Applied Statistics Group

Subject n°22

Text mining and care pathway: what are the causes of mortality in heart failure patients?

Report

Final deliverable

Keywords: Heart failure / GHM / Care pathway / Clustering / Pattern Mining / Cox Model / Survival Random Forest

Students (ENSAE) :
Tristan Amadei
Tristan Kirscher
Antoine Klein

Coordinator (CREST) :
Dr. Roxana Fernandez

Support (AP-HP) :
Dr. Anne-Isabelle Tropeano
Juliette Murris (PhD C)

July 5, 2023

Contents

List of Figures

List of Tables

0 Access to the appendix

At the end of the paper is the appendix which contains additional information regarding our work. We reference this appendix several times throughout our paper, thus we have added clickable hyperlinks that can take the reader directly to the referenced object; and back to the section one was reading before going to the appendix.
Those links are highlighted in yellow, the following way: *appendix* for instance.

1 Context and objective of the project

Heart failure is a disease of the cardiovascular system characterized by an inability of the heart to pump enough blood to meet the body's oxygen and nutrient needs. In France, more than 1.5 million people suffer from this disease, mainly people over 60 years old[9]. Heart failure has a significant impact on patients' quality of life and results in nearly 200,000 hospitalizations each year[4].

The objective of the study is to clarify the causes of mortality in heart failure patients, who are increasingly older. Knowing the main causes of mortality in these patients and their most frequent care pathways will have a major public health impact.

To do this, we have data extracted from the EGB (*Echantillon Généraliste des Bénéficiaires*, random sample representative of 1/97th of the population with a follow-up of at least 2 years, from the French health insurance databases).

To answer our question, we first characterize the care pathways of patients through the study of sequential patterns: using GHM codes (*Groupes Homogènes de Malades*) defining hospitalizations, it is possible to find similarities in the care pathways, associated with a diagnosis.

Once these pathways are identified, a survival analysis will predict the survival trajectory after first hospitalization.

2 Description of the data

2.1 Data source

The data are separated into three databases, containing information on 24,311 patients, their hospitalizations, and their consumption (social security). The collection period is from 2010 to 2016. Their attributes are described in the Figure 1.

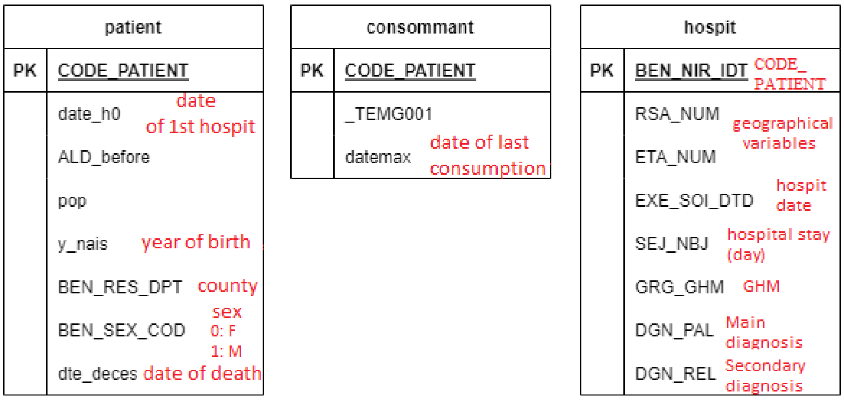


Figure 1: Relationship diagram of the entities of the different databases

2.2 Identification and description of the population

Some patients did not meet the inclusion criteria of our study which are :

- Patients have to be adult
- Patients have to be hospitalized into our window of interest (2010 - 2016)
- Patients must not have already been hospitalized prior to this study

We therefore proceeded to a selection stage, which also enabled us to remove chronic hospitalizations, which were too frequent in our work. A flowchart, which can be found *in the annex*, makes light of all the steps the database went through during its cleaning. At the end of this data cleaning phase, we were left with:

- 10,051 patients
- 85,594 hospitalizations
- 9,985 consumptions

We also anonymized the database, by *pseudonymizing* the patient codes and by keeping only the month and year of the dates of death, for ethical reasons.
A visualization of the age distribution gives:

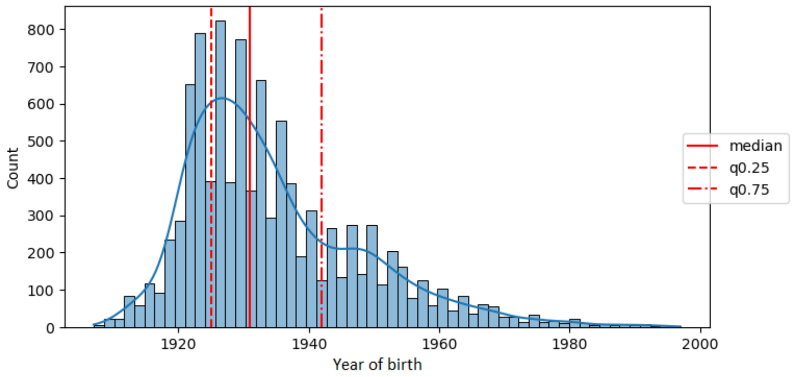


Figure 2: Distribution of the years of birth of our population

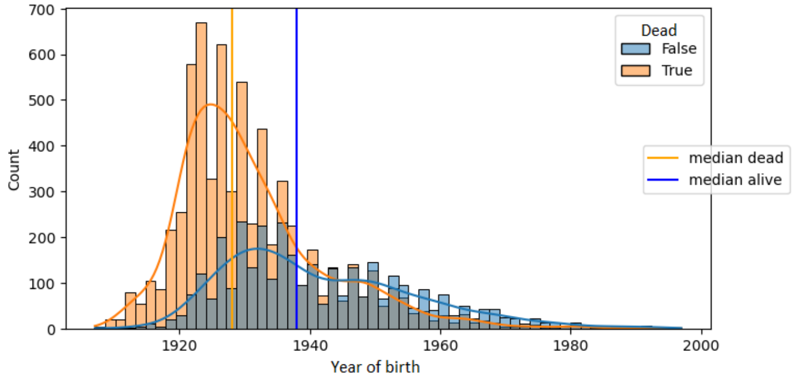


Figure 3: Distribution of the years of birth of our population between dead and alive

We observe an older cohort with a concentration of births around the 1920s-1930s. An overview of the descriptive statistics on the patient population is provided in Table 1 below.

Description of our covariables			
Statistics	**Year of birth**	**Gender (M=1/W=2)**	**Nb of days of survival after admission**
Count	10 051	10 051	10 051
Mean	1935.6	1.5	1130.8
Standard Error	13.5	0.50	951.3
Min	1907	1	0
25%	1925	1	171
50%	1931	2	1101
75%	1942	2	1836
Max	1997	2	3384

Table 1: Descriptive statistics on the population

As gender can be an explanatory variable, it is important to work on a parity study. This seems to be the case in our population: women are only slightly over-represented, as can be seen in Figure 4 below.

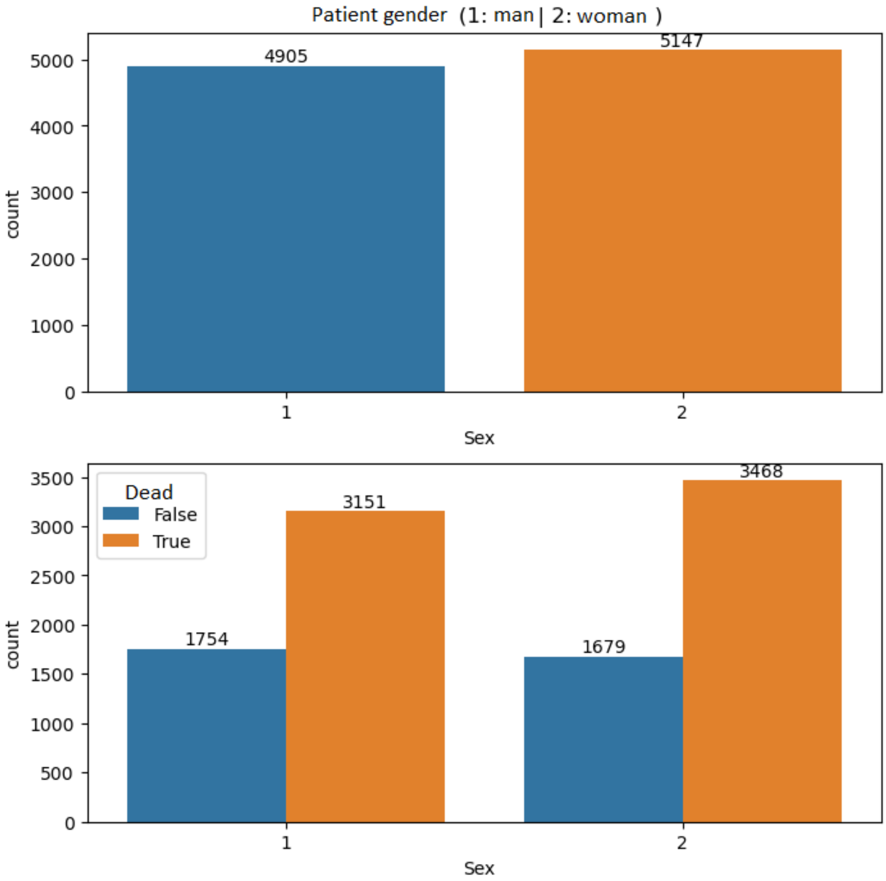


Figure 4: Patient gender, then stratified by whether they died or not

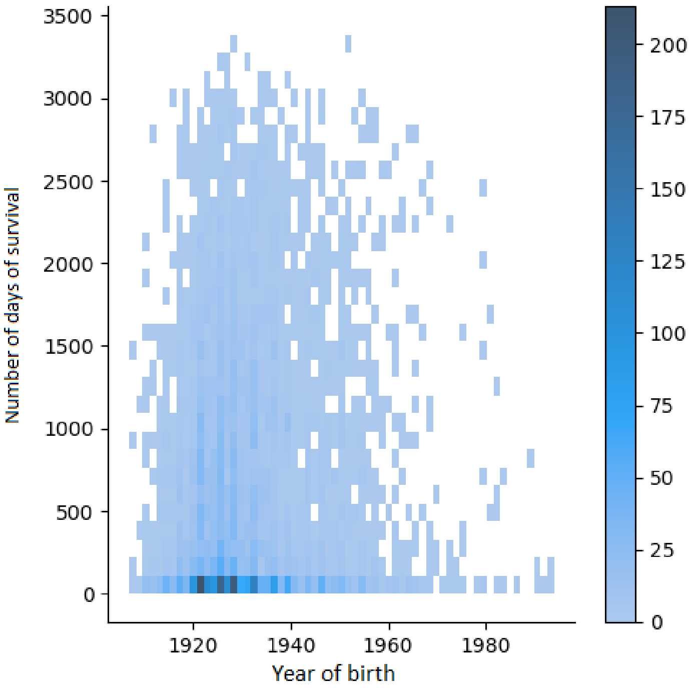


Figure 5: Survival trajectory of our population according to age

We see on Figure 5 above that elderly patients tend to die shortly after their first hospitalization. If this does not tell us anything about causality, we will remember that the study time window is appropriate.

2.3 GHM Code

As introduced in section 1, we will be working with GHM codes which characterize the different hospitalizations patients undergo.
The syntax of a GHM code is very insightful regarding one's hospitalization. It takes the following form: 05M092, which, in this case, would be the code for a Heart failure and circulatory shock, level 2.
A GHM code can be divided as follows:

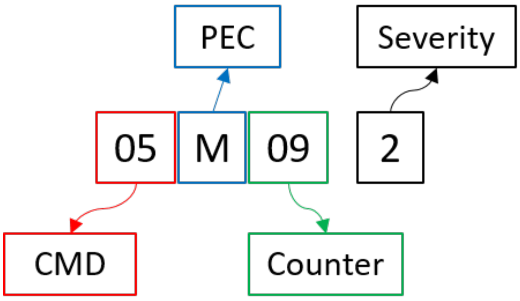


Figure 6: Dividing the GHM code into meaningful parts

- CMD : principal diagnosis category, ranging from 01 to 28

- PEC : $\begin{cases} \text{M : surgery performed in an operating room} \\ \text{K : surgery not performed in an operating room} \\ \text{C : no surgery} \end{cases}$

- Counter : identifier to distinguish different codes with the same starting architecture

- : Severity : $\begin{cases} \{1, 2, 3, 4\} \text{ : indicates the increasing severity of the hospitalization} \\ \text{E : patient is deceased} \\ \text{J : ambulatory surgery} \\ \text{Z : non-segmented} \\ \text{T : very short duration} \end{cases}$

Hence, the syntax of this code is very important and brings information that must not be discarded. We will now look at the care trajectory of each patient, i.e. the sequences of their successive hospitalizations defined by their GHM.

3 Patients clustering

We will now look at the care pathways of each patient, *i.e.* their successive hospitalization sequences, represented by their GHM codes.
In order to obtain results more easily interpretable, and that would more easily respect the hypotheses of models that will be used later on, we decided to split the pool of patients into clusters. We drew our inspiration from a similar work conducted in Jessica Pinaire's thesis *"Exploring Trajectories of Patients via Medico-Economic Databases: Application to Myocardial Infarction."*[15]. Nonetheless, we wanted these clusters to closely fit the patients' hospital journeys, and we wanted not to take into account anything other than the care pathways: not the gender, not the age, etc. Thus we chose to use **unsupervised learning** to establish those clusters, instead of using information on the patients in order to create fixed clusters.

CODE_PATIENT	hospit_course
P0	05K051,05M042,05M16T,05M09T,05M092,05C191,05M20Z,05M20Z
P6	02C05J,02C05J,05M093,04M132,05C222,23M103,04M053
P8	11M041,06C194

Table 2: Excerpt of the health care pathways database

As you can see in Table 2 above, patients are only represented by their health care pathways, with each hospitalization being represented by its GHM code; note that we do not take into account the date of these hospitalizations, all we care about is their order with regards to the patient who underwent them.

3.1 Distance metric

As explained in the section *2.3 GHM Code*, the syntax of the GHM codes is important; we thus had to keep it unaltered, and find a way to cluster patients based on this information.
To implement our clustering algorithm, we thus needed to establish our own distance metric, that would compute the distance between two patients' hospitalization courses.
Let's first define the ***Levenshtein distance***[10], which we'll use within our own distance function. This function computes the distance between two strings of characters, using the following formula:

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

where

- |a| = number of letters in the word a
- $a_{1:}$ = word a without its first letter (we consider the words to be 0-indexed)
- a[0] = first letter of a

In other words, the Levenshtein distance calculates the minimum number of edits to perform on single-characters to transform the word a into the word b.

