’23M20’, ’23M20’, ’23M20’]
4	2082	0.651	[’Death’]	1111	0.347	[’05M09’, ’Death’]	487	0.152	[’05M09’, ’05M09’, ’Death’]
5	224	0.579	[’05M09’]	132	0.341	[’05M09’, ’05M09’]	75	0.194	[’05M09’, ’05M09’, ’05M09’]

Table 2: Second most occurring patterns in health care pathway for each patient clusters

Cluster	**Count**	**Freq.**	**Top2 pattern (len1)**	**Count**	**Freq.**	**Top2 pattern (len2)**	**Count**	**Freq.**	**Top2 pattern (len3)**
1	777	0.569	[’05M09’]	431	0.316	[’05M09’, ’05M09’]	256	0.187	[’05M09’, ’05M09’, ’05M09’]
2	2032	0.401	[’05M09’]	489	0.097	[’**04M05**’, ’Death’]	104	0.021	[’05M09’, ’05M09’, ’05M09’]
3	18	0.514	[’05M09’]	10	0.286	[’23M20’, ’16M11’]	8	0.229	[’23M20’, ’23M20’, ’23M20’]
4	1586	0.496	[’05M09’]	642	0.201	[’05M09’, ’05M09’]	285	0.089	[’05M09’, ’05M09’, ’05M09’]
5	223	0.576	[’Death’]	129	0.333	[’05M09’, ’Death’]	75	0.194	[’05M09’, ’05M09’, ’Death’]

Table 3: Third most occurring patterns in health care pathway for each patient clusters

Cluster	**Count**	**Freq.**	**Top3 pattern (len1)**	**Count**	**Freq.**	**Top3 pattern (len2)**	**Count**	**Freq.**	**Top3 pattern (len3)**
1	456	0.334	[’**05K10**’]	255	0.187	[’**04M05**’, ’Death’]	124	0.091	[’**05K10**’, ’05M09’, ’Death’]
2	615	0.121	[’**04M05**’]	418	0.083	[’05M09’, ’05M09’]	102	0.020	[’02C05’, ’05M09’, ’Death’]
3	13	0.371	[’06M03’]	9	0.257	[’23M20’, ’05M09’]	8	0.229	[’23M20’, ’23M20’, ’23M20’]
4	823	0.257	[’02C05’]	505	0.158	[’**04M05**’, ’Death’]	208	0.065	[’02C05’, ’05M09’, ’Death’]
5	167	0.432	[’**05K10**’]	88	0.227	[’05K10’, ’05M09’]	56	0.145	[’05K10’, ’05M09’, ’05M09’]

Table 4: Proportions of top 10 ICD-10 codes in nth hospitalization after first hospitalization for heart failure - Deceased patients

	0	1	2	3	4	5	6	7	8	9
05M09	1.000	0.169	0.155	0.143	0.131	0.134	0.137	0.124	0.124	0.123
Death		0.165	0.148	0.156	0.143	0.144	0.154	0.156	0.132	0.137
05K10		0.043	0.020	0.021	0.014	0.019	0.017	0.023	0.022	0.015
05M08		0.024	0.017	0.016				0.016		
04M05		0.023	0.028	0.033	0.026	0.030	0.029	0.018	0.023	0.015
23M20		0.019	0.019	0.016	0.022	0.018	0.017	0.020	0.022	0.015
04M13		0.017	0.014	0.016		0.016	0.019	0.018		
02C05		0.016	0.017	0.017	0.016	0.014	0.019	0.020		
16M11		0.015	0.020	0.020	0.019	0.023	0.021	0.023	0.022	0.022
05K06		0.013			0.015	0.016			0.014	
04M20			0.014		0.016	0.019	0.016	0.014	0.017	
06M03				0.015					0.015	
09M05										0.014
10M16							0.019			0.014
11M04										0.018
23M06					0.012					
23Z02									0.014	0.017

A.2 Clustering

A.2.1 Distance matrix

Because we are working with strings of characters instead of numerical values - and we must keep those strings, we cannot encode them into numerical values - we cannot directly apply clustering algorithms on our data. With N the total number of patients, the distance matrix A writes

$$\forall (i,j) \in [1,N], A_{i,j} = \mathcal{D}_P(patient_i, patient_j) \tag{1}$$

Two properties are noted:

- $\forall i \in [1,N], A_{i,i} = 0$,
- $A^T = A \Leftrightarrow \forall (i,j) \in [1,N], A_{i,j} = A_{j,i}$.

