J Intell Inf Syst (2011) 37:333–353
DOI 10.1007/s10844-011-0168-1

Extracting and summarizing the frequent emerging graph patterns from a dataset of graphs

Guillaume Poezevara · Bertrand Cuissart · Bruno Crémilleux

Received: 18 January 2010 / Revised: 19 July 2010 / Accepted: 23 June 2011 /
Published online: 16 July 2011

Abstract Emerging patterns are patterns of great interest for discovering information from data and characterizing classes. Mining emerging patterns remains a challenge, especially with graph data. In this paper, we propose a method to mine the whole set of frequent emerging graph patterns, given a frequency threshold and an emergence threshold. Our results are achieved thanks to a change of the description of the initial problem so that we are able to design a process combining efficient algorithmic and data mining methods. Moreover, we show that the closed graph patterns are a condensed representation of the frequent emerging graph patterns and we propose a new condensed representation based on the representative pruned graph patterns: by providing shorter patterns, it is especially dedicated to represent a set of graph patterns. Experiments on a real-world database composed of chemicals show the feasibility and the efficiency of our approach.

Keywords Data mining · Emerging patterns · Condensed representation · Subgraph isomorphism · Chemical information

1 Introduction

Discovering knowledge from large amounts of data and data mining methods are useful in a lot of domains such as chemoinformatics. One of the goals in chemoinformatics

G. Poezevara · B. Cuissart · B. Crémilleux (✉)
Laboratoire GREYC-CNRS UMR 6072, Université de Caen Basse-Normandie,
Caen, France
e-mail: bruno.cremilleux@unicaen.fr

G. Poezevara
e-mail: guillaume.poezevara@unicaen.fr

B. Cuissart
e-mail: bertrand.cuissart@unicaen.fr
URL:http://www.greyc.unicaen.fr/

is to establish relationships between molecules and a given activity (e.g., toxicity). Such a relationship may be characterized by graphs associating atoms and chemical bonds. The combinations of theses graphs are called graph *patterns* and are required in real-world data mining tasks such as the prediction of toxicity in chemoinformatics. A difficulty of this task is the number of potential patterns which is very large. By reducing the number of extracted patterns to those of a potential interest given by the user, the constraint-based pattern mining (Ng et al. 1998) provides efficient methods. A very useful constraint is the emerging constraint (Dong and Li 1999): emerging patterns (EPs) are patterns whose frequency strongly varies between two classes (the frequency of a pattern corresponds to the ratio of examples in the database supporting this pattern). It is a powerful measure to highlight contrasts between examples. EPs enable us to characterize classes (e.g., toxic versus non-toxic chemicals) both in a quantitative and qualitative way. EPs are at the origin of various works such as powerful classifiers (Li et al. 2001). From an applicative point of view, we can quote various works on the characterization of biochemical properties or medical data (Li and Wong 2001).

Even if a lot of progress has recently been made in the constraint-based pattern mining, mining EPs remains difficult because the anti-monotone property which is at the core of powerful pruning techniques in data mining (Mannila and Toivonen 1997) cannot be applied. As many data mining methods, the complexity of mining the complete and correct set of EPs is exponential in the number of items in the worst case. As EPs are linked to the pattern frequency, naive approaches for mining EPs extract frequent patterns in a class and infrequent patterns in the set of the other classes because the frequency and infrequency constraints satisfy (anti-)monotone properties and therefore there are techniques to mine such a combination of constraints. Unfortunately, such an approach only extracts a subset of the whole set of EPs. That is why some techniques use handlings of borders but it is very expensive (Dong and Li 1999). In the context of patterns made of items (i.e., database objects are described by items), an efficient method based on a prefix-freeness operator leading to interval pruning was proposed (Soulet and Crémilleux 2009; Soulet et al. 2007). More generally, most of the works on EPs are devoted to the itemset area and there are very few attempts in areas such as chemoinformatics where chemicals are modeled by graphs (Borgelt et al. 2005; De Raedt and Kramer 2001). These last two works are based on a combination of monotone and anti-monotone constraints and they do not extract the whole collection of EPs and are limited in patterns of length 1. Note that in the case of EP of size 1, there exist anti-monotone measures for convex statistical functions, such as chi-square (Morishita et al. 2000). Mining patterns in a graph dataset is a much more challenging task than mining patterns in itemsets.

In this paper, we tackle this challenge of mining emerging graph patterns. Our main contribution is to propose a method mining all frequent emerging graph patterns. This result