Actually, we want to use the Levenshtein ratio instead of the distance, we then just have to normalize this distance, as follows:

$$lev(a,b) = \frac{lev\_dist(a,b)}{max(|a|,|b|)} = \frac{lev\_dist(a,b)}{|a|} \tag{2}$$

Indeed, throughout our study, we will only use the Levenshtein ratio with words of the same number of characters, thus we will always have $max(|a|,|b|) = |a| = |b|$.
As you will see below, we will end up summing several Levenshtein ratios to calculate the distance between GHM codes; thus, normalizing this Levenshtein distance allows for the GHM distance not to be too important.
We then used this string-distance function to implement our own metric to compute the distance between two GHM codes:

$$\begin{aligned} ghm\_distance(a,b) = w_1 \cdot lev(a_{0:2}, b_{0:2}) + w_2 \cdot lev(a[2], b[2]) \\ + w_3 \cdot lev(a_{3:5}, b_{3:5}) + w_4 \cdot lev(a[5], b[5]) \end{aligned} \tag{3}$$

where

- $(w_i)_{i \in [\![1,4]\!]} \in \mathbb{N}_+^4$ are weights to determine
- $a_{i:j}$ represents the word formed by keeping the letters of a between i (included) and j (excluded)
- $a[i]$ represents the i-th letter of a

Let's take an example to get a better idea of how this function works. Let's set hypothetical weights : $\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \end{pmatrix}$, and a = 05M092; b = 05K051.

Note that a and b are strings of characters.

$$\begin{aligned} ghm\_distance(a,b) &= 4 \cdot lev(05,05) + 3 \cdot lev(M,K) + 2 \cdot lev(09,05) + 1 \cdot lev(2,1) \\ &= 4 \cdot 0 + 3 \cdot \frac{1}{1} + 2 \cdot \frac{1}{2} + 1 \cdot \frac{1}{1} = 5 \end{aligned}$$

We can now leverage this function that computes the distance between GHM codes to calculate the distance between health care pathways.
Our first idea was to go through the two hospitalization courses we wanted to compare, and sum up the distances GHM per GHM. However, we ran into a phase-shift problem: say you have two patients that have undergone the exact same hospitalizations, in the exact same order, except the first patient has had a first hospitalization before all this. Then we would like those two patients to be within the same cluster, for they are quite similar; but using this summation technique, the distance between those patients might actually be quite important because of this shift of but one hospitalization.
The solution we came up with to address this issue was to implement a sort of filter: the idea was to compare the i-th GHM code of patient 1 with the (i-1)-th, i-th and (i+1)-th GHM codes of patient 2, compute these distances and keep the minimum.
Let's understand it through an example: let's take the first two patients and their health care pathways in the table at the very beginning of *3 Patients clustering*. We want to compute this filter to get the distance associated to the third GHM code of the first patient (colored in red on Figure 7 below).

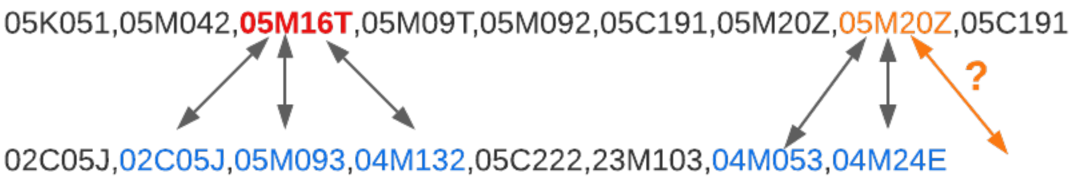

05K051,05M042,05M16T,05M09T,05M092,05C191,05M20Z,05M20Z,05C191
?
02C05J,02C05J,05M093,04M132,05C222,23M103,04M053,04M24E

Figure 7: Distance of 1 GHM using the filter approach

$$\text{We compute the three distances} \begin{cases} ghm\_distance(\text{05M16T, 02C05J}) \\ ghm\_distance(\text{05M16T, 05M093}) \\ ghm\_distance(\text{05M16T, 04M132}) \end{cases}$$

and we keep the minimum of these three values. Let's call the function that calculates this distance $ghm\_filter\_distance(patient_1, patient_2, index_{ghm})$.
Let's note the fact that, when a GHM code does not have three counterparts in the other hospitalization course, as it is the case with the orange GHM code on the figure, we compute the missing distances as the maximum distance, $i.e$ the sum of the weights of the model. The idea behind this choice lies on the fact that patients that have a significant difference in terms of number of hospitalizations must be considered as not alike; thus we want the distance between those two patients to be large enough for them not to be in the same cluster.

We can finally write a function that uses all the previous results to calculate the distance between the health care pathways of two patients:

$$\begin{aligned} patient\_distance(patient_1, patient_2) = & \frac{1}{2}\sum_{i=0}^{N_1-1} ghm\_filter\_distance(patient_1, patient_2, i) \\ & + \frac{1}{2}\sum_{i=0}^{N_2-1} ghm\_filter\_distance(patient_2, patient_1, i) \end{aligned} \quad (4)$$

where

- N_1 : number of GHM codes in the health care pathway of patient 1
- N_2 : number of GHM codes in the health care pathway of patient 2

The idea is the following:

1. we compute the sum of the distances (using the filter approach) of the GHM codes of patient$_1$ with regards to patient$_2$
2. we do the same for patient$_2$ with regards to patient$_1$
3. we compute the mean of these two sums

The idea behind this approach was to respect the symmetry condition:

$$patient\_distance(patient_1, patient_2) = patient\_distance(patient_2, patient_1)$$

3.2 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.
The first step was then to calculate the distance matrix A, such that

$$\forall (i, j) \in [\![1, N]\!], A_{i,j} = patient\_distance(patient_i, patient_j)$$

with N the total number of patients.
We can note two properties of this matrix, that can be directly deduced from the distance function:

- $\forall i \in [\![1, N]\!], A_{i,i} = 0$
- the matrix is symmetric: $A^T = A \Leftrightarrow \forall i \in [\![1, N]\!], A_{i,j} = A_{j,i}$

Hence, we only need to compute the distance of the lower triangular part of the matrix. This is quite important because our metric is fairly lengthy to compute, this then allows us to halve the total computational time from around 3h down to 1h30min.
However, this still takes an unsatisfactory long time to compute: indeed, as this distance matrix is dependent on the weights that act as hyperparameters in this context, we will need to compute it with a lot of different weights to try and pick the best. Hence, we still need to significantly reduce this computational time.
The idea we came up with was to parallelize our calculations, using multiprocessing in Python. In our working environment, when running Python code, we had 72 CPU cores, thus we divided the calculation of this distance matrix into 72 processes that would run at the same time. Furthermore, as we would compute only the lower triangular matrix, calculating the first rows of said matrix was dramatically faster than calculating the latter ones. Hence, when dividing the rows to compute into our different processes, we would decrease the number of rows to compute in one process and the indices of those rows were increasing.
The results were quite formidable: computing this distance matrix would now only take up to 6-7 minutes on average, as opposed to the 3 hours originally required.

3.3 Clustering algorithm

Everything is now finally set and ready for us to implement our clustering of patients. Because we are working with string data, it is not possible for us to apply PCA to visualize it. Hence, it was rather hard to choose the clustering algorithm.
Because we could not visualize the data, we decided not to try and work with density-based clustering techniques, nor with spectral clustering approaches. Hierarchical clustering could not apply to a dataset of patients; thus we were left with partition-based clustering techniques. K-Means[11] could not be used as it is based on the euclidean distance because it computes the centroids of the clusters rather than the medoids; which **K-Medoids**[13] does, hence why we chose to work with this algorithm.
An explanation can be found in the section ***The K-Medoids Algorithm*** of the appendix, explaining what is the K-Medoids algorithm and how it works.

3.4 Hyperparameters

3.4.1 Clusters assessment

If you remember, in section *3.1 Distance metric*, our metric was based on hyperparameters, namely the weights $W = (w_1, w_2, w_3, w_4)^T$. Furthermore, our clustering algorithm must be given a fixed number of clusters, which in our case, will also represent a hyperparameter.
Our strategy was to use cross-validation to find the optimal hyperparameters. Hence, we had to find a way to assess clusters returned by our K-Medoids algorithm, given some hyperparameters.
Let's take a step back and remind ourselves why we want to clusterize our data. The goal behind clustering the data is to consider, within each cluster, patients that have followed similar health care pathways, and on the contrary, consider patients from different clusters to have followed quite different ones. In this context, we want the frequency of GHM codes within a cluster to be greater than the frequency of the same GHM code within the whole dataset.
Using the *PrefixSpan algorithm* that will be explained later on, for each cluster:

1. we calculate the patterns of length {1, 2, 3} of GHM codes that appear most often, as well as their frequencies within the cluster
2. we calculate the frequency of those patterns but taking into account the whole dataset
3. we compute the subtraction of the frequency within cluster minus the frequency in the whole dataset

We are now left with 3 values for each cluster, stored in a dataframe. Let's show a hypothetical example of this dataframe in Table 3 below:

cluster	diff_length_1	diff_length_2	diff_length_3
1	0.000012	0.000007	0.000005
2	0.000009	0.000007	0.000001
3	0.000006	0.000005	0.000001

Table 3: Difference between frequency within cluster minus the frequency in the whole dataset for GHM codes

Finally, we calculate the mean of these differences of length 1, the one for differences of length 2 and the one for length 3. And we now get the score attributed to this clustering by computing the mean of these means.

3.4.2 Cross-Validation

We can now use cross-validation to find the best hyperparameters for our problem. We had a lot of different possible combinations, as we wanted to test:

- $0 \leq w_4 \leq w_3 \leq w_2 \leq w_1 \leq 100$
- number of clusters $\in [\![2, 20]\!]$

As the distance matrix is quite long to compute, we decided to use the ***Optuna library***[1] to converge more quickly towards the optimal hyperparameters.
The goal was to find the set of hyperparameters that would maximize the score obtained in the previous section.
The results we obtained were the following:

- $W^T = [85, 75, 55, 40]$
- 5 clusters

3.5 Clusters visualization

Everything is now set up for us to implement our clustering algorithm with our custom metric, and with optimal hyperparameters. Let's now visualize our clustered population.

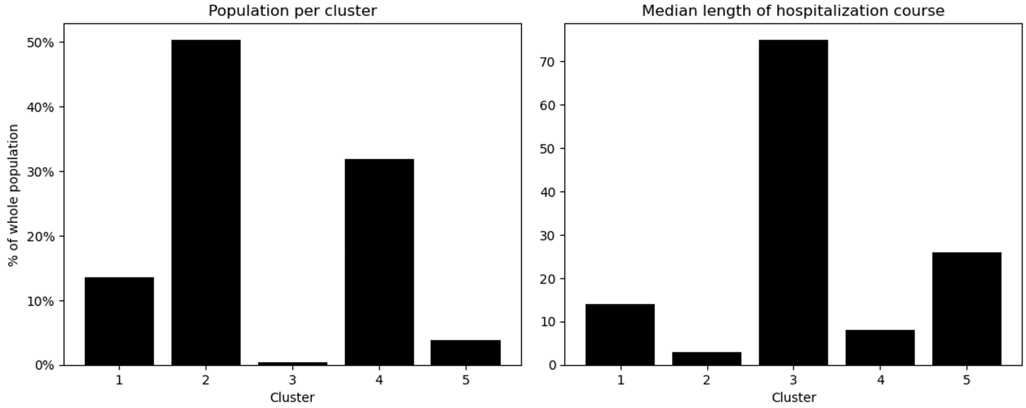


Figure 8: Population and median length of hospitalization course per cluster

What is interesting here is the cluster 3. There are only 35 patients in this cluster, and they seem to have been through considerably more hospitalizations than the other patients of the database. Thus, it is actually good news - and a testatement of sort that our clustering was successful - that these outliers were set aside in a cluster of their own.
If we take a look at other features, there does not seem to be any significant difference with the initial dataset; which shows that clusters, indeed, did get calculated from the hopitalization courses, and those courses only. The visualization of said features are to be found *in the appendix.*

We may also try and visually assess the clusters by the distance between points of each cluster and their medoid. For each cluster:

1. we select 50 patients randomly
2. for each patient
    - for each GHM code in the health care pathway, we check if this code appears in the hospitalization course of the patient medoid of the cluster

- if so, the code will be displayed in green in the figure below
- otherwise, it will be displayed in grey

As we can see on the figure below, it is very interesting to notice that there are quite a good amount of green spots, and it is quite reassuring. However, the cutoff is quite brutal, as we are looking for the presence of GHM codes instead of trying to compute a distance.

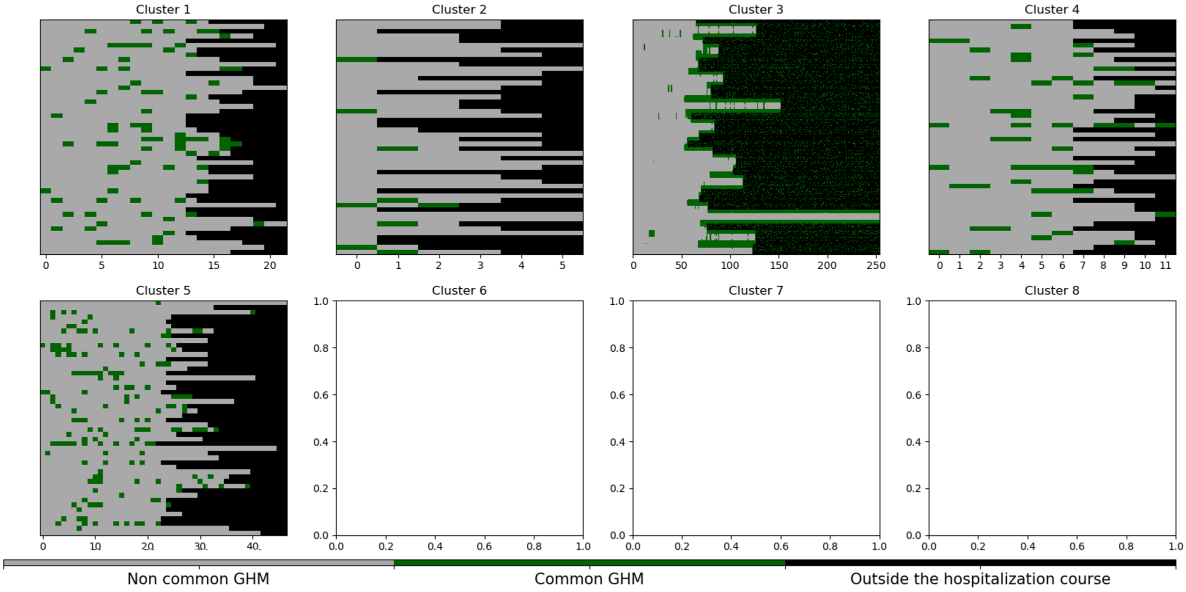


Figure 9: GHM codes in common with the medoid - per cluster

We changed the previous approach to cope with the issue of the brutal cutoff. We followed the same approach, but instead of checking the presence of each GHM code, we calculate the minimum distance between this GHM code and all the GHM codes of the medoid.