A.2.2 K-medoids algorithm

The algorithm proceeds iteratively:

1. Select k random medoids from the dataset;
2. Each data point is assigned to its closest medoid and calculates the total distance between them;
3. Improve the clustering by iteratively swapping one of the medoids with a non-medoid point and recompute the total distance;
4. Whenever the total distance decreases, the swap is accepted and the new point becomes the medoid for the cluster.

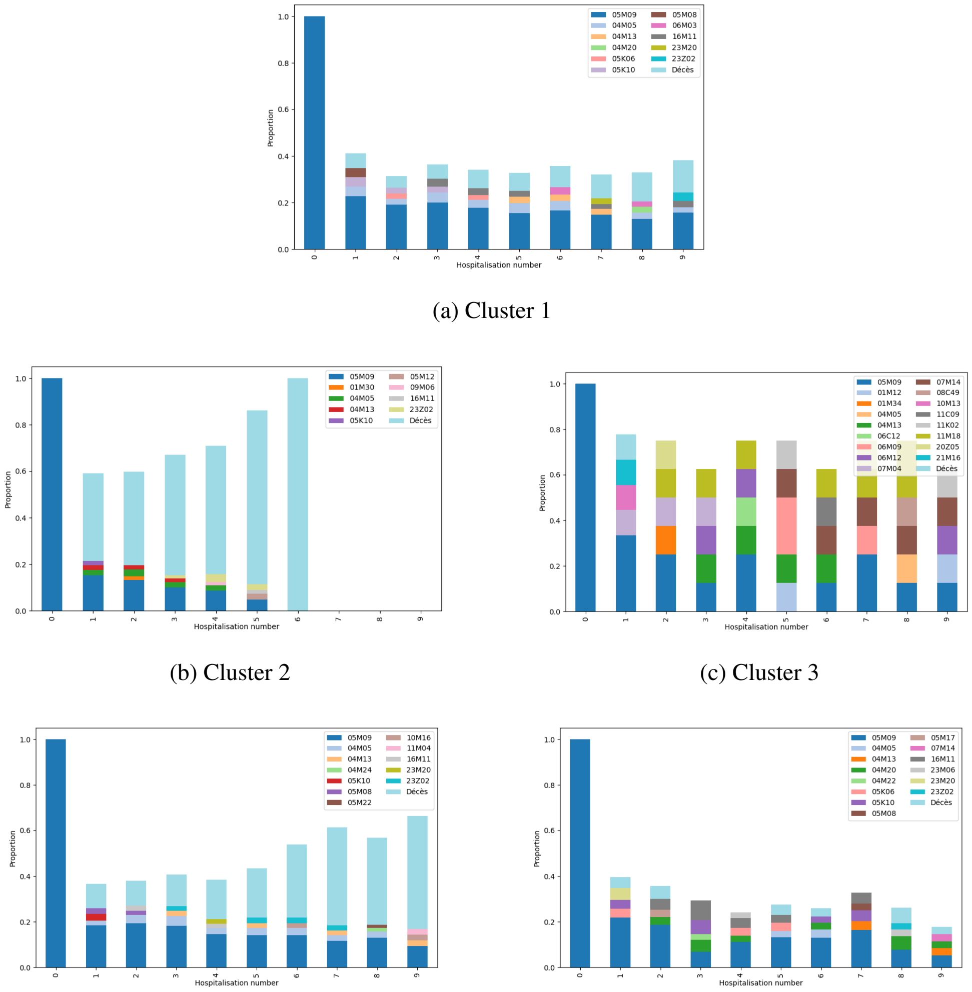
This process is repeated until no further improvement can be made. The pseudo-code is provided below.

Algorithm 2 K-Medoids Algorithm

```
Require: $D$: dataset, $k$: number of clusters
Ensure: $C$: set of clusters, $M$: set of medoids
Initialize $M$ with $k$ random data points from $D$
Assign each data point in $D$ to its closest medoid
Calculate the total distance $TD$ of all data points to their assigned medoids
$change \leftarrow$ true
$iter \leftarrow 1$
while change do
    $change \leftarrow$ false
    for all $m \in M$ do
        for all $p \in D \setminus M$ do
            Swap $m$ with $p$
            Assign each data point in $D$ to its closest medoid
            Calculate the total distance $TD'$ of all data points to their assigned medoids
            if $TD' < TD$ then
                $M \leftarrow$ updated set of medoids
                $C \leftarrow$ updated set of clusters
                $TD \leftarrow TD'$
                $change \leftarrow$ true
            else
                Swap $m$ with $p$                                    ▷ Revert swap
            end if
        end for
    end for
end while
return $C$, $M$
```

