Chronicles mining in a database of drugs exposures

Yann Dauxais[1], David Gross-Amblard[1], Thomas Guyet[2], and André Happe[3]

[1] Université Rennes-1/IRISA - UMR6074
[2] AGROCAMPUS-OUEST/IRISA - UMR6074
[3] Plateforme PEPS/CHRU Brest

Abstract. Pharmaco-epidemiology is the study of uses and effects of health products (medical devices and drugs) on population. A new approach consists in using large administrative databases to perform such studies on care pathways which contain drugs exposures and medical problems, like hospitalizations. In this context, knowledge discovery techniques becomes mandatory to support clinicians in formulating new hypotheses. Since care-pathways are based on timestamped events and can be complex, we choose a temporal pattern mining approach. In this paper, we adapt existing chronicle mining algorithms in order to mine care-pathways. We present our method to extract all the frequent chronicles and the challenges we encountered. Finally, we present our first experimental results and our perspectives.

Keywords: Sequences mining, temporal data mining, care-pathway

1 Introduction

In classical pharmaco-epidemiology studies, people who share common characteristics are recruited to build a cohort. Then, meaningful data (drug exposures, diseases, etc.) are collected from people of the cohort within a defined period. Finally, a statistical analysis highlights the links (or the lack of links) between drug exposures and adverse effects. The main drawback of cohort studies is the time required to collect the data. Indeed, in some cases of health safety, health authorities have to answer quickly to pharmaco-epidemiology questions.

Using medico-administrative databases is an alternative to classical pharmaco-epidemiology studies. Data is immediately available and it concerns a wide population. Medico-administrative databases have been build primary to ensure health reimbursements. They record with some level of details, for all insured, all drug delivery and all medical procedure. In France, the SNIIRAM national database contains such data for more than 60 millions of insured within a sliding period of 3 years.

The challenges of making pharmaco-epidemiology studies from medico-administrative databases are 1) to abstract the administrative data into meaningful information for a specific study, and 2) to support the clinicians in their analysis of this large amount of data.

This article is focused on the second challenge and deals more specially with the extraction of frequent temporal patterns in a database of care-pathways. In a preliminary step, a dedicated method enables to translate medico-administrative data into patient care-pathways. A care-pathway is a sequence of drug exposures and medical procedures. Each element of the sequence is timestamped and each drug exposure has a time period. We propose to use sequential pattern mining to extract frequent behaviours in the patient care-pathways.

Among all the temporal patterns, *chronicles* [3] appear to be interesting to extract meaningful patterns from timestamped events. A chronicle can be briefly defined as a set of events linked by constraints indicating the minimum and maximum time elapsed between to events. A care-pathway contains point-based events and interval-based events (*e.g.* drug exposures) and a chronicle can express a complex temporal behaviour, for instance: "*The patient was exposed to a drug X between 1 and 2 years, he met his doctor between 400 to 600 days after the beginning of the exposure and, finally, he was hospitalized.*".

In this article, we propose a new algorithm to extract frequent chronicles from a database of sequences of point-based events and interval-based events in which events can be repeated.

2 Events, sequences and chronicles

In this section, we introduce some formal definitions of sequential data, chronicle pattern and of the chronicle mining task.

Definition 1. Let $\mathbb{E}$ be a set of event types and $\mathbb{T}$ a time domain where $\mathbb{T} \subseteq \mathbb{R}$, an **event** is a pair (e, t) where $e \in \mathbb{E}$ and $t \in \mathbb{T}$. We assume that $\mathbb{E}$ is totally ordered and we denote its order by $\leq_{\mathbb{E}}$.

An **event sequence** S is a tuple $\langle SID, \langle (e_1, t_1), (e_2, t_2), ..., (e_n, t_n)\rangle\rangle$ where SID is the sequence identifier in the database and $\langle (e_1, t_1), (e_2, t_2), ..., (e_n, t_n)\rangle$ is a finite sequence of events. For all $i, j, i < j \Rightarrow t_i \leq t_j$. If $t_i = t_j$, then $e_i <_{\mathbb{E}} e_j$.