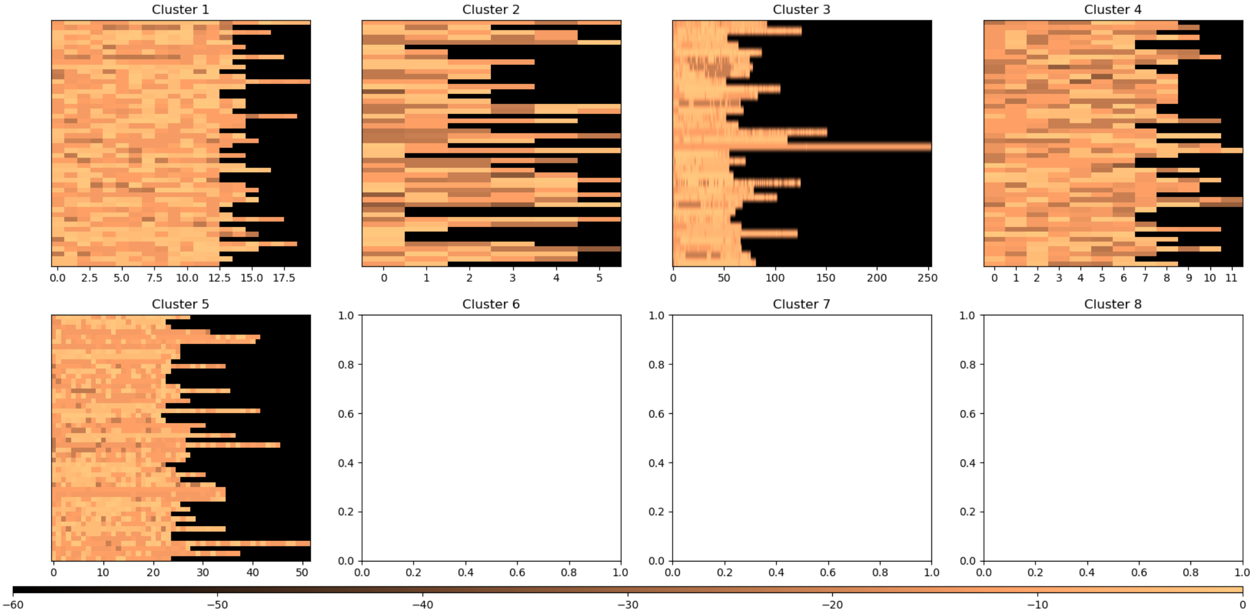


Figure 10: Distance of GHM codes with the medoid - per cluster

The darker it gets, the further the GHM code is from the hospitalization course of the patient medoid. Again, especially when it comes to the clusters 1 and 5, the graphs are quite light meaning the clusters contain datapoints that are quite close to one another.
Let's note that the visualization of the cluster 3 for the two previous figures is different from the others, but it is not so important as this cluster is composed of the outliers of the dataset.

4 Health care pathways analysis

The aim of this section is to extract frequent health care pathway patterns, risky, or even potentially lethal trajectories. Their identification will improve future predictions, as well as trying to better understand the causes of death in heart failure patients. We will notably use the clusters defined previously in section 3 to see if patterns emerge according to patient groups.

The methodology and steps pursued in the following section are inspired by a similar work conducted in Jessica Pinaire's thesis *"Exploring Trajectories of Patients via Medico-Economic Databases: Application to Myocardial Infarction."*[15]. By applying the same framework to our new research question, we hope to contribute to the broader understanding of the factors that cause the death of heart failure patients.

4.1 Sequential Pattern Mining

4.1.1 Preliminary definitions

Sequential Pattern Mining (SPM) is a data mining technique used to discover frequently occurring sequential patterns or subsequences in a sequence database or time-series data. It involves analyzing a collection of sequences to discover patterns that frequently occur together. The goal of sequential pattern mining is to find patterns that occur in a specific order and with a certain frequency. Here is an overview of the key components of SPM:

Definition 1 (Itemset)**.** Let $I = i_1, i_2, ..., i_k$ be the set of all items. A subset of I is called an *itemset.*

In this study, a pattern or itemset consists in a GHM, see subsection 2.3.

Definition 2 (Event Sequence)**.** An *event sequence* $s =< e_1, e_2, ..., e_m >$ is an ordered list of itemsets, where $e_i \subseteq I$ for $1 \leq i \leq m$.

The event sequence database is the starting point for sequential pattern mining. It is defined in the setup of this study in the section 4.1.2.

Definition 3 (Subsequence)**.** An event sequence $s_0 =< r_1, r_2, ..., r_p >$ is a *subsequence* of $s =< e_1, e_2, ..., e_m >$ if there exist integers $1 \leq i_1 \leq i_2 \leq ... \leq i_p \leq m$ such that $r_1 \subseteq e_{i_1}$, $r_2 \subseteq e_{i_2}$, ..., $r_p \subseteq e_{i_p}$.

Definition 4 (Support)**.** Let $B = s_1, ..., s_n$ be a database of sequences. The *support* of a sequence s, $Freq_B(s)$, is the number of sequences in B that have s as a subsequence.

The higher the support, the more frequently the pattern occurs in the database.

Definition 5 (Frequent Sequential Pattern)**.** An event sequence s is *frequent* and called a *frequent sequential pattern* if its support is greater than or equal to a minimum threshold $k\sigma > 0$: $Freq_B(s) \geq k\sigma$.

4.1.2 Event sequence database

In order to analyze patients health care pathways, we built a dataset containing, for each patient, their care pathway in terms of GHM, by combining the *hospitalization* and *patient* databases. In this dataset, which is divided into relative timestamps corresponding to the occurrence of hospitalization, time is perceived as a discontinuous variable. For example, t_1, the date of the second hospitalization, is different for P1 (patient 1) and P2.

The example from Table 4 is taken from this dataset and corresponds to the health care pathway of patient P6. The corresponding sequence database is : ['02C05J', '05M093', '04M132', '05C222', '23M103', '04M053', '04M24E']. Their second hospitalization is associated with GHM *05M093* for *Cardiac failure and circulatory shock, level 3*. However, in order not to create too many singular patterns and for interpretability reasons, we decided, after discussion with the medical staff, to truncate the GHMs in such a way as to remove the severity indicator. The latter example database thus becomes: ['02C05', '05M09', '04M13', '05C22', '23M10', '04M05', '04M24'].

GHM	**02C05**	**05M09**	**04M13**	**05C22**	**23M10**	**04M05**	**04M24**
Hospitalization n°	0	1	2	3	4	5	6
Timestep	t_0	t_1	t_2	t_3	t_4	t_5	t_6

Table 4: Explanation of the sequence database for the patient P6 with corresponding timesteps and hospitalization numbers

Furthermore, in order to have a sequential database that also contains the death event for deceased patients, it is necessary to add it manually. Since the data in the hospitalization database only includes the GHMs of the different hospitalizations, we make a join with the patient database to retrieve this information. To do so, we add the event 'Décès' at the end of the care pathway of patients for whom the variable 'Mort' (deceased) is true.

4.1.3 Frequent Sequential Patterns in health care pathways

Frequent patterns are being extracted from health care pathways using a *sequential pattern mining* algorithm. To do so, we experimented several SPM algorithms. Ultimately, the *PrefixSpan*[5] algorithm was retained and we computed frequencies for each of the patient clusters identified in Section 3. The functioning of the algorithm as well as the motivation for its choice are detailed in the *APPENDIX* .

We thus obtain a dataset containing, for each patients clusters, the most frequent patterns for different temporal stamp lengths (length 1, length 2, etc.). The tables below provide the frequencies and counts of the most frequent patterns of lengths 1, 2, and 3 for the entire population (Table 5) and for the five clusters (Table 7, Table 8, and Table 9 in the *APPENDIX*).

Top k	**Count**	**Freq.**	**Top1 pattern (len1)**	**Count**	**Freq.**	**Top1 pattern (len2)**	**Count**	**Freq.**	**Top1 pattern (len3)**
k=1	6618	0.658	['Décès']	3298	0.328	['05M09', 'Décès']	1206	0.120	['05M09', '05M09', 'Décès']
k=2	4637	0.461	['05M09']	1632	0.162	['05M09', '05M09']	728	0.072	['05M09', '05M09', '05M09']
k=3	1897	0.189	['05K10']	1331	0.132	['04M05', 'Décès']	374	0.037	['04M05', '05M09', 'Décès']

Table 5: First, second, and third most occurring patterns in health care pathway for all patients

Some remarks on results of Table 5 (whole population):

- 6618 health care pathways contain the unique event 'Décès'. This is to be expected since we identified in subsection 2.2 that 6618 patients of our dataset died.
- 4637 health care pathways contain the GHM '05M09'. That is to say 46% of the population had at least one hospitalization for heart failure.

- The patterns ['05M09'], ['05M09', '05M09'] and ['05M09', '05M09', '05M09'] are omnipresent. Patients with heart failure therefore tend to have repeated hospitalizations for this reason. It is also the GHM most frequently associated with the 'Décès' (death) event. The sequences ['05M09', 'Death'] (support=0.328) and ['05M09', '05M09', 'Death'] (support=0.12) are therefore among the most frequent patterns.
- Hospitalization for heart failure ('05M09') is the last hospitalization before death in 33% of care paths.
- Another GHM is highlighted in the analysis of frequent patterns: '04M05' for *Pneumonias and common pleurisy*, with the sequence ['04M05', 'Décès'] (support=0.132). This means that we might be able to identify causes of mortality for heart failure patients other than heart failure itself. This is the purpose of the following.

Let us now study frequent patterns in more detail by exploring them at the the patient clusters (developed in section 3) level. A subset of these results can be found in Table 7, Table 8 and Table 9. We will not dwell on the results for cluster 3, because we have previously explained that these were outliers for which the outcomes cannot be interpreted.

Most of the frequent GHMs of the overall population previously found appear again, but it is interesting to see that a new GHM are discovered in the sequences with the 'Décès' (death) event: **05K10** (diagnostic procedures using vascular route). This is shown in Table 9, especially for clusters 1 and 5. However, it is difficult to draw conclusions for our question from the presence of this GHM in frequent sequences, because it is a gesture proposed for example to explain a decompensation / heart failure episode.

4.2 Heart failure patients healthcare trajectory visualization

In this section, the aim is to highlight the trajectories of heart failure patients to visualize patient flows through the different care pathways. In order to focus on the causes of death of patients with heart failure and to avoid having sequential databases that are too long and difficult to interpret, the trajectory is studied from the first hospitalization for heart failure. That is to say, the hospitalization at t_0 described in the subsubsection 4.1.2 corresponds to a GHM '05M09' for every patient of the population studied here (cf. the proportion of 1.0 of GHM '05M09' for the first hospitalization at t_0 in Table 10).

4.2.1 Visualization of frequent spatio-temporal patterns

The purpose of this section is to give us an overview of the similarity of care pathways for heart failure patients.

In Figure 11, we represent for each successive hospitalization, the 10 most frequent corresponding GHMs (in proportion, for the nth hospitalization after first heart failure episode). It is therefore a visualization of the results of the Table 10. As explained previously, the relative time axis is fixed here with the first hospitalization for heart failure at t_0, hence the proportion of 1.0 for '05M09' in hospitalization 0.

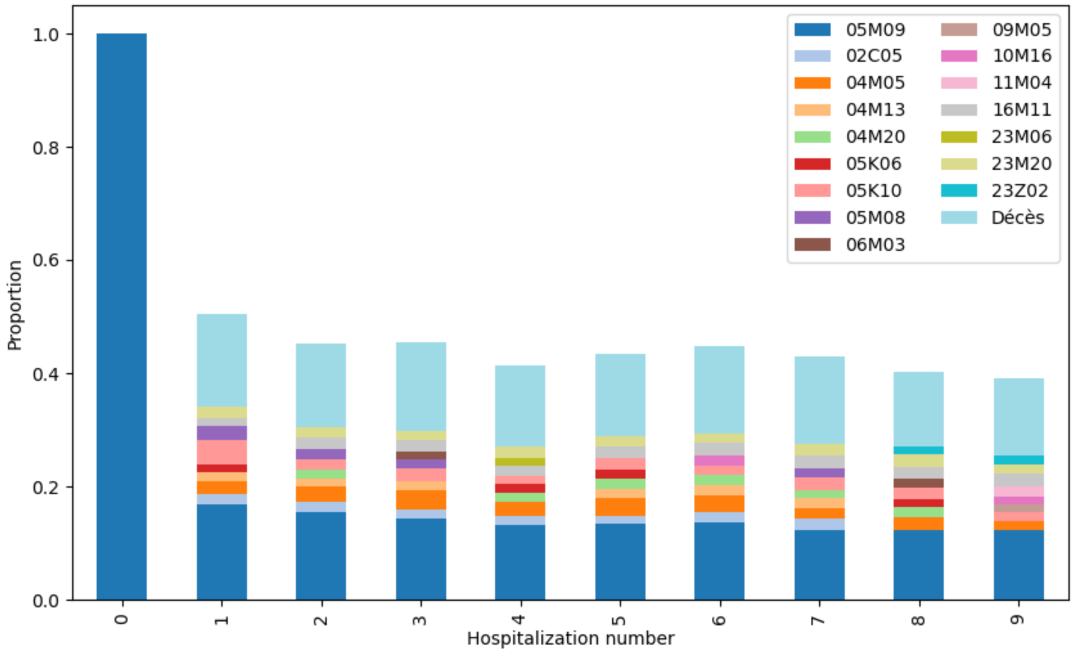


Figure 11: Most frequent GHMs after first hospitalization for heart failure - Deceased patients

If we consider the 10 most frequent GHMs for the nth hospitalization after an episode of heart failure, we notice that they represent nearly 50% of hospitalizations in proportion for the hospitalization 1 to 10. There are therefore great similarities in the care pathways of these patients.

Setting aside '05M09' heart failure and deaths events, the following GHMs can thus be highlighted:

- **04M05**: Pneumonias and common pleurisy.
- **04M13**: Pulmonary edema and respiratory distress.
- **05K10**: Diagnostic procedures using vascular route
- **23M20**: Other symptoms and reasons for seeking medical care under CMD 23
- **16M11**: Other disorders of erythrocyte lineage
- **05M08**: Arrhythmias and cardiac conduction disorders
- **04M20**: Chronic bronchopneumopathies with superinfection

At the cluster level, the graphs in the Figure 17 in the *APPENDIX* allow us to see the proportions of the GHMs listed previously and locate them temporally in the sequence of hospitalizations.

For instance, we had observed in the section 3 that clusters 2 and 4 were those with the oldest patients and shortest care paths : we therefore find very high proportions of deaths in the first hospitalizations. Patients with heart failure who died in cluster 2 had no more than 6 hospitalizations after their first episode of heart failure.

4.2.2 Sankey diagrams of frequent patient flows

The interest of this visualization is that it allows, by combining it with the results of the analysis of frequent patterns, to keep the temporality. That is to say, we can easily see the patterns that are upstream of the 'Décès' (death) event, which can give some insights to identify the causes of mortality of heart failure patients.

If we want to visualize all the patients' trajectories, we get too many edges and the graph quickly becomes unreadable. To remedy this problem, we display only the frequent patterns for the patients of the group studied.

We notice that the extraction of frequent patterns on our sequential database leads to very low support values (order of magnitude of 0.01), and in particular on subpopulations within clusters. Even if these patterns are considered frequent from a SPM point of view, we lose a lot of significance and we must be very careful with these results. In this perspective, we limit ourselves to trajectories of length 3, a length beyond which the supports are far too weak to be significant.

The Sankey diagrams (Figure 12 below and Figure 18 in the *APPENDIX*) are derived from interactive .html plots that can be downloaded on the *GitHub* repository of our project (see Appendix E). In particular, we can see for each link the proportion (in relation to the size of the cluster) and the number of patients concerned by hovering with the mouse. The value 'nan' has been added to facilitate the creation of the Sankey. This is a zero value at the end of the care pathway after the patients death. It is used to differentiate the time index of death ('Décès' at t_1, t_2 or t_3). Without it, all deaths would be added to the last level of the graph. Thus, all the death events point to a 'nan' value.