A.2.3 Clusters visualization

Except for cluster 3, there were no significant differences amongst clusters. Cluster 3 with only 35 patients, stood out due to a considerably higher number and/or longer duration of hospitalizations compared to other patients. Other clusters included 1,366 patients (mean (sd) age = 78 (12.4) and 44.8% were women), 5,063 patients (mean (sd) age = 83 (13.8) and 55.6% were women), 3,200 patients (mean (sd) age = 81 (12.3) and 47.8% were women), and 387 patients (mean (sd) age = 72 (13.8) and 44.2% were women), respectively.



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

(d) Cluster 4

(e) Cluster 5

Figure 5: Most frequent ICD-10 codes after first hospitalization for heart failure - Deceased patients - Cluster 1 to 5

Here is the Python code to generate the figures above:

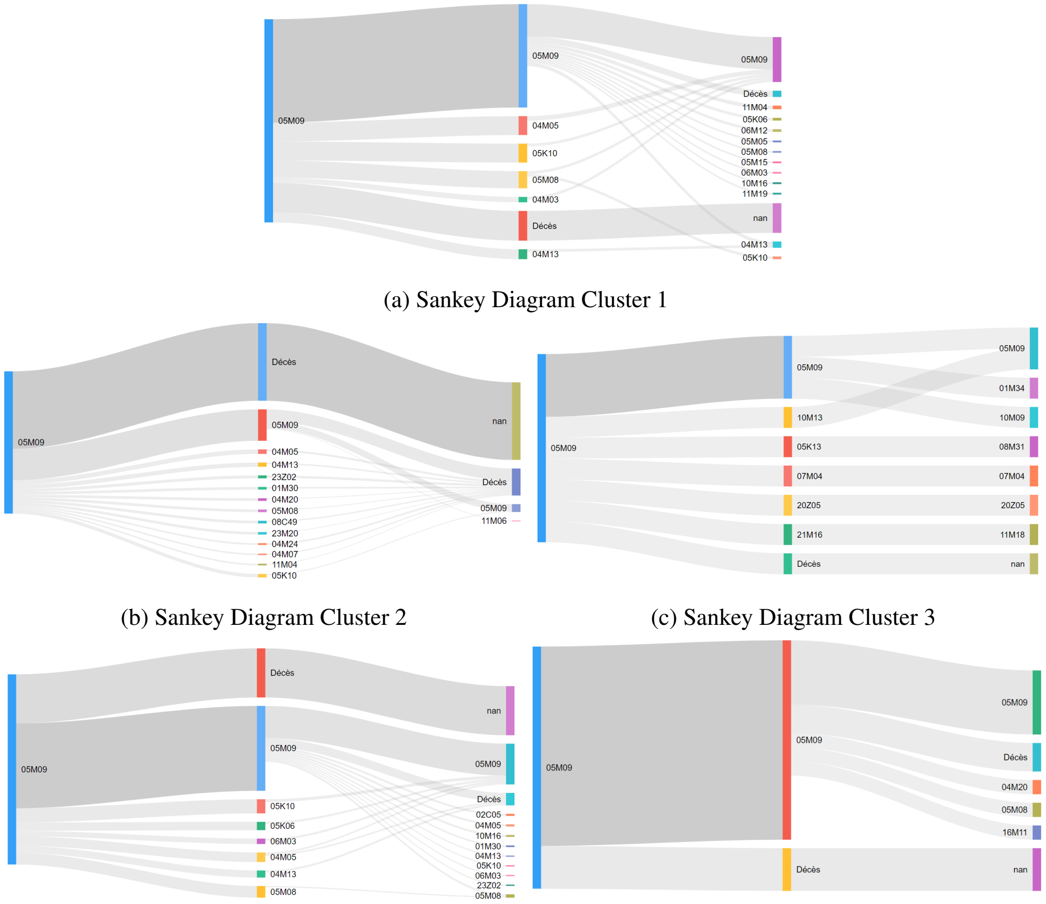
Listing 1: Python code to compute most frequent ICD-10 codes after first hospitalization for heart failure

```
length_care_pathway = 10
top_k = 5 # we'll consider the top 5 most frequent ICD-10 codes
          # for each step in the care pathways

database_top_k_frequent_codes = {}
```

```
for i in range(length_care_pathway):
    # for hospitalization i in all care pathways,
    # consider the 5 most frequent ICD-10 codes found
    # the k most frequent codes per hospitalization step were found previoulsy
    # through with text mining algorithms
    top_k_frequent_codes_i = df.iloc[i].sort_values(ascending=False)[:top_k]
    database_top_k_frequent_codes[f'{i}'] = top_k_frequent_codes_i

database_top_k_frequent_codes = pd.DataFrame(database_top_k_frequent_codes).transpose
database_top_k_frequent_codes.plot(kind='bar', stacked=True));
```