In pharmaco-epidemiology studies, a sequence is the care-pathway of a patient identified by SID. A care-pathway consists of point-based events and interval-based events. A point-based event (e, t) represents a medical consultation or a delivery of a drug where e is its event type (consultation or drug name). For drug exposures, which are commonly represented by interval-based events, we use two point-based events (e_s, t_s) and (e_f, t_f) where e_s (resp. e_f) is an event type corresponding to the interval beginning (resp. ending) of an event e.

Example 1 (Database of sequences, $\mathcal{S}$)**.**

SID	sequence
1	$(A_s, 1)$, $(B, 3)$, $(A_f, 4)$, $(C, 5)$, $(B, 20)$
2	$(B, 1)$, $(A_s, 4)$, $(B, 5)$, $(D, 6)$, $(A_f, 8)$, $(C, 9)$
3	$(C, 1)$, $(D, 2)$, $(C, 2)$, $(B, 7)$
4	$(B, 1)$, $(B, 3)$, $(A_s, 7)$, $(A_f, 9)$, $(C, 11)$, $(D, 12)$

The database contains four sequences. There are one type of interval-based event (A) and three types of point-based events (B, C and D).

We will now define the notion of chronicle, which is a pattern of events and a set of temporal constraints. We begin by defining the latter:

Definition 2. A **temporal constraint**, denoted $e_1[t^-, t^+]e_2$, is a tuple where $(e_1, e_2) \in \mathbb{E}$, $e_1 \leq_{\mathbb{E}} e_2$ and $(t^-, t^+) \in \mathbb{T}$, $t^- \leq t^+$. A temporal constraint is satisfied by a pair of events $((e, t_1), (e', t_2))$, $e \leq_{\mathbb{E}} e'$ iff $e = e_1$, $e' = e_2$ and $t^- \leq t_2 - t_1 \leq t^+$. We say that $e_1[a, b]e_2 \subseteq e_1'[a', b']e_2'$ iff $e_1 = e_1'$ and $e_2 = e_2'$ and $[a, b] \subseteq [a', b']$. Hence $\subseteq$ is a partial order on the set of the temporal constraints.

Definition 3. A **chronicle** is a pair $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ such that $\mathcal{E} = \{e_1, \ldots, e_n\}$, $e_i \in \mathbb{E}$, where $\forall i, j$, $1 \leq i < j \leq n$, $e_i \leq_{\mathbb{E}} e_j$; and such that $\mathcal{T}$ is a set of temporal constraints where there is at most one temporal constraint between two events of the chronicle, *i.e.* $\forall e, e' \in \mathcal{E}$, $|\{e[a, b]e' \mid e[a, b]e' \in \mathcal{T}\}| \leq 1$. $\mathcal{E}$ is called a multiset. It is a set of events allowing repetitions.

Example 2. Figure 1 illustrates the chronicle $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ where $\mathcal{E} = \{e_1 = A_s, e_2 = A_f, e_3 = B, e_4 = B, e_5 = C\}$ and $\mathcal{T} = \{e_1[2, 4]e_2, e_1[-4, 2]e_3, e_2[-8, 1]e_3, e_2[1, 2]e_5, e_3[2, 17]e_4, e_4[-15, 8]e_5\}$. (A_s, A_f) can be seen as a pair of events representing an interval event A that starts with event A_s and that finishes with event A_f.

We can notice that the graph is not complete. The lack of arc between two nodes can be interpreted as a $[-\infty, +\infty]$ constraint. But, in most case, a more restrictive constraint can be deduced from the other constraints. For instance, a temporal constraint $A_s[3, 6]C$ can be deduced from constraints between A_s and A_f, and between A_f and C.

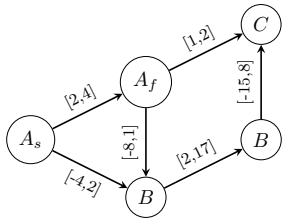

Fig. 1: Chronicle example.