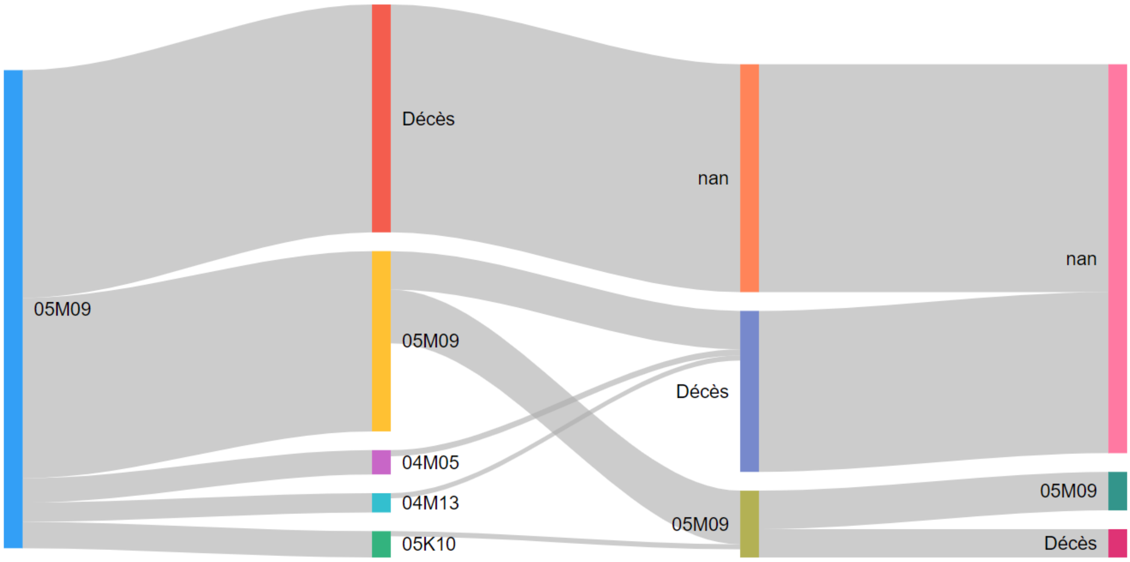


Figure 12: Sankey Diagram whole population

By visualizing the frequent trajectories for the entire heart failure patient population, we can identify several frequent trajectories leading to death: a succession of hospitalizations for heart failure ('05M09'), hospitalization for *pleurisy* ('04M05') or for *Pulmonary edema and respiratory distress* ('04M13') after the first heart failure.

As before, frequent trajectories leading to patient death are also identified for each of the clusters in Figure 18. There is no evidence of new frequent GHM preceding patient death. However, it is interesting to observe the different trajectories structures and proportions at the cluster level.

5 Survival analysis

5.1 Goal of the survival analysis

As we now have a clustering for the patients and insights of their health care pathway, we implement a survival analysis to obtain survival prediction.

Definition 6 (Censored data)**.** Censoring is a condition in which the value of a measurement or observation is only partially known.

Definition 7 (Survival Model)**.** A model is called survival if it contains censored data.

In our case, we have for each patient, in addition to different control variables, their survival time and their dead or alive status. The survival time is then **right-censored** by the study window. The objective is to estimate, for a cluster of patients, the **probability of survival over time**. The model must be **interpretable** with statistical guarantees since its predictions will have clinical implications. We have chosen the library *Lifelines* [3] and its **Cox model**. The latter allows us to obtain p-values [6] and confidence intervals. Wanting to see the marginal impact of having or not a cardiac shock in one's care pathway, we decided to **stratify** our study. We note CHOC the boolean variable that indicates whether the patient has experienced cardiac shock or not.

5.2 Checking the Cox model assumption

Before fitting a Cox model, we need to verify that the **hypothesis** assumed by the model are relevant.

5.2.1 Log-rank test

The first assumption leads to the **distribution** of our dataset. The distribution of appearance of death has to be significantly different between each cluster. To test this, we run a *log-rank test* [19]. It compares estimates of the **hazard functions** of the two groups at each observed event time. We obtain for the inter-clusters :

Log-rank test on our dataset : target =cluster					
H0: (i,j) have different hazard functions					
p-value test for cluster (i with j)	1	2	3	4	5
1	<0.05	>0.05	>0.05	>0.05	>0.05
2	>0.05	<0.05	>0.05	>0.05	>0.05
3	>0.05	>0.05	<0.05	>0.05	>0.05
4	>0.05	>0.05	>0.05	<0.05	>0.05
5	>0.05	>0.05	>0.05	>0.05	<0.05

Thus, at the level 5%, we can't reject that the distribution of one cluster is different from an another. Our clustering is **relevant** for the survival analysis.

We repeat the same process for the covariable cardiac shock. CHOC value refers to a care pathway which contains a cardiac shock whereas NO CHOC refers to one with none of them.

Log-rank test on our dataset : target = CHOC		
H0: (i,j) have different hazard functions		
p-value test (i with j)	CHOC	NO CHOC
CHOC	<0.05	>0.05
NO CHOC	>0.05	<0.05

Thus, at the level 5%, we can't reject that the distribution of care pathway with a cardiac showk is different from the others with none of them. The presence of cardiac shock within a care pathway is **significantly relevant** in term of survival.

We then repeat the process for the gender variable.

Log-rank test on our dataset : target = Gender		
H0: (i,j) have different hazard functions		
p-value test (i with j)	MEN	WOMEN
MEN	<0.05	>0.10
WOMEN	>0.10	<0.05

Thus, at the level 10%, we can't reject that the distribution of men is different from women. If it is **less significant** than the others, it seems that gender will have an impact of the survival trajectory.

Conclusion of this part: It is **legitimate** to perform a survival study **stratified** by our covariates.

5.2.2 No autocorrelation in the residuals

Another hypothesis made by the Cox model is the fact that covariates should have a **constant impact over time** on survival. This means that the residuals of our model must behave **like white noise**. We therefore test the absence of autocorrelation of our residuals through a Ljung-Box test [18].

Without stratification on clusters, a Cox model fit on the whole population give the results :

Ljung-Box test on our dataset		
H0: Absence of autocorrelation in our residuals		
Features	p-value	Wish
AGE	0.02	>0.05
GENDER	0.36	>0.05
CHOC	10e-36	>0.05
Average number of days in hospital	10e-9	>0.05

Thus, at the level 5%, only the gender respects the assumptions of the Cox model.

5.2.3 Assumption of proportional hazards

The last assumption made by the Cox model is the **proportionality of the hazards**.

Definition 8 (Cox Hazard function)**.**

$$\forall i\ \forall t \in \mathbf{R},\ \lambda(t|X_i) = \lambda_0(t)\exp(X_i * \beta)$$

Definition 9 (Proportionality)**.**

$$\forall i\ \forall j\ \forall t \in \mathbf{R},\ \frac{\lambda(t|X_i)}{\lambda(t|X_j)} = \exp((X_i - X_j) * \beta)$$

It means that the ratio of impact of two variables is **independent of time**.

We can test this with an another *log-rank test* [3] on each variable. We obtain without cluster stratification on the whole population :

Log-rank test on our dataset		
H0: Proportionality of the hazard functions		
Features	p-value	Wish
AGE	<0.005	>0.05
GENDER	0.01	>0.05
CHOC	<0.005	>0.05
Average number of days in hospital	<0.005	>0.05

Thus, at the level 5%, we can reject the proportionality of the hazard functions for each features : the Cox Model is **not feasible**.

5.2.4 Significance of the coefficients

To test whether the estimated coefficients are relevant or not, we perform a T-Test :

Student test on our dataset		
H0: Nullity of the coefficients		
Features	p-value	Wish
AGE	<0.005	<0.05
GENDER	<0.005	<0.05
CHOC	<0.005	<0.05
Average number of days in hospital	<0.005	<0.05

Thus, at level 5%, we reject the nullity of each coefficient : there are **all significant**. It illustrates perfectly the importance of checking the assumptions of a model **before** plotting its prediction.

Still, the predicted trajectory of survival is shown there : *APPENDIX*

5.3 Model adjustment to better respect the assumptions

We just saw that we can't fit a Cox model on our dataset because the assumptions are not satisfied. We then go through different solution axes.

5.3.1 Penalisation Cox Model

The first adjustment is to **penalise** our model in order to **reduce over-fitting and take into account multi-correlation of explanatory covariates**. We thus perform a **L1 penalisation** [17]. The same verification process provides :

Verification test on our penalised Cox model						
	Ljung-Box Test		Log-Rank Test		Significance of the coefficients	
	H0: Absence of autocorrelation residuals		H0: Hazard functions proportionality		H0: Nullity of the coefficients	
Features	p-value	Wish	p-value	Wish	p-value	Wish
AGE	0.001	>0.05	<0.005	>0.05	<0.005	<0.05
GENDER	0.48	>0.05	0.43	>0.05	0.07	<0.05
CHOC	1e-45	>0.05	<0.005	>0.05	<0.005	<0.05
Average nb of days	1e-30	>0.05	<0.005	>0.05	<0.005	<0.05

We see an **improve** in term of hazard function proportionality, mostly for the gender variable with the **significance trade-off**. It performs better but it is **still not feasible**.

The predictions are given there : *APPENDIX*

We observe a **smooth** trajectory of survival which is the same for both men and women : only having a shock plays a key role.

5.3.2 Interpolation based on splines

We can combine this penalisation with a **Spline Interpolation**. It enables both the limiting of the over-fitting and the smoothing of the trajectory predicted. The main issue with this approach is the fact that our covariables are **no longer clinical variables** (gender, age, choc, average days in hospital) but instead spline combination of these. In particular, we lost much in **interpretability**. This process provides :

Verification test on our penalised Cox model						
	Ljung-Box Test		Log-Rank Test		Significance of the coefficients	
	H0: Absence of autocorrelation residuals		H0: Hazard functions proportionality		H0: Nullity of the coefficients	
Features	p-value	Wish	p-value	Wish	p-value	Wish
Spline 1	1e-5	>0.05	<0.005	>0.05	<0.005	<0.05
Spline 2	0.92	>0.05	0.48	>0.05	<0.005	<0.05
Spline 3	1e-6	>0.05	<0.005	>0.05	<0.005	<0.05
Spline 4	0.59	>0.05	0.50	>0.05	<0.005	<0.05

We see that Spline 2 and Spline 4 respect all assumption of a Cox Model, we could use them to predict the probability of survival. Yet, after discussing with clinicians, we decided not to because they wanted to **stick with interpretable clinical variable**. They wanted to have a model which can determine the marginal effect and not the effect of a combination of clinical variables.

5.3.3 A stratified Cox Model with our clustering

Our last idea to make it work is to **stratify it with our cluster**. The intuition is pretty simple : our population is very heterogeneous, it is normal not to be able to find a feasible common regression. We will obtain more accurate result if we fit a penalised Cox Model **on each cluster**.

For this section, in order to get a nice way to represent our results, we decided to set to blue all test that respects the assumption and to black when it doesn't.

The results are in the *APPENDIX*

We observe that **none** of our covariates meet all the model assumptions. This indicates an alpha exploration budget that is too low given the size of our dataset and its diversity. This result, which may seem **pessimistic**, also reveals that parametric models do not offer enough degrees of freedom for such a diverse clinical cohort.

Still, the predicted survival trajectory given a cluster is there : *APPENDIX*

From these results, several observations emerge :

- The clusters do not all have the same number of individuals, which means that the accuracy of our **predictions varies** a lot between models. In particular, cluster 3 provides much more uncertain predictions than cluster 5.
- Old age is an **aggravating factor** in mortality for all clusters. What is interesting is to see that this impact is **not of the same order** between clusters. In particular, the elderly in cluster 2 have a much more pessimistic trajectory than those in cluster 1.
- Time spent in hospital is again a sign of a pessimistic trajectory. However, we should not see this as causal: it is those with the most serious pathologies who stay the longest (**selection bias**). Nevertheless, for cluster 3, we notice that the people who stayed for a long time have a better survival trajectory: is this because the care provided allows them to remedy their pathology?

Definition 10 (Concordance Index)**.** The concordance index or C-index is a generalization of the area under the ROC curve (AUC) that can take into account censored data to correctly provide a reliable ranking of the survival times based on the individual risk scores.

Metrics on the test_set				
Clusters	AIC	Wish	Concordance Index	Wish
Cluster **1**	8495	As low as possible	0.612	Near 1
Cluster **2**	41748	As low as possible	0.64	Near 1
Cluster **3**	39	As low as possible	0.887	Near 1
Cluster **4**	24015	As low as possible	0.621	Near 1
Cluster **5**	1773	As low as possible	0.543	Near 1

Table 6: Predictive power for a test_set (20% of the dataset)

We observe **good predictive capabilities** with a mean concordance index of **0.66** , especially since our models are of the same order of power regardless of the cluster.

5.4 Survival Random Forest

We also wanted to compare the predictions of the Cox Model with **non-parametric approaches**: the Survival Random Forest [7] and the Survival Gradient Boosting [12]. They are an extension of the well-known Random Forest and Gradient Boosting for censored data. We use the library Scikit-Survival [16] .

As any usual machine learning algorithm, we have to **optimise hyperparameters** : in our case the depth of the trees. A grid search over this parameter provides a choice of 10, see *APPENDIX*

We thus use Forest of trees with depth 10. We then plot for each cluster the most and least optimistic trajectory:

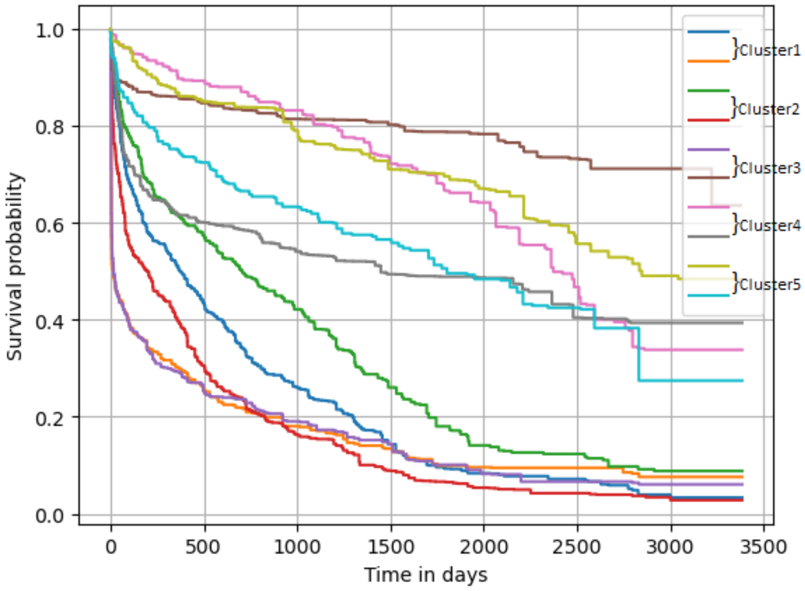


Figure 13: Survival Trajectory for 2 people in each cluster

We obtain **similar trajectories** with the most pessimistic faith for cluster 2 and cluster 4. In term of predictive power, we recover the performance of Cox model : approximately **0.68**.

The same performance goes for Survival Gradient Boosting, see *APPENDIX*

Conclusion of this part : The relaxation of the parametric model **loses explainability** without bringing any predictive power. We will then keep a parametric Cox approach. The latter concludes that age, average length of stay and the presence of cardiac shock are aggravating factors for mortality.