Table 5: Samples of the database_top_k_frequent_codes dataframe

	04M05	04M13	04M24	05K10	05M08	05M09	05M22	10M16	11M04	16M11	23M20	23Z02	Death
0	NaN	NaN	NaN	NaN	NaN	1111.000000	NaN	NaN	NaN	NaN	NaN	NaN	
1	23.000000	NaN	NaN	34.000000	28.000000	204.000000	NaN	NaN	NaN	NaN	NaN	118.000000	
2	37.000000	NaN	NaN	NaN	18.000000	191.000000	NaN	NaN	NaN	23.000000	NaN	NaN	108.000000
3	39.000000	20.000000	NaN	NaN	NaN	160.000000	NaN	NaN	NaN	NaN	NaN	18.000000	124.000000
4	21.000000	NaN	NaN	NaN	NaN	110.000000	NaN	NaN	NaN	15.000000	14.000000	NaN	133.000000



(a) Sankey Diagram Cluster 1

(b) Sankey Diagram Cluster 2

(c) Sankey Diagram Cluster 3

(d) Sankey Diagram Cluster 4

(e) Sankey Diagram Cluster 5

Figure 6: Sankey Diagrams: patient frequent flows at the cluster level

Here is the Python code for the `generateSankey` function using the `prefixspan` library:

Listing 2: Python code for generating Sankey diagrams using PrefixSpan

```
from prefixspan import PrefixSpan

def generateSankey(df_care_pathways, cluster_n, top_k_frequent_codes):
    # df_care_pathways represents the pathways of deceased heart-failure patients
    df_pathways_cluster = df_care_pathways.loc[df_care_pathways.cluster == cluster_n
    nb_patients_cluster_n = len(df_pathways_cluster)
    df_pathways_23 = df_pathways_cluster.astype(str).iloc[:,2:4]
    # keep first and second ICD-10 codes
    df_pathways_34 = df_pathways_cluster.astype(str).iloc[:,3:5]
    # keep second and third ICD-10 codes

    corpus_heart_failures = []
    for patient in range(nb_patients_cluster_n):
        icd_codes = [df_pathways_23.iloc[patient][0], df_pathways_23.iloc[patient][1
        corpus_heart_failures.append(icd_codes)

    ps = PrefixSpan(corpus_heart_failures)
    ps.minlen = 2
    output_corpus = ps.topk(top_k_frequent_codes)

    corpus_heart_failures_2 = []
    for patient in range(nb_patients_cluster_n):
        if str(df_pathways_34.iloc[patient][1])!=’nan’:
            icd_codes = [df_pathways_34.iloc[patient][0], df_pathways_34.iloc[patien
            corpus_heart_failures_2.append(icd_codes)
        else:
            corpus_heart_failures_2.append([df_pathways_34.iloc[patient][0], ’nan’])

    ps_corpus_2 = PrefixSpan(corpus_heart_failures_2)
    ps_corpus_2.minlen = 2
    output_corpus_2 = ps_corpus_2.topk(top_k_frequent_codes)

    return output_corpus, output_corpus_2
```

The outputs of the function `generateSankey` above are then used with HTML code to create the clean Sankey outputs.

We also evaluated the clusters by calculating the minimum distance between the ICD-10 code of each data point and the ICD-10 codes of the cluster's medoid (see Figure 7). The shading in the graphs indicates the extent to which the ICD-10 code deviates from the hospitalization pattern of the patient medoid. Lighter areas, particularly in clusters 1 and 5, suggest that the data points within these clusters exhibit relatively close similarity to one another.

A.3 SURVIVAL ANALYSIS

We define the survival function

$$S(t) = \mathbb{P}[T > t] = 1 - F(t) = \int_t^\infty f(u)du$$

with f the density and F the distribution function. T is the time of occurrence of some specific event, time to death in our case. We introduce censoring C to the labels, whenever the patient died or not with $T^* = \min(T, C)$ and $\Delta = \nVdash_{\{T \leq C\}}$.