Given two chronicles $\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ and $\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$, we define the partial order $\preceq$ where $\mathcal{C}_1 \preceq \mathcal{C}_2$ if $\mathcal{E}_2 \subseteq \mathcal{E}_1$ and there is a strictly increasing function f where $\forall i, j$, $1 \leq i < j \leq |\mathcal{E}_2|$, $1 \leq f(i) < f(j) \leq |\mathcal{E}_1|$, $e_i, e_j \in \mathcal{E}_2$, $e_{f(i)}, e_{f(j)} \in \mathcal{E}_1$, $e_{f(i)}[a, b]e_{f(j)} \in \mathcal{T}_1$, $e_i[a', b']e_j \in \mathcal{T}_2$, $e_{f(i)}[a, b]e_{f(j)} \subseteq e_i[a', b']e_j$. If $\mathcal{C}_1 \preceq \mathcal{C}_2$ and $\mathcal{C}_1 \neq \mathcal{C}_2$, we say that $\mathcal{C}_1$ is **more specific** than $\mathcal{C}_2$ or is a child of $\mathcal{C}_2$. On the contrary, $\mathcal{C}_2$ is **more general** than $\mathcal{C}_1$ or is a parent of $\mathcal{C}_1$. An **extended child**

$\mathcal{C}'$ of a chronicle $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ is $\mathcal{C}' = (\mathcal{E} \cup \{e\}, \mathcal{T}')$ where $\mathcal{T}'$ is the union of $\mathcal{T}$ and of a set of temporal constraints between e and e_i for all e_i in $\mathcal{E}$. A **specialized child** $\mathcal{C}'$ of a chronicle $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ is $\mathcal{C}' = (\mathcal{E}, \mathcal{T} \backslash \{\tau\} \cup \{\tau'\})$ where $\tau' \subset \tau$.

Definition 4. Let $s = \langle (e_1, t_1),\ ...,\ (e_n, t_n) \rangle$ be a sequence and $\mathcal{C} = (\mathcal{E} = \{e'_1, \ldots, e'_m\}, \mathcal{T})$ a chronicle. An **occurrence** of the chronicle $\mathcal{C}$ in s is a subsequence $\tilde{s} = \langle (e_{f(1)}, t_{f(1)}),\ ...,\ (e_{f(m)}, t_{f(m)}) \rangle$ such that there exists a function f where $\forall i, j,\ 1 \leq i < j \leq m,\ 1 \leq f(i) \leq n,\ 1 \leq f(j) \leq n,\ f(i) \neq f(j)$ such that 1) $e'_i = e_{f(i)},\ e'_j = e_{f(j)}$ and 2) $t_{f(j)} - t_{f(i)} \in [a, b]$ where $e'_i[a, b]e'_j \in \mathcal{T}$. $\mathcal{C}$ **occurs** in s, denoted by $\mathcal{C} \in s$, iff there is at least one occurrence of $\mathcal{C}$ in s.

Definition 5. The **support** of a chronicle $\mathcal{C}$ in a database of sequences $\mathcal{S}$ is the number of sequences in which $\mathcal{C}$ occurs: $\mathit{support}(\mathcal{C}, \mathcal{S}) = |\{S \mid S \in \mathcal{S} \text{ and } \mathcal{C} \in S\}|$. Given a minimal threshold $\sigma_{min} \in \mathbb{N}$, a chronicle $\mathcal{C}$ is said **frequent** in $\mathcal{S}$ iff $\mathit{support}(\mathcal{C}, \mathcal{S}) \geq \sigma_{min}$.

According to the anti-monotony property, if a chronicle $\mathcal{C}$ is frequent then all chronicles more general than $\mathcal{C}$ are frequent. One can easily be convinced of this by observing that, if a chronicle $\mathcal{C}'$ is more general than a chronicle $\mathcal{C}$, then $\mathcal{C}'$ has at least the same support as $\mathcal{C}$ because any occurrences of $\mathcal{C}$ is necessarily an occurrence of $\mathcal{C}'$. The proof is omitted for space reason.

3 Related works

The first algorithm dedicated to chronicle mining was proposed by Dousson and Duong [3]. This algorithm was originally designed for chronicle discovery in journal logs of telecommunication alarms. Recently, several improvements have been proposed [1, 2, 4]. All of these approaches are based on the anti-monotonicity property of frequency on the chronicle set.