6 Discussion

We have identified several areas for the improvement of our study, which could be the focus of future research. First, the addition the *Diagnostic Principal* (DP) of the hospitalizations could provide more detailed information on the patients' medical history, allowing us to better understand their healthcare pathways. However, obtaining this data is challenging, as only a small portion of hospitalizations in our dataset have DP filled in, this study is restricted to GHMs for this reason. Second, the inclusion of geographic data could help us better understand how patients' location affects their healthcare outcomes. This could include information on the patient's region, neighborhood, or even their proximity to healthcare facilities. Finally, our sequential pattern mining analysis identified certain risky patterns that could potentially be included as covariates in our survival model as we did with cardiac shock. This would allow quantifying the excess mortality of these risky healthcare pathways. We believe that incorporating these factors could improve the accuracy of our predictions and lead to a better understanding of the causes of heart failure patients' death.

Nonetheless, we believe that our study has some strengths that are noteworthy to mention. Firstly, the study population is large from a data standpoint, providing a robust basis for analysis. Additionally, the utilization of unsupervised clustering in our methodology is a notable highlight. Our approach avoids any apriori assumptions about the population, reducing potential biases associated with certain patient characteristics. In comparison, Jessica Pinaire's thesis[15] employs "patient contexts" and defines groups based on age, sex, and number of hospitalizations, which appears to be less robust in our perspective.

7 Conclusion

This study is part of the search for causes of mortality in heart failure patients. We had at our disposal a medical cohort of 10,000 patients followed over several years with their successive GHM. From this dataset, we elaborated a metric adapted to our situation to carry out a relevant clustering. We then retained 5 clusters.

The identification of frequent hospitalization patterns and the visualization of the care pathways of heart failure patients allowed us to highlight frequent hospitalization causes and trajectories leading to patient death. In the end, even if other frequent reasons have been identified, it seems to emerge from this study that heart failure patients die following hospitalizations for this very reason.

Finally, a survival analysis was conducted to establish predictions of death. The selected cohort did not meet all the conditions for a parametric analysis (too much diversity in survival trajectories). However, it provides information on the dynamics according to age, sex, the presence of a cardiac shock, and the average length of stay in hospital. A nonparametric analysis by Survival Random Forest was conducted. The latter did not perform any better than the parametric one, while leading to a loss of interpretability. We will therefore retain the use of Cox models stratified on each cluster.

Thus, this study, which is more methodological than operational, paves the way to the use of GHM Big Data to better understand heart failure.

References

[1] 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 and Data Mining*, 2019.

[2] Mohammed Al-Maolegi and Bassam Arkok. An improved apriori algorithm for association rules. *CoRR*, abs/1403.3948, 2014.

[3] Cameron Davidson-Pilon. lifelines: survival analysis in python. *Journal of Open Source Software*, 4(40):1317, 2019.

[4] Santé Publique France. Insuffisance cardiaque. `https://www.santepubliquefrance.fr/maladies-et-traumatismes/maladies-cardiovasculaires-et-accident-vasculaire-cerebral/insuffisance-cardiaque`. Accessed: 2022-11-02.

[5] Chuancong Gao. *PrefixSpan*, *BIDE*, and *FEAT* in python 3, 2018.

[6] Hossein Hassani and Mohammad Reza Yeganegi. Selecting optimal lag order in ljung–box test. *Physica A: Statistical Mechanics and its Applications*, 541:123700, 2020.

[7] Hemant Ishwaran, Michael S. Lauer, Eugene H. Blackstone, Min Lu, and Udaya B. Kogalur. randomForestSRC: random survival forests vignette, 2021.

[8] Amina Kemmar, Yahia Lebbah, Samir Loudni, Patrice Boizumault, and Thierry Charnois. Prefix-projection global constraint and top-k approach for sequential pattern mining. *Constraints*, 22, 04 2017.

[9] l'Assurance Maladie ameli. Insuffisance cardiaque, définition et causes. `https://www.ameli.fr/assure/sante/themes/insuffisance-cardiaque/definition-causes`. Accessed: 2022-11-02.

[10] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February 1966.

[11] Youguo Li and Haiyan Wu. A clustering method based on k-means algorithm. *Physics Procedia*, 12 2012.

[12] Pei Liu, Bo Fu, and Simon X. Yang. Hitboost: Survival analysis via a multi-output gradient boosting decision tree method. *IEEE Access*, 7:56785–56795, 2019.

[13] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 03 2009.

[14] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. pages 215–224, 02 2001.

[15] Jessica Pinaire. *Exploring trajectories of patients via medico-economic databases : application to myocardial infarction.* PhD thesis, Université de Montpellier, 2017.

[16] Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6, 2020.

[17] J Ranstam and J A Cook. LASSO regression. *British Journal of Surgery*, 105(10):1348–1348, 08 2018.

[18] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[19] Robert F. Woolson. Rank tests and a one-sample logrank test for comparing observed survival data to a standard population. *Biometrics*, 37(4):687–696, 1981.

A Figures

Back to section 0

A.1 Description of the data

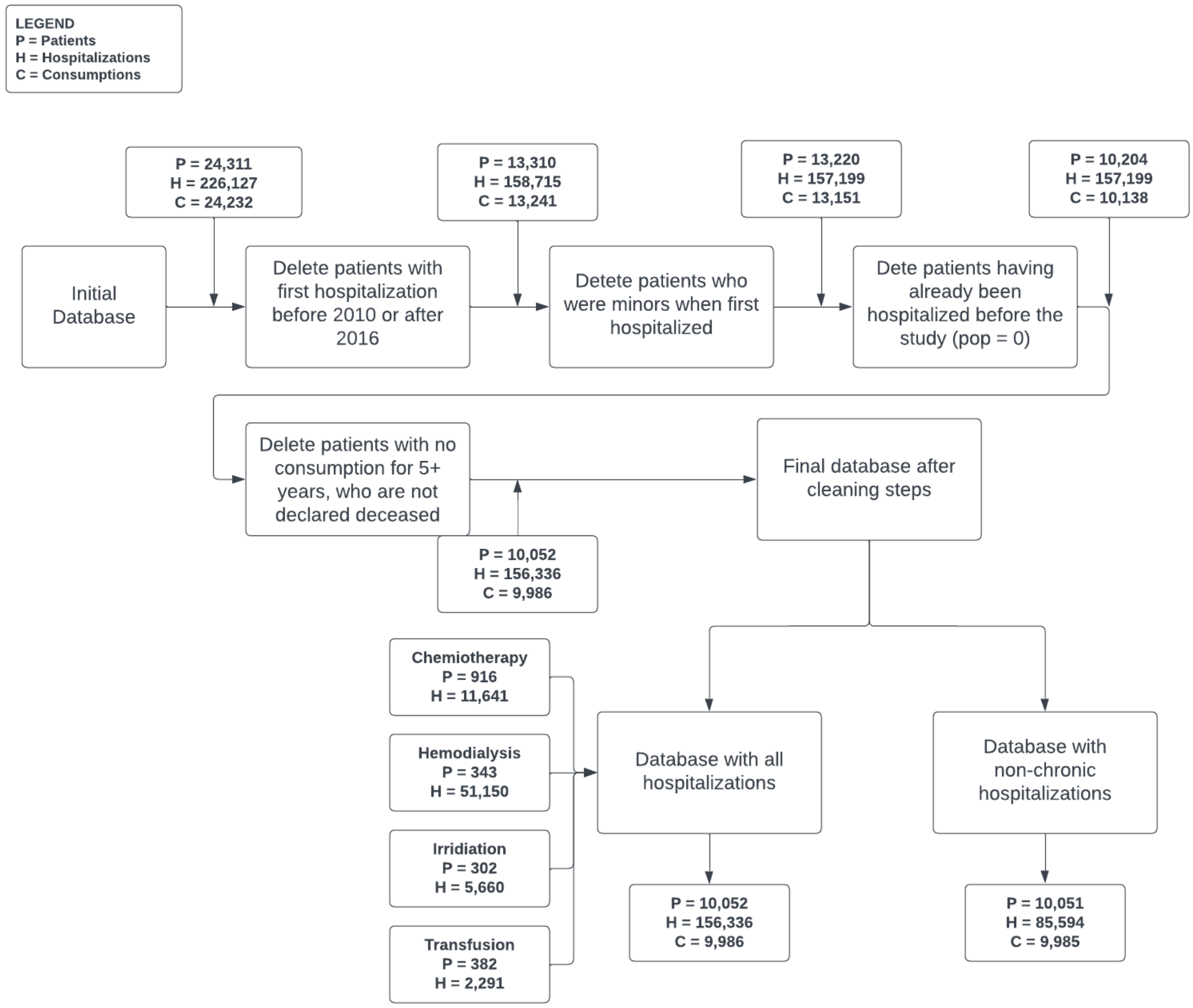


Figure 14: Flowchart of data cleaning steps

Flowchart displaying the data cleaning steps, detailed in the section *2.2 Identification and description of the population* .

A.2 Cluster Visualization

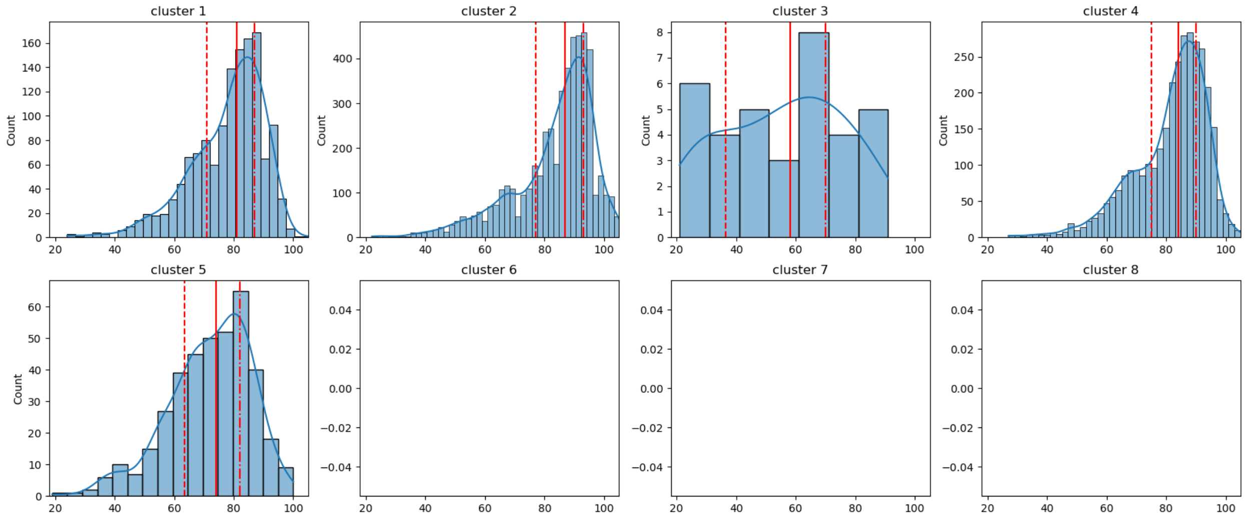


Figure 15: Age distribution per cluster

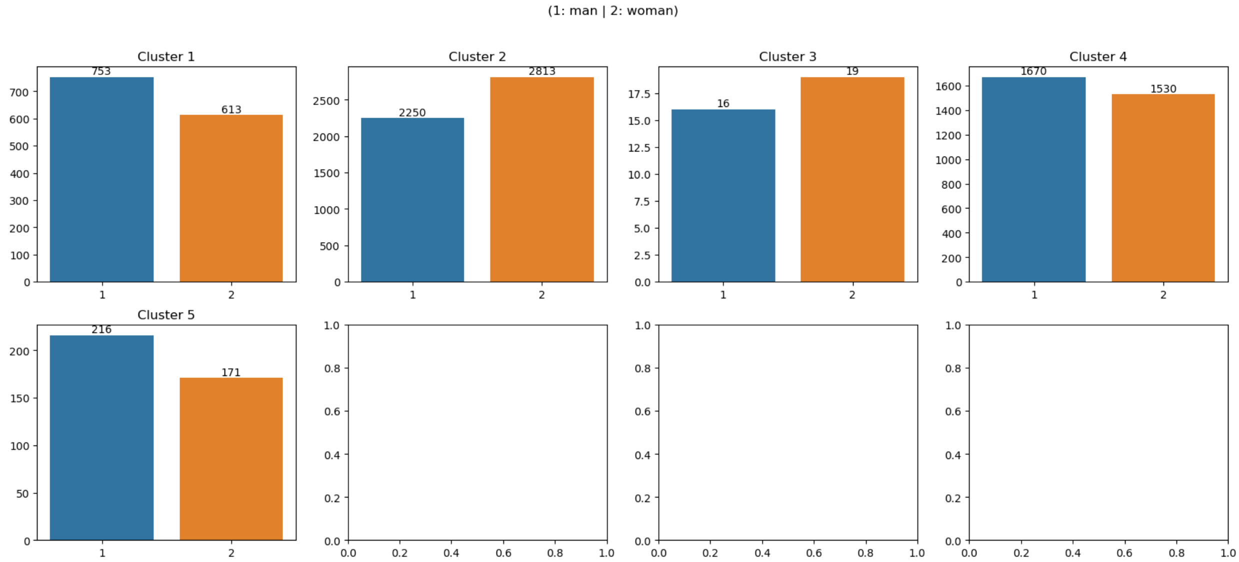
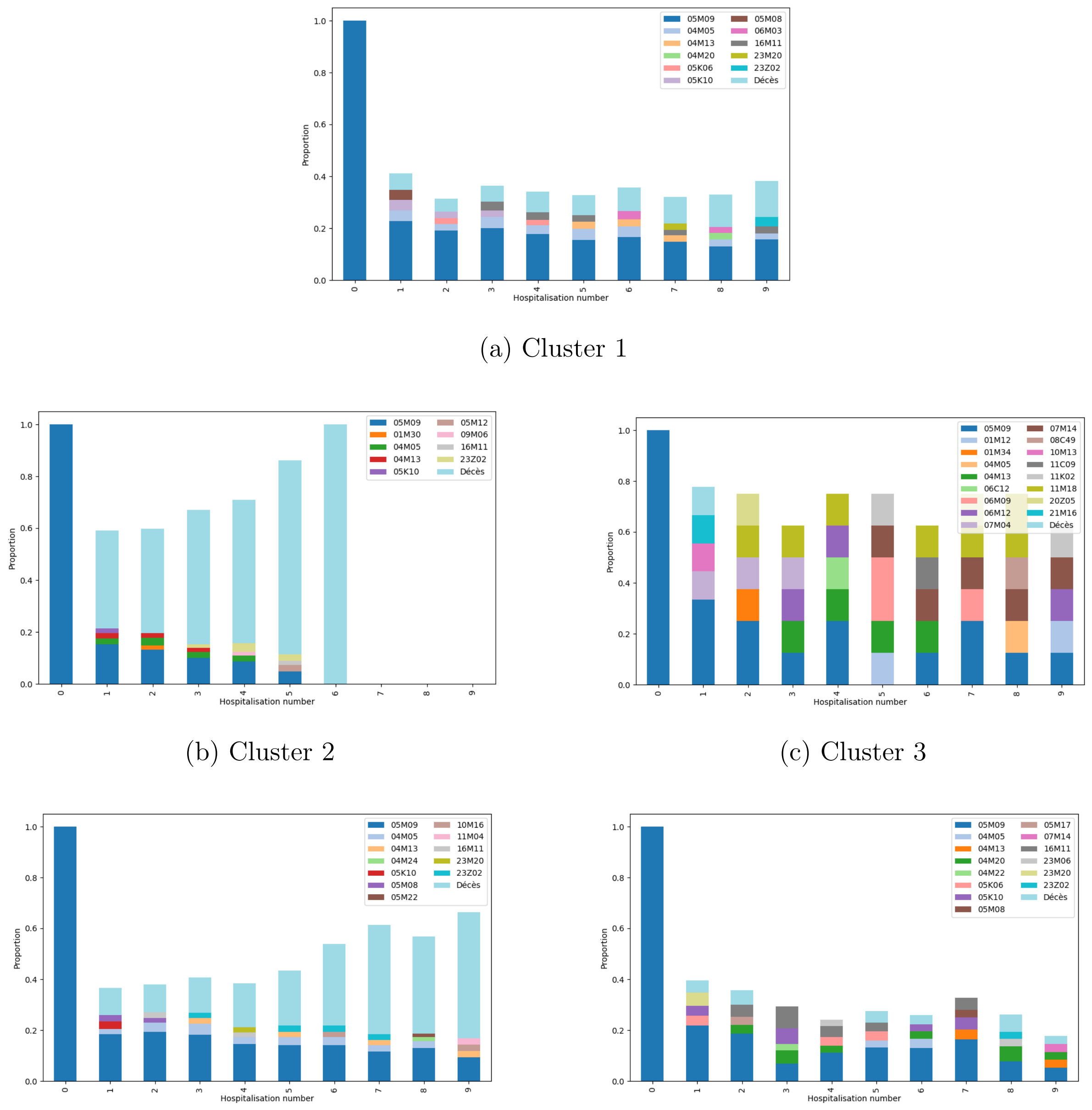


Figure 16: Gender distribution per cluster: 1 = man - 2 = woman

Figures representing the age and gender distribution in the final clusters implemented with optimal hyperparameters and our custom distance metric, as explained in *3.5 Cluster Visualization*. Overall details are to be found in the *3 Patients Clustering*.