A convenient way of modeling relationships between T and explanatory features X is the hazard function. Let $\lambda(t|X)$ denote the hazard function associated with the distribution of T given X.

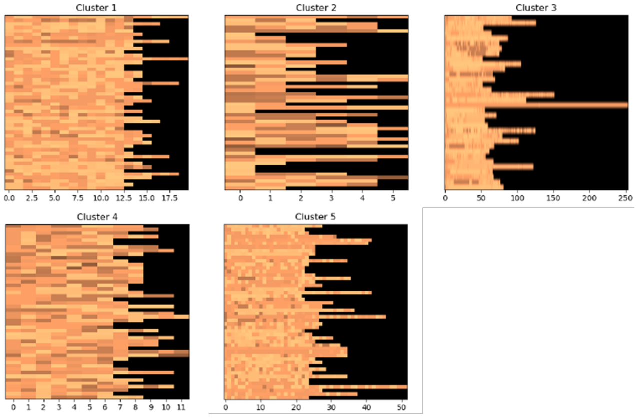


Figure 7: Distance of ICD-10 codes with the medoid for each cluster

The Cox proportional hazard (CPH) is the most common approach to model survival events (Cox, 1972). It writes

$$\lambda(t|X) = \lambda_0^*(t)\exp(f^*(X))$$

λ_0^* is the baseline hazard function and usually $f^*(X) = X^T\beta$ with $\beta \in \mathbb{R}^d$.

This model assumes hazards to be proportional, i.e. hazard ratios to be independent of time as

$$\frac{\lambda_i(t|X_i = x_i)}{\lambda_j(t|X_j = x_j)} = \exp(f^*(x_i) - f^*(x_j))$$

This approach can be penalized and/or include splines to handle non-linear relations (Tibshirani, 1996). This semi-parametric approach was applied to the data, however its main assumptions were never met. For this reason ensemble methods were selected.

Here is the Python code for the `Survival random forest` function using the `scikit survival` library:

Listing 3: Python code for generating the Grid Search over the number of estimator in our survival random forest

```
data=profil_patient[[’y_nais’,’BEN_SEX_COD’,’Nb_hospit’,’CHOC’,’Nb_jours_sej’]]
Label= profil_patient[[’Mort’,’Nb_survie’]].to_records(index=False)

X_train, X_test, y_train, y_test = train_test_split(data, Label, test_size=0.25)
Liste_score = []
Liste_nb_estimators= []
rsf = RandomSurvivalForest(min_samples_split=10,
                           min_samples_leaf=15,
                           n_jobs=-1,
                           random_state=random_state,
                           verbose=0)
for i in range(1, 20):
    n_estimators = i * 1
    rsf.set_params(n_estimators=n_estimators)
    rsf.fit(X_train, y_train)
    Liste_score.append(rsf.score(X_test, y_test))
    Liste_nb_estimators.append(n_estimators)
    print(rsf.score(X_test, y_test))


plt.plot(Liste_nb_estimators, Liste_score)
plt.xlabel(””)
plt.ylabel(””)
plt.grid(True)
```

Table 6: Evaluation for each cluster

Cluster	**AIC**	**C-Index**
1	8,495	0.612
2	41,748	0.640
3	39	0.887
4	24,015	0.621
5	1773	0.543

A.4 Clinical discussion

The cohort of this study was comparable to other similar studies, namely in terms of age and gender, and with poor prognosis with high rate of hospitalization and mortality (Farré et al., 2017). The results obtained from this work may thus be comparable to those of other similar groups. For this reason, age and being male being strongly associated with mortality were expected findings (Simpson et al., 2020).

Nevertheless, our study is subject to several limitations. First, the inclusion of primary diagnoses alongside the principal diagnoses in hospitalizations could provide a more comprehensive view of patients' medical histories, enabling a deeper understanding of their healthcare trajectories. Unfortunately, acquiring such data presents a challenge, as only a limited subset of hospitalizations in our dataset contains this information. Consequently, our analysis is constrained to ICD-10 codes. Besides, the sequential pattern mining analysis uncovered certain high-risk patterns that could potentially serve as additional features in our survival model, similar to our inclusion of cardiac shock. Integrating these factors into our model would enable to quantify the excess mortality associated with these risky healthcare sequences. We believe that this additional information could enhance the precision of our predictions and yield a more nuanced understanding of the underlying causes of mortality among heart failure patients.