Cram [2] proposed the *HCDA* algorithm which improves the first algorithm by mining the complete set of frequent chronicles. Those two approaches start with the extraction of frequent temporal constraints between pairs of events and then chronicles are generated by combining these constraints. The method of Dousson and Duong chooses only a representative of each type of temporal constraint while *HCDA* keeps all frequent temporal constraints in a graph of temporal constraints. These two methods process journal logs, *i.e.* a single long sequence. In our mining task, we have a database of sequences and the definition of the pattern support is based on a number of supported sequences. As a consequence, we can not apply these algorithms for our task. For this reason, we propose an adaptation of them.

In *CCP-Miner*, Huang [4] proposed to use chronicle mining on clinical pathways. Their data comes from inpatient electronic health record. Contrary to journal logs, a set of clinical pathways is a database of sequences. To simplify the evaluation of the support, *CCP-Miner* considers that an event type occurs at most one time in a pathway. Moreover, *CCP-Miner* is not complete. Chronicles are not obtained from the complete set of frequent multisets of event types but only those containing by frequent closed sequences.

Subias et *al.* [6] recently proposed an alternative support evaluation which is the number of sequences in which the number of occurrences of a chronicle in one sequence is above a given threshold. This support measure is not relevant in our application.

In parallel to alternative support, several approaches have been proposed to extract chronicles with simpler temporal constraints. For instance, Àlvarez et *al.* [1] use a similarity criterion to cluster together different temporal arrangements between events. Quiniou et *al.* [5] proposed an inductive logic programming approach to mine chronicles with quantified temporal constraints.

To the best of our knowledge, there is no algorithm that can extract a complete set of chronicles from a database in which sequences may contain duplicated events. The proposed method tackles this specific issue.

4 Complete chronicle mining in a database of sequences

The chronicle mining task is a classical pattern mining task. The search space is structured by a partial order, $\preceq$, (see section 2) and the frequency is an anti-monotonic measure in this space. As a consequence, the classical "generate and test" strategy can be applied: candidate k-patterns are generated from $(k-1)$-frequent patterns, frequency of candidates is evaluated. Then, the two main problems to tackle are 1) how to efficiently browse the search space and 2) how to evaluate the frequency of a pattern.

In this article, we propose an algorithm to extract the frequent chronicles in a database of sequences. This algorithm combines the approaches of $HCDA$ [2] and of $CCP\text{-}Miner$ [4]. We use the $CCP\text{-}Miner$ strategy that first extracts the multisets of event types and then add temporal constraints over those multisets. The generation of the temporal constraint is adapted from $HCDA$ in order to deal with databases of sequences. This two improvements are explained in the following section but before that, we detail the support evaluation process.

Enumerating sequences of a database that support a chronicle is simpler than the original chronicle enumeration of $HCDA$. In fact, evaluating the number of occurrences of a chronicle in a single sequence is very combinatorics because of the repetition of the events. Our support measure corresponds to the number of sequences where a chronicle occurs. For each sequence, we just have to search for one occurrence of this chronicle. Moreover, this support definition simplifies the construction of bases of constraints (see section 4.3).

4.1 A two steps strategy

Our algorithm is illustrated in Algorithm 1. Let $\mathcal{S}$ be a set of event sequences and σ_{min} be the minimal support threshold. Firstly, the $extractMultisets$ function generates ES, the set of all frequent multisets of event types $\mathcal{E}$ accordingly to σ_{min}. On the contrary to the $CCP\text{-}Miner$ approach, the algorithm does not generate chronicles from closed sequences. Multisets mining is an easy extension of itemsets mining and we do not detail with step of the algorithm.

Algorithm 1 Main algorithm for chronicle mining

1: $CS \leftarrow \emptyset$
2: $ES \leftarrow extractMultisets(\mathcal{S}, \sigma_{min})$
3: **for** each $e \in ES$ **do**
4: $\quad CS \leftarrow CS \cup extendChronicles(e, \sigma_{min})$
5: **return** CS

Then, multisets are extended in frequent chronicles and their temporal constraints are specialized. This step is performed for each multiset by the function *extendChronicles*. The set CS corresponds to the frequent chronicles. We detail this part of the algorithm in the following sections.