A.3 Healthcare trajectory visualization



(a) Cluster 1

(b) Cluster 2

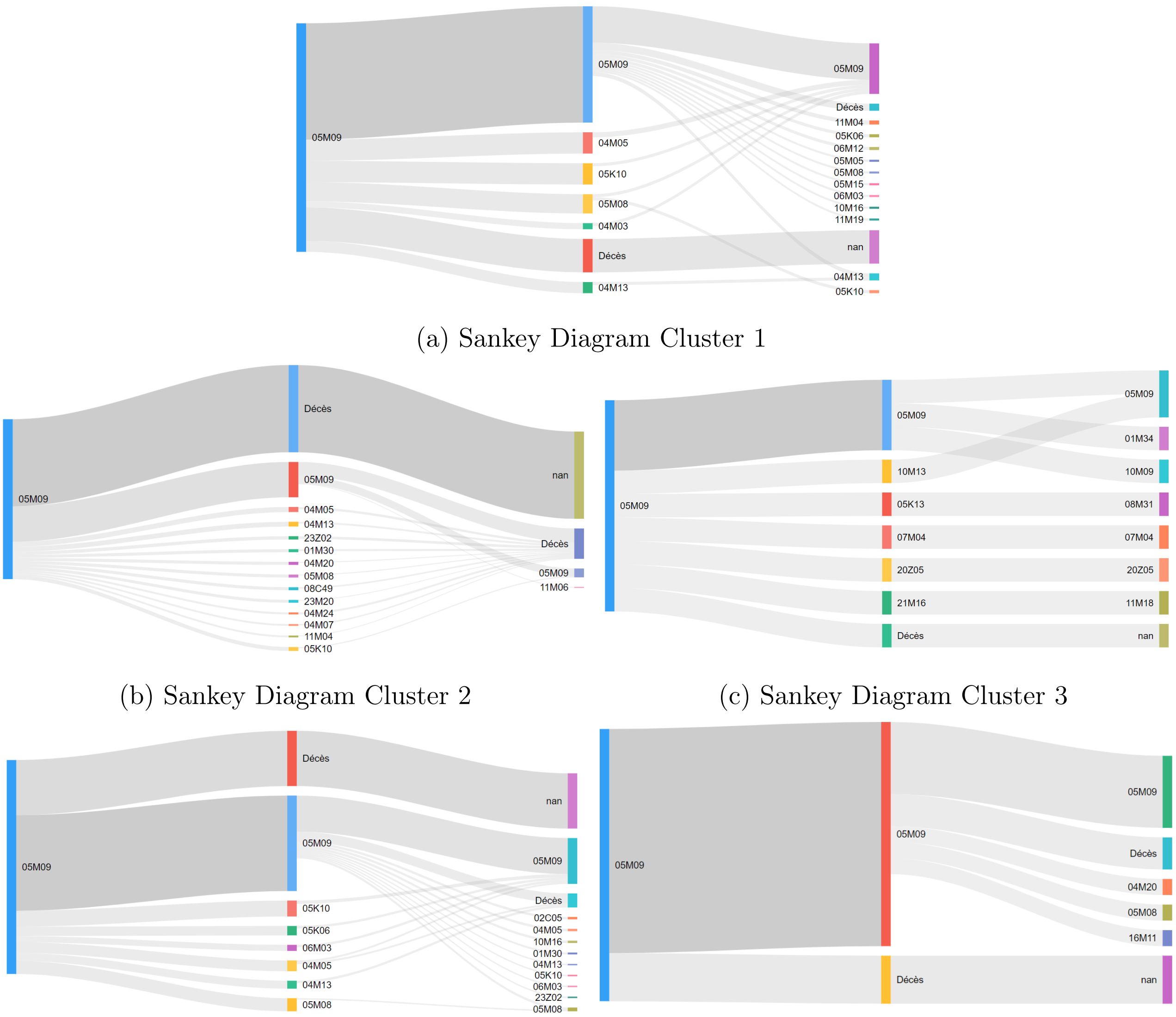
(c) Cluster 3

(d) Cluster 4

(e) Cluster 5

Figure 17: Most frequent GHMs after first hospitalization for heart failure - Deceased patients - Cluster 1 to 5

Back to section 4.2.1



(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 18: Sankey Diagrams: patient frequent flows at the cluster level

Back to section 4.2.2

A.4 Survival Analysis

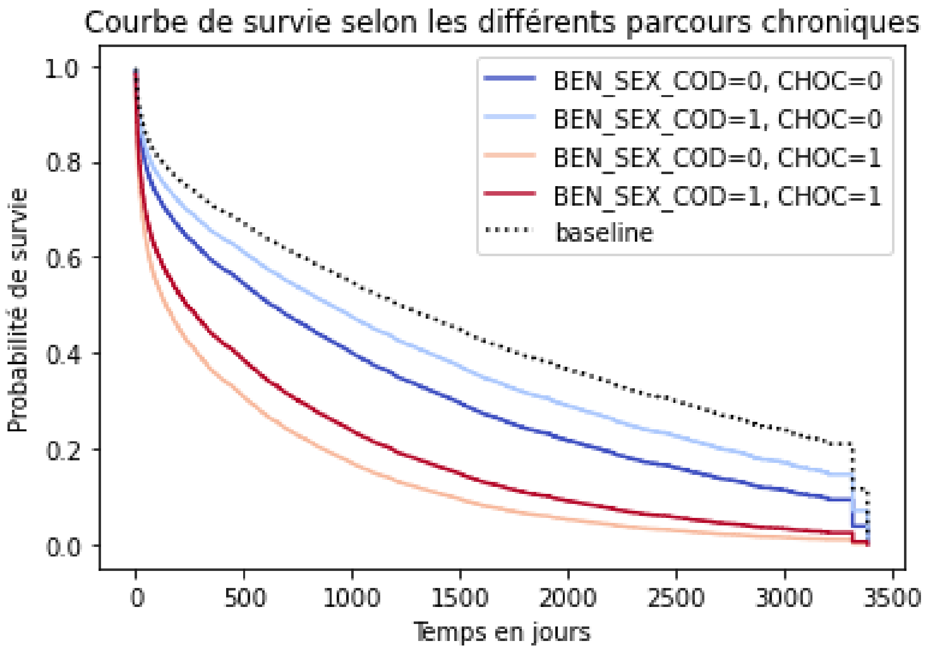


Figure 19: Probability of survival across time stratified by sexe and by CHOC

Details are to be found in the section *Significance of the coefficients*.

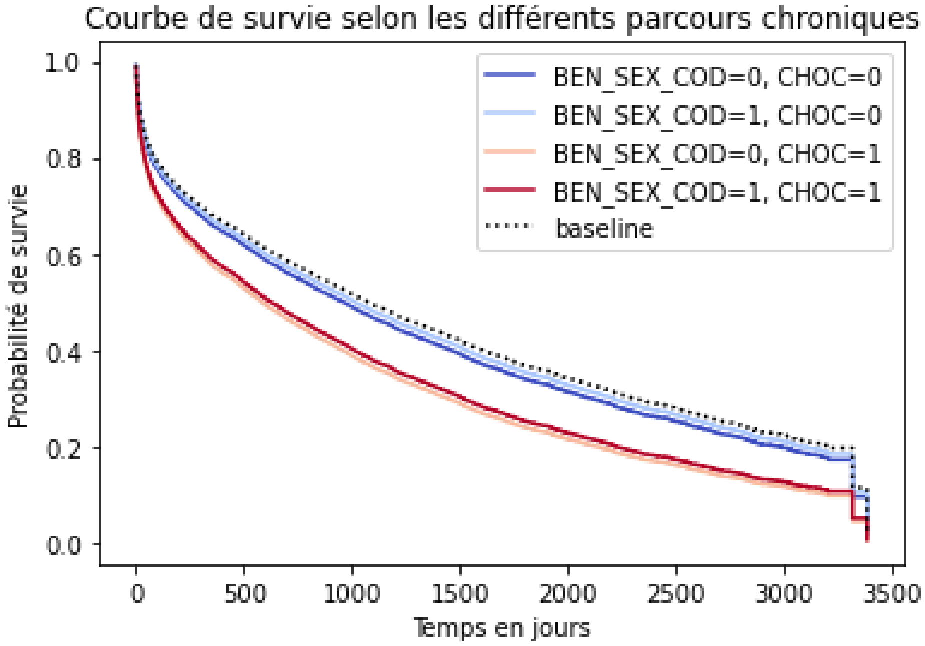


Figure 20: Probability of survival across time stratified by sex and by CHOC with a penalised model

Details are to be found in the section *Penalisation Cox Model*.