4.2 From multisets to chronicles

This section presents the generation of frequent chronicles from frequent multisets. Given a multiset $\mathcal{E}$, the exploration consists in generating all combinations of temporal constraints on $\mathcal{E}$, such that corresponding chronicles are frequent.

Temporal constraint bases To ensure the efficiency and the completeness of candidate generation, we use **temporal constraint bases** (TCB). A TCB is a set of graphs of temporal constraints (one per pair of events). Figure 2 illustrates a graph of temporal constraints.

Definition 6. A graph of temporal constraints $\mathcal{G}$ is a directed acyclic graph in which a node is a temporal constraint, τ, and children of τ are temporal constraint included in τ. The root of the graph is called the **top-constraint**. In our algorithm, we consider that a temporal constraint $\tau = e_1[a, b]e_2$ has at most two children, $\tau_{left} = e_1[a, b']e_2$ and $\tau_{right} = e_1[a', b]e_2$ where $b' < b$ and $a' > a$.

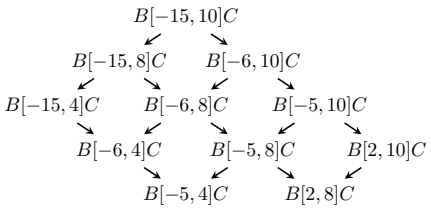

Fig. 2: A graph of temporal constraints for the pair of events (B, C).

Specialization of multisets Let $\mathcal{E}$ be a multiset. The "top-chronicle" $(\mathcal{E}, \mathcal{T})$ is generated such that for each $\tau \in \mathcal{T}$, τ is a top-constraint. Then, function

extendChronicles generates the complete set of frequent chronicles from $(\mathcal{E}, \mathcal{T})$ by specializing the temporal constraints on the multiset $\mathcal{E}$. The specialization of a chronicle consists in specializing a temporal constraint τ according to the specialization defined in its graph of temporal constraints.

The generation of chronicles is a "generate and test" approach. The specialization is done recursively to extract the complete set of frequent chronicles. Each time a chronicle $\mathcal{C}$ is specialized, its frequency in the database is evaluated. If its support is not above the minimal support, the chronicle is pruned. According to the anti-monotony property, we know that not any chronicle $\mathcal{C}'$, $\mathcal{C}' \preceq \mathcal{C}$, will be frequent.

The enumeration of specialized chronicles is done without redundancy. It is ensured thanks to the definition of a relation order $\lhd_{\mathcal{E}}$ amongst all chronicles sharing the same multiset $\mathcal{E}$. Let $\mathcal{C} = (\mathcal{E}, \mathcal{T} = \{t_1, \ldots, t_n\})$ and $\mathcal{C}' = (\mathcal{E}, \mathcal{T}' = \{t'_1, \ldots, t'_n\})$ be chronicles, $\mathcal{C}' \lhd_{\mathcal{E}} \mathcal{C}$ iff $\exists k$, $1 \leq k < n$ such that 1) $\forall i$, $1 \leq i < k$, $t_i = t'_i$, 2) $\forall j$, $k < j < n$, $t_j = t'_j$ and t_j is a top-constraint and 3) $t'_k = t_{k_{right}}$ otherwise $t'_k = t_{k_{left}}$ if $\nexists \tau$, $\tau_{left} = t'_k$.

It can be shown that $\mathcal{C}' \lhd_{\mathcal{E}} \mathcal{C} \Rightarrow \mathcal{C}' \preceq \mathcal{C}$ and that there exists at most one chronicle $\mathcal{C}$ such that $\mathcal{C}' \lhd_{\mathcal{E}} \mathcal{C}$. These properties ensure a unique and complete traversal of the search space. For space reason, we omit the proof of these properties.

4.3 Generation of the Base of Temporal Constraints

This section presents the generation of the TCB. The smaller are the TCB, the more efficient is the multiset specification. On the other hand, these bases must be complete, *i.e.* the chronicle mining algorithm of the section 4.1 must extract the complete set of frequent chronicles.

In these objectives, the algorithm generates the smaller complete TCB from the sequence database. A first algorithm has been proposed in *HCDA*. Our algorithm improves it by considering two specificities of our dataset:

1. the enumeration of chronicle occurrences in a database of sequences
2. the specificities of events that encode the period of an interval event with a pair of point-based events

Let $(e, e') \in \mathbb{E}^2, e \leq_{\mathbb{E}} e'$ be a pair of point-based events. We denote by $\mathcal{A}_{ee'} \subset \mathbb{R}$, the list of pairs (a, SID) for each co-occurrence $((e, t), (e', t'))$ in a sequence where a is the duration $t' - t$ and SID is the identifier of the sequence. The lists corresponding to all the pairs (e, e') in $\mathbb{E}^2$ can be filled in one pass of the database by generating all co-occurrences present in each sequence. We can notice that the duration can be negative if e' occurs before e. After this step, the lists are sorted by duration in ascending order and the duplicates of couple are removed. Similar lists are built from start/finish events of intervals.

The temporal constraint graph generation is given in the Algorithm 2. Each list $\mathcal{A}_{ee'}$ corresponds to a graph $\mathcal{G}_{ee'}$. To respect our support measure we check whether elements of $\mathcal{A}_{ee'}$ correspond to at least σ_{min} sequences. Otherwise $\mathcal{G}_{ee'}$ is empty. In the other case, the root of $\mathcal{G}_{ee'}$ is $e[a, b]e'$ where a is the duration of

Algorithm 2 Temporal constraint graph generation

```
1: function ConstructGraph(A_ee', σ_min)
2:     τ ← ∅
3:     if |{SID | (a, SID) ∈ A_ee'}| ≥ σ_min then
4:         (a, s) ← first(A_ee')
5:         (b, t) ← last(A_ee')
6:         τ ← e[a, b]e'
7:         τ_left ← ConstructGraph(A_ee'\{(b, t)})
8:         τ_right ← ConstructGraph(A_ee'\{(a, s)})
9:     return τ
```

the first element of $\mathcal{A}_{ee'}$ and b that of the last one. Then we built $\mathcal{G}_{ee'}$ recursively by defining the left child of a node as the graph corresponding to $\mathcal{A}_{ee'}$ without its last element and right child to $\mathcal{A}_{ee'}$ without its first element.

Finally, we can notice that our algorithm can take into account some classical constraints of the sequential pattern mining task. These constraints require additional parameters given by the expert. For example, it is possible to define a maximal/minimal size of chronicles, *i.e.* the number of events in their multiset. We can also use a maximal window constraint mwc to constraint events of chronicles to occurs together in a temporal window of maximal size mwc.

5 Experiments and results

We implemented a first prototype of our algorithm in C++ and we evaluate its efficiency on a real dataset of care-pathways.

5.1 Rational

The objective of our pharmaco-epidemiological study is to assess whether or not brand-to-generic antiepileptic drugs substitution is associated with seizure-related hospitalization. Our data represents 1,810,600 deliveries of 7,693 different drugs for 8,378 patients treated for epilepsy within a period from 03/02/2007 to 12/29/2011. We collected also 12,347 seizure-related hospitalizations on the same period concerning 7,754 patients.

In a first step, a naive algorithm abstracts drug deliveries into drug exposures. The algorithm transforms several point-based events $(e_1, \ldots, e_n)$, some drug deliveries, in a single interval-based event e, a drug exposure if 1) $n \geq rep_{min}$ and 2) two successive events are not spaced with more than gap_{max} time units. rep_{min} and gap_{max} are input parameters. We arbitrary choose to set gap_{max} to 30 and rep_{min} to 2 for our experiments.

To reduce the computing time of the TCB generation, we prefer to test our prototype on 10% percent of the original dataset corresponding to 839 care-pathways. In fact, the number of chronicles generated is not disturbed because

minimal support (σ_{min})	**mwc**	**#frequent chronicles**	**CPU time (s)**
84	90	463,006	440
84	91	917,952	537
84	92	1,506,243	730
84	93	1,868,923	808
84	94	2,596,401	1,053
84	95	3,878,436	1,479
85	90	342,598	231
86	90	246,724	209
87	90	173,872	167
167	180	1,861,607	958

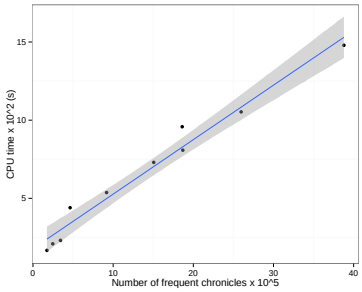

Fig. 3: Execution time results of our prototype. On the left, table of results ; On the right, CPU time wrt. number of frequent chronicles.

the minimal support constraint is defined as a percentage of the number of care-pathways. We constraint the generated chronicles to be ended by a specific event corresponding to an hospitalization for epileptic crisis. We are interested in the number of frequent chronicles generated (containing more than 2 events) and in the computing times.