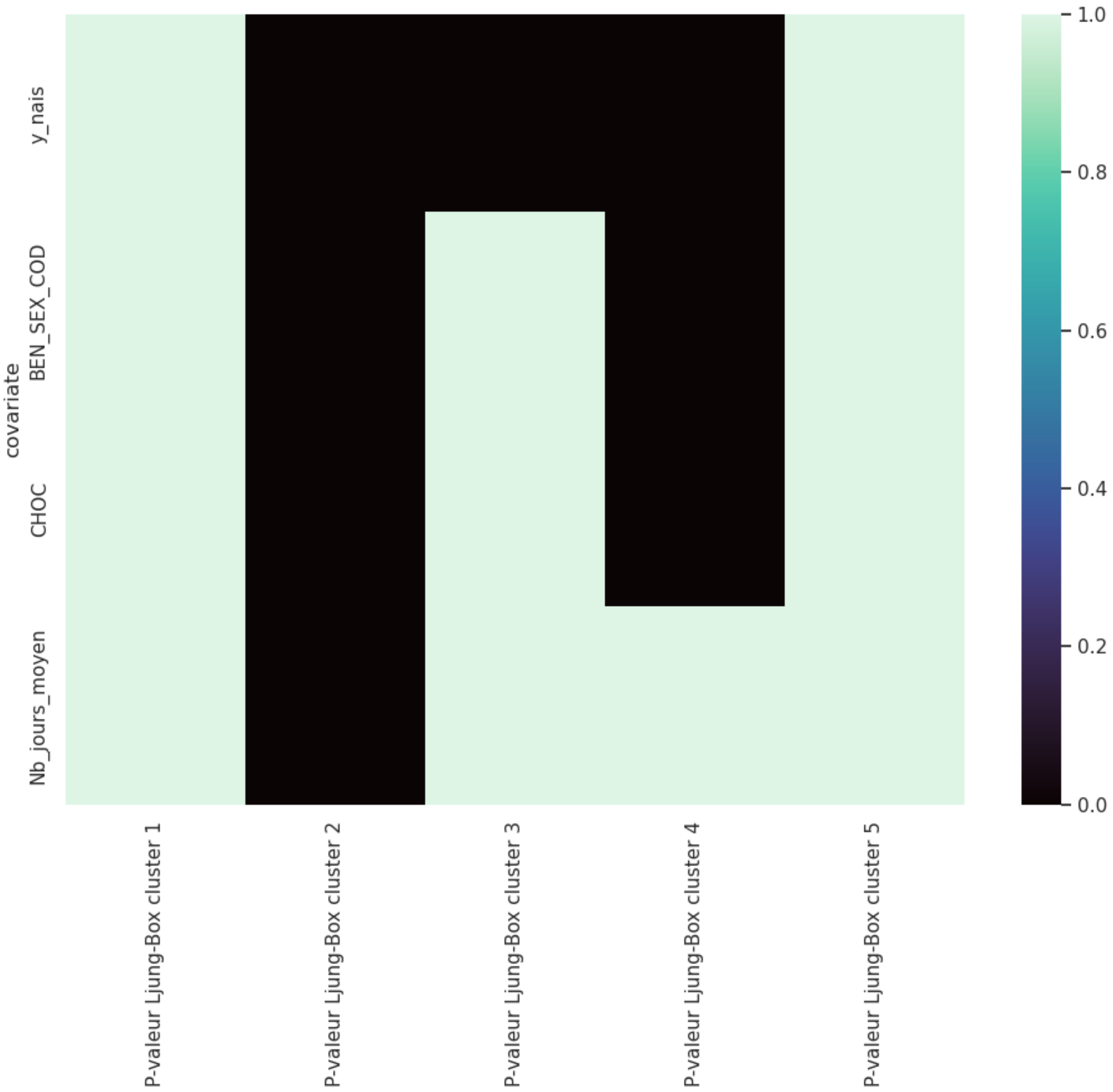


Figure 21: Ljung-Box test performed on each cluster and for each covariable

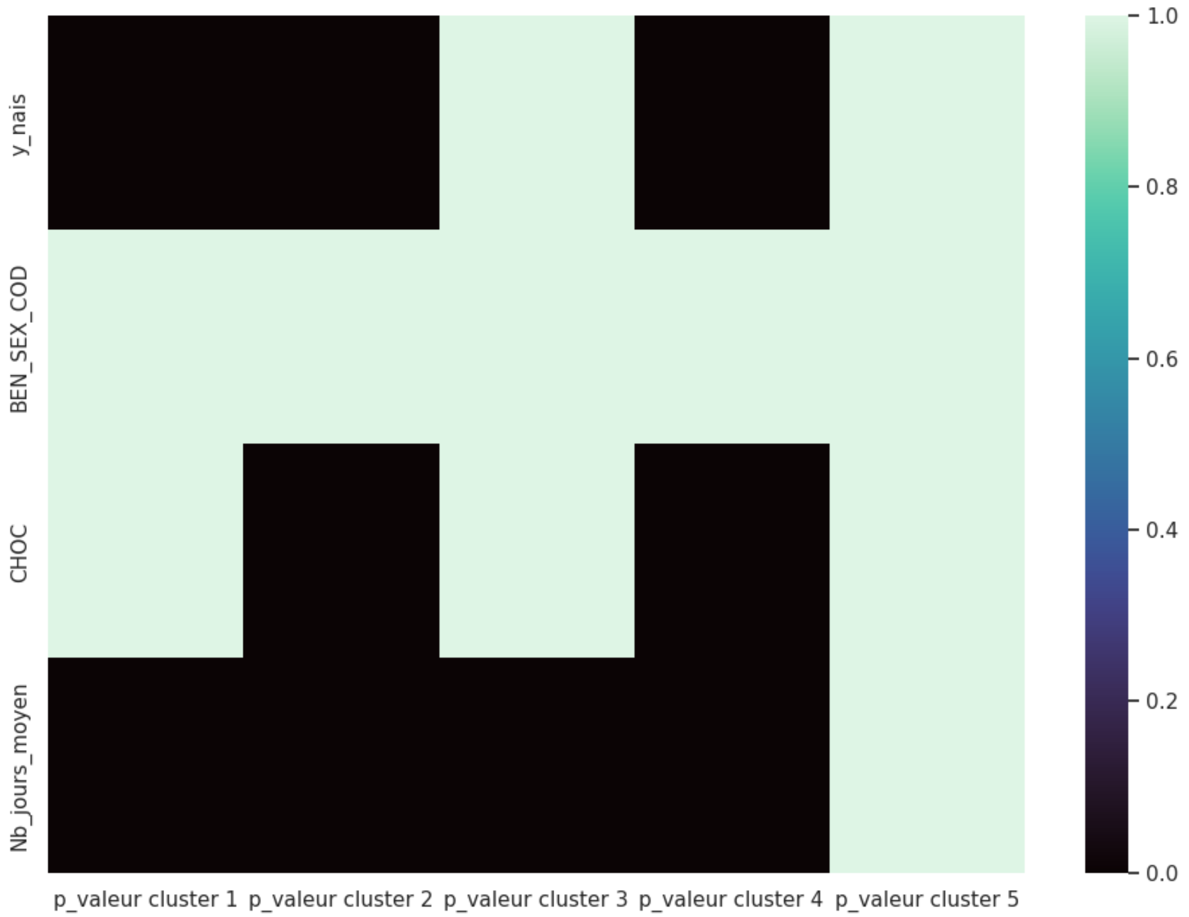


Figure 22: Log-Rank test to verify hazard function proportionality on each cluster and on each covariable

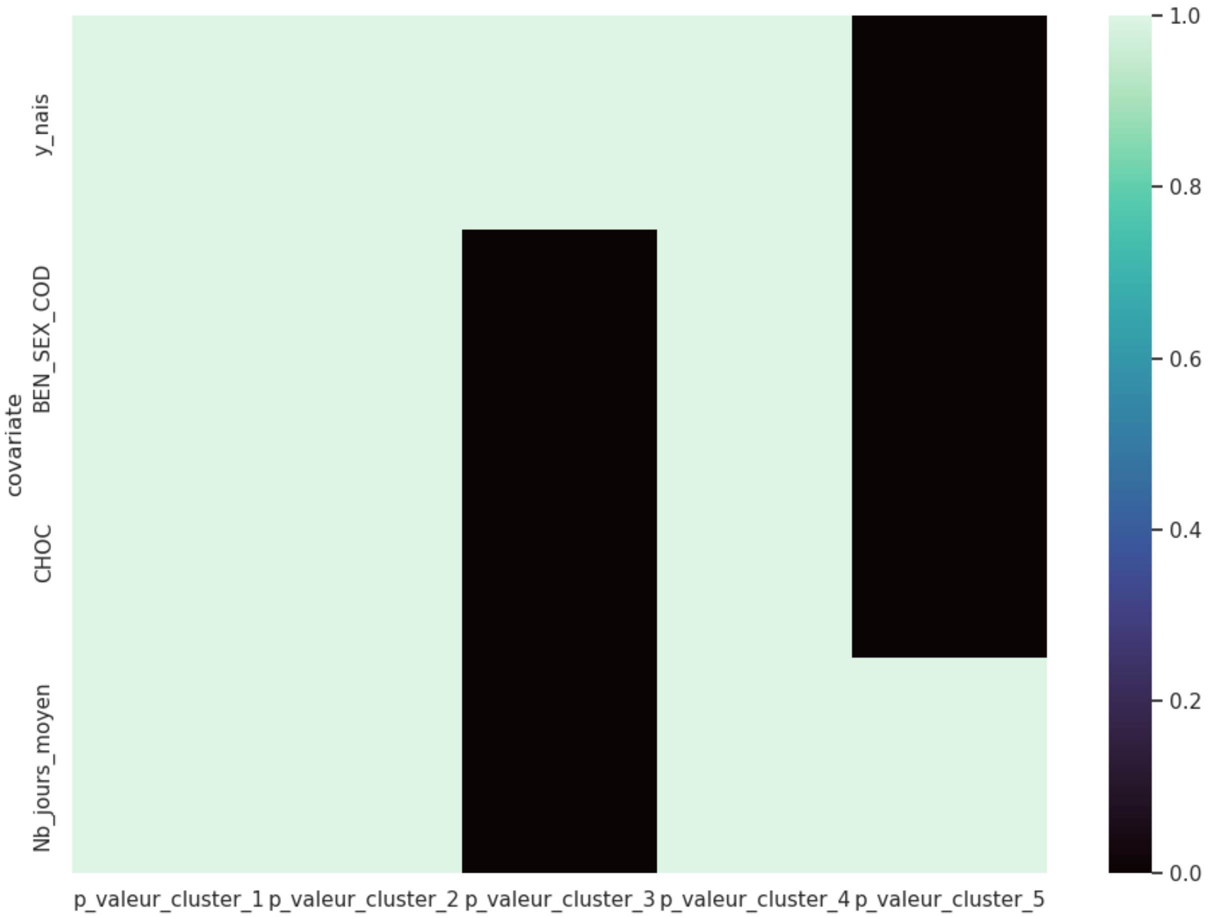


Figure 23: T-Test to verify significance of the coefficients

Details are to be found in the section *A stratified Cox Model with our clustering*.

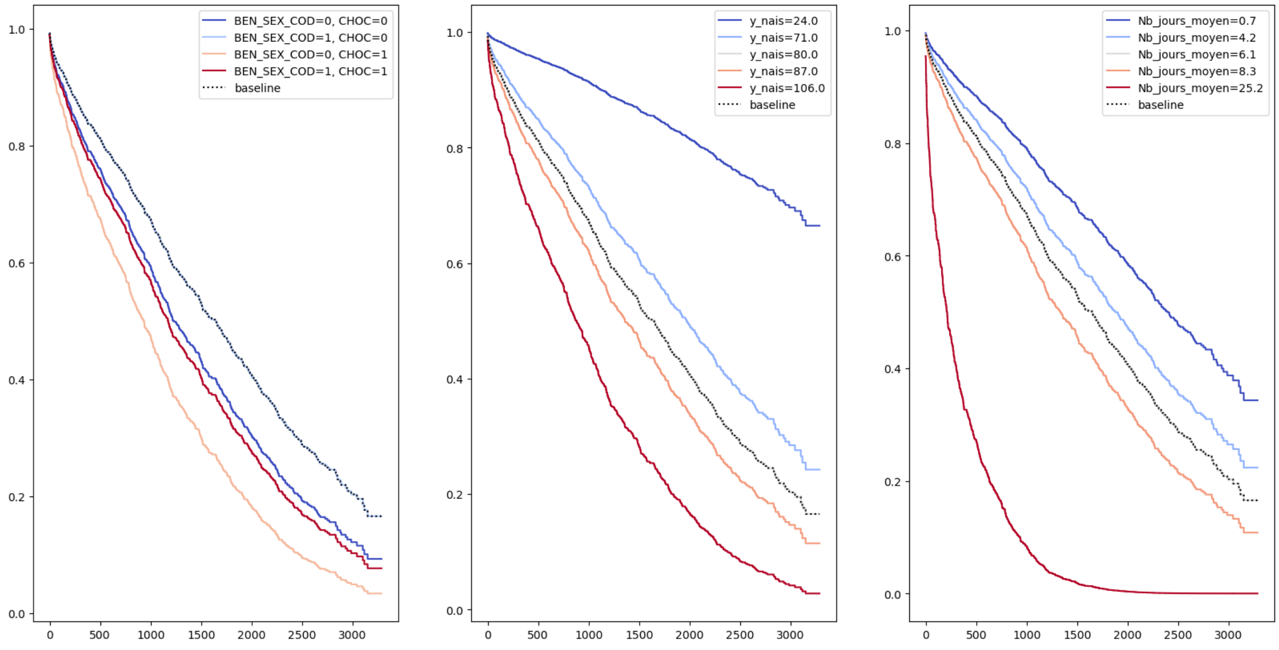


Figure 24: Survival Trajectory for **Cluster 1** stratified by :
Gender and Cardiac Shock | Age | Duration stayed at the hospital

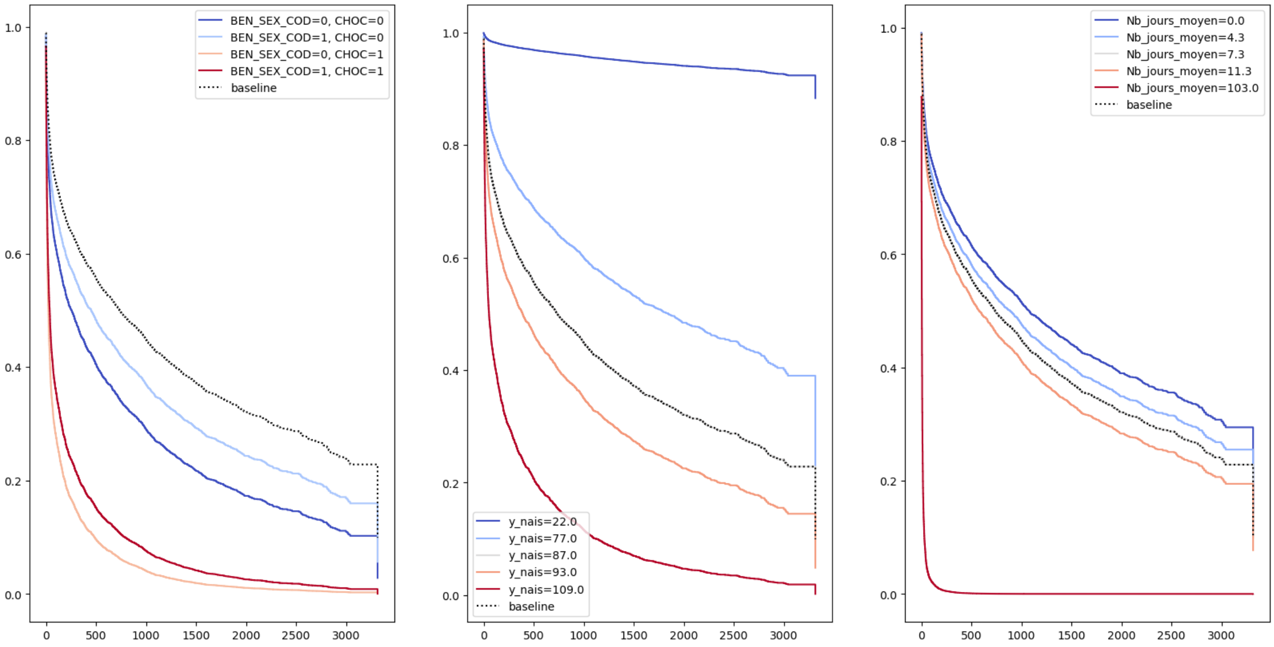


Figure 25: Survival Trajectory for **Cluster 2** stratified by :
Gender and Cardiac Shock | Age | Duration stayed at the hospital

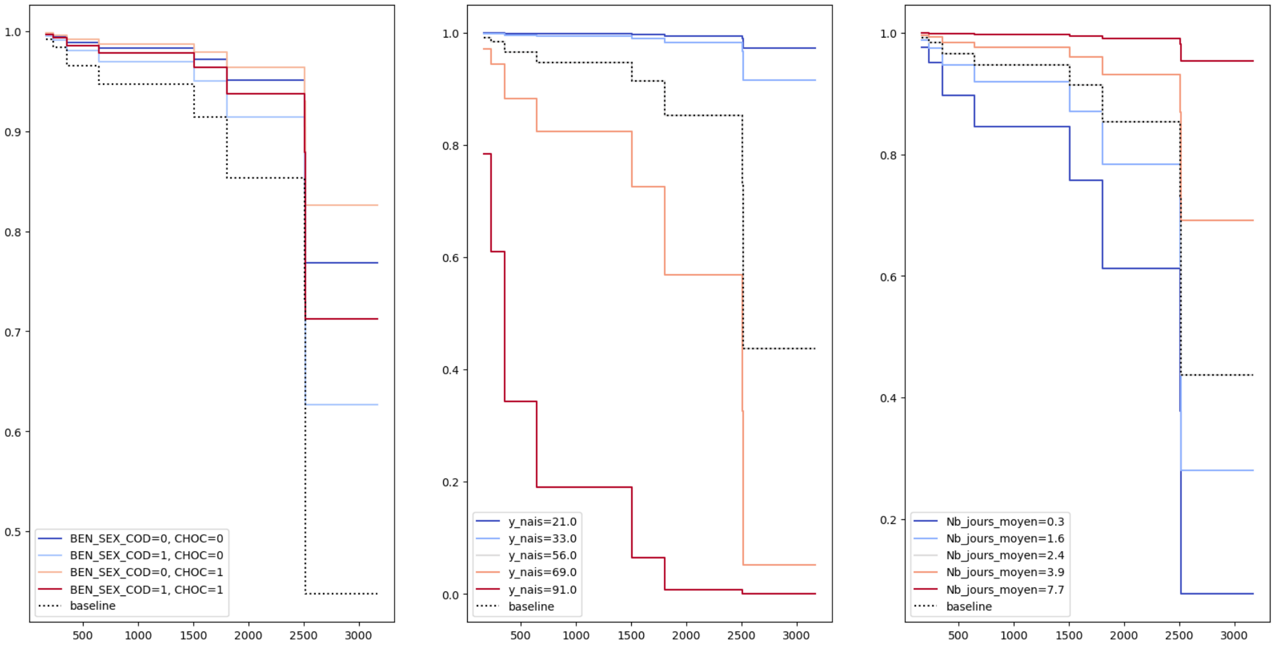


Figure 26: Survival Trajectory for **Cluster 3** stratified by :
Gender and Cardiac Shock | Age | Duration stayed at the hospital

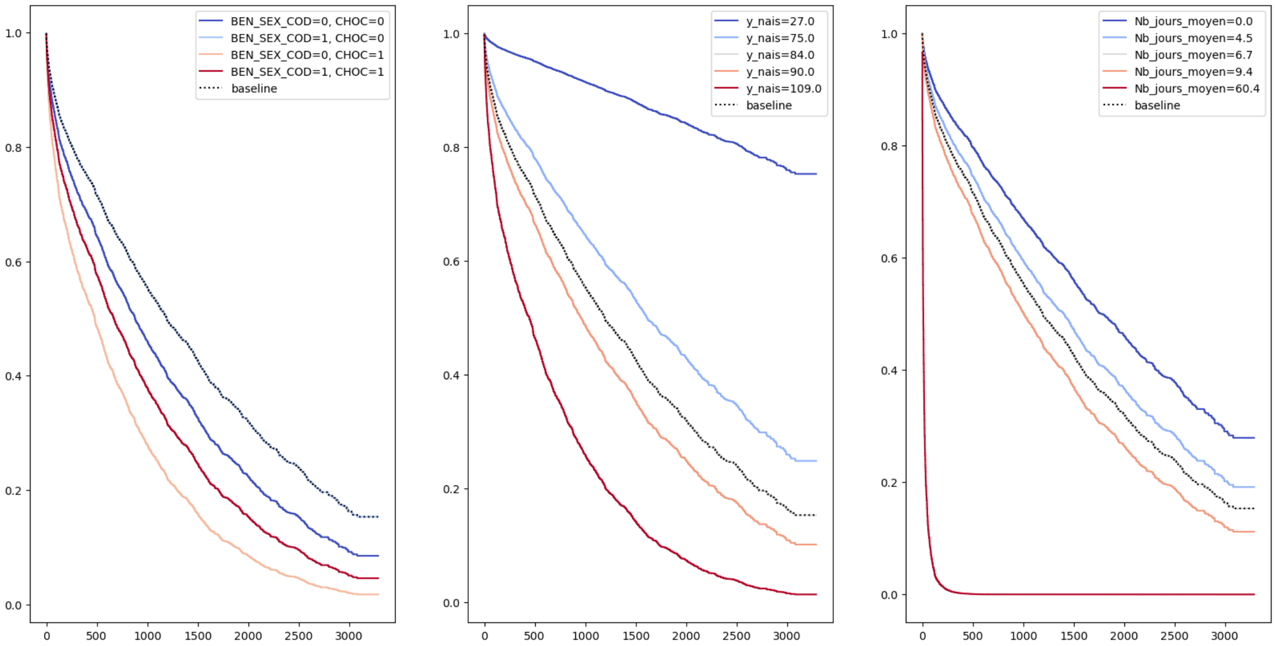


Figure 27: Survival Trajectory for **Cluster 4** stratified by :
Gender and Cardiac Shock | Age | Duration stayed at the hospital

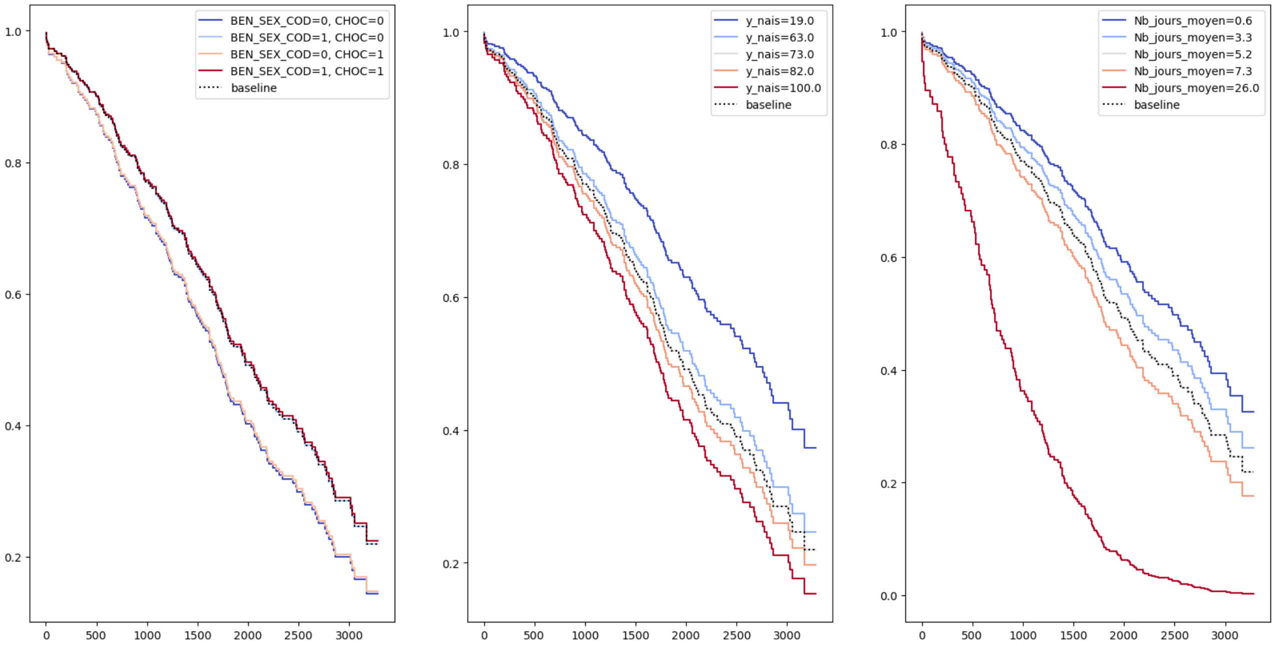


Figure 28: Survival Trajectory for **Cluster 5** stratified by :
Gender and Cardiac Shock | Age | Duration stayed at the hospital

Details are to be found in the section *A stratified Cox Model with our clustering*.

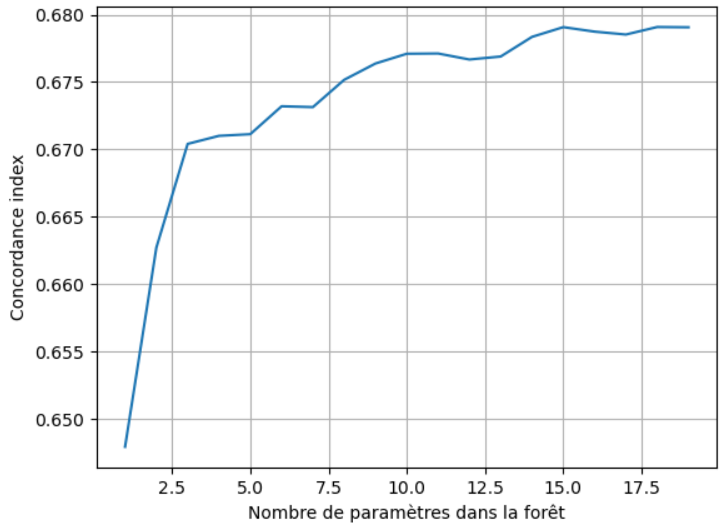


Figure 29: Grid search over the number of parameters with the concordance metric: 10 seem to the best compromise

Details are to be found in the section *Survival Random Forest*.

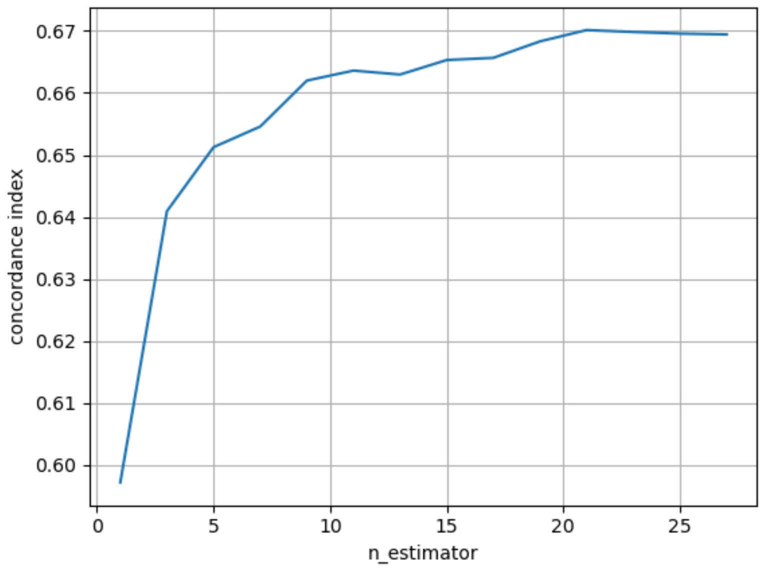


Figure 30: Grid Search with Survival Gradient Boosting

Details are to be found in the section *Survival Random Forest*.

B Tables

Back to section 4.1.3

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

Table 7: 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**', 'Décès']	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	['Décès']	129	0.333	['05M09', 'Décès']	75	0.194	['05M09', '05M09', 'Décès']

Table 8: Second 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**', 'Décès']	124	0.091	['**05K10**', '05M09', 'Décès']
2	615	0.121	['**04M05**']	418	0.083	['05M09', '05M09']	102	0.020	['02C05', '05M09', 'Décès']
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**', 'Décès']	208	0.065	['02C05', '05M09', 'Décès']
5	167	0.432	['**05K10**']	88	0.227	['05K10', '05M09']	56	0.145	['05K10', '05M09', '05M09']

Table 9: Third most occurring patterns in health care pathway for each patient clusters

	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
Décès		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

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

C The K-Medoids Algorithm

Let's explain the algorithm used in the section *3.3 Clustering Algorithm*.
The K-Medoids algorithm is a clustering method that aims to partition a given dataset into k clusters, where each cluster is represented by a single data point called a medoid. The algorithm proceeds iteratively:

1. first randomly selects k initial medoids from the dataset
2. assigns each data point to its closest medoid and calculates the total distance between each data point and its assigned medoid
3. attempts to improve the clustering by iteratively swapping one of the medoids with a non-medoid point and recalculating the total distance of the resulting clustering
4. If the total distance decreases, the swap is accepted, and the new point becomes the medoid for that cluster

This process is repeated until no further improvement can be made.
The pseudo-code below can give a more thorough idea of how the algorithm works.

Algorithm 1 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 $C$, $M$
```

In our case, the dataset is represented by the distance matrix that would have been computed prior to executing this algorithm.

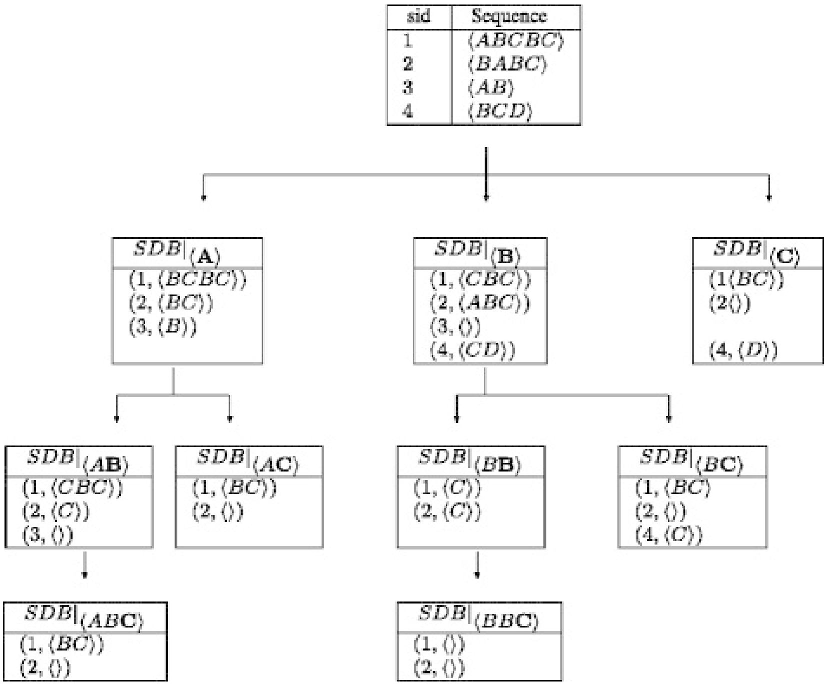
D The PrefixSpan Algorithm

Let's explain the algorithm used in the section *4.1 Sequential Pattern Mining*.
The PrefixSpan algorithm is a sequential pattern mining algorithm that 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.

Algorithm 2 Pseudocode of the PrefixSpan Algorithm

```
1. Start with an empty set of frequent patterns.
2. For each item in the first sequence of the input data, create a singleton pattern and add it to the set of frequent patterns.
3. Recursively search for longer patterns by extending each frequent pattern in the set. For each pattern:
    (a) Construct a database of all sequences that contain the pattern as a subsequence.
    (b) If the database contains at least minsup sequences, output the pattern as a frequent pattern.
    (c) 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.
    (d) Compute the support of the new pattern by concatenating the support of the item with the support of the database.
    (e) If the new pattern is frequent in the database, add it to the set of frequent patterns and continue the recursive search.
```

The algorithm uses a compact representation of frequent patterns called "sequence patterns". A sequence pattern is a tuple (P, sup), where P is a sequential pattern and sup is the number of sequences in the database that contain P. The sequence patterns are used to efficiently compute the support of candidate patterns and avoid unnecessary database scans. An example on a simple database of sequences is given Figure 31, taken from [8].



sid	Sequence
1	⟨ABCBC⟩
2	⟨BABC⟩
3	⟨AB⟩
4	⟨BCD⟩

$SDB|_{\langle \mathbf{A} \rangle}$
(1, ⟨BCBC⟩)
(2, ⟨BC⟩)
(3, ⟨B⟩)
$SDB|_{\langle \mathbf{B} \rangle}$
(1, ⟨CBC⟩)
(2, ⟨ABC⟩)
(3, ⟨⟩)
(4, ⟨CD⟩)
$SDB|_{\langle \mathbf{C} \rangle}$
(1⟨BC⟩)
(2⟨⟩)
(4, ⟨D⟩)
$SDB|_{\langle A\mathbf{B} \rangle}$
(1, ⟨CBC⟩)
(2, ⟨C⟩)
(3, ⟨⟩)
$SDB|_{\langle A\mathbf{C} \rangle}$
(1, ⟨BC⟩)
(2, ⟨⟩)
$SDB|_{\langle B\mathbf{B} \rangle}$
(1, ⟨C⟩)
(2, ⟨C⟩)
$SDB|_{\langle B\mathbf{C} \rangle}$
(1, ⟨BC⟩
(2, ⟨⟩)
(4, ⟨C⟩)
$SDB|_{\langle AB\mathbf{C} \rangle}$
(1, ⟨BC⟩)
(2, ⟨⟩)
$SDB|_{\langle BB\mathbf{C} \rangle}$
(1, ⟨⟩)
(2, ⟨⟩)

Figure 31: PrefixSpan aglorithm execution example

The PrefixSpan algorithm is known for its efficiency and scalability, particularly for mining long sequential patterns. Throughout this work, we have tried to use other SPM algorithms, notably the famous *Apriori*[2]. We found the same support for patterns of length 1. In terms of speed, *PrefixSpan* was the fastest, as the literature tends to confirm.

This[14] paper introduces the *PrefixSpan* algorithm and provides a detailed analysis of its performance. The authors compare *PrefixSpan* to other frequent sequential pattern mining algorithms, including *Apriori* and *GSP*, and demonstrate that *PrefixSpan* outperforms these algorithms in terms of runtime and memory usage.

E Relevant files attached to the project

The codes used and files generated during this project are available on the following Github repository: `https://github.com/Kirscher/TextMining_Parcours_de_soin`.

The repository includes a comprehensive list of scripts used for data processing, analysis, visualization and modelization. The files can be easily downloaded and used to replicate the results of the project or adapt the methods for other research purposes.