5.2 Results

To distinguish the time to generate TCB and the time to extract frequent chronicles, we started to run this generation on our dataset with different couples of parameters. Parameters of this generation are a minimal support threshold σ_{min} and a maximal window constraint mwc. We ran 3 generations, one for $f_{min} = 10\%$ ($\sigma_{min} = 84$) and $mwc = 90$ days which generates 76,503 temporal constraints, an other for $f_{min} = 20\%$ and $mwc = 180$ days which generates 236,942 temporal constraints and a last one for $f_{min} = 20\%$ and $mwc = 20$ days which generates 0 temporal constraint. The computing time of the three generations is about 135 seconds. We can conclude that the generation time of the TCB only depends on the number and on the size of sequences but not on the parameters of the algorithm.

The Figure 3 illustrates computing times for different settings of our experiment. We can first notice that our algorithm generates millions of chronicles. Moreover, we precise that, for this dataset, all frequent chronicles have a multiset of events containing 3 events and that they are mainly specialization of the same multiset. By setting the minimal support threshold, we notice that the number of returned patterns is very sensitive to the maximal window constraint parameter. We next remark that the computing time is linear with the number of chronicles. If we only look at the settings which extract more than one million of chronicles, we observe that our algorithm can extract about 2300 chronicles per second.

6 Conclusion

Chronicles seem to be relevant to represent interesting patterns for pharmaco-epidemiology studies. Their expressiveness enables to model complex temporal behaviours of patients in the health care system (*e.g.* consultation, hospitalization and drugs delivery). In this article, we proposed a chronicle mining algorithms to the specificities of our database of sequences: sequences with interval-based events and sequences with repeated events. Our algorithm extracts the complete set of chronicles from a database of sequences. It has been implemented and evaluated on a real dataset of care-pathways. The experiments shown that our algorithm was able to generate very large numbers of chronicles.

We are now facing a classical pattern mining issue: the deluge of frequent patterns. Our main perspective is to tackle this issue. Several research directions can by studied, for instance, a heuristic to explore the search space or a method to extract a smaller set of chronicles like closed chronicles. Another way to reduce the number of frequent chronicles could be to consider as similar the chronicles with same multisets of event types and "similar" temporal constraint sets. Finally, visualization could help clinicians to define interesting patterns during the extraction, and the clinician's feedback could pilot the algorithm to the patterns he/she considers as more interesting.

Acknowledgements

This work is a part of the PEPS (Pharmaco-epidemiology of health products) funded by the French national agency for medicines and health products safety.

References

1. Álvarez, M.R., Félix, P., Cariñena, P.: Discovering metric temporal constraint networks on temporal databases. Artificial Intelligence in Medicine 58(3), 139–154 (2013)
2. Cram, D., Mathern, B., Mille, A.: A complete chronicle discovery approach: application to activity analysis. Expert Systems 29(4), 321–346 (2012)
3. Dousson, C., Duong, T.V.: Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 620–626 (1999)
4. Huang, Z., Lu, X., Duan, H.: On mining clinical pathway patterns from medical behaviors. Artificial Intelligence in Medicine 56(1), 35–50 (2012)
5. Quiniou, R., Cordier, M.O., Carrault, G., Wang, F.: Application of ILP to cardiac arrhythmia characterization for chronicle recognition. In: Proceeding of the conference on Inductive Logic Programming. pp. 220–227 (2001)
6. Subias, A., Travé-Massuyès, L., Le Corronc, E.: Learning chronicles signing multiple scenario instances. In: Proceedings of the 19th World Congress of the International Federation of Automatic Control. pp. 397–402 (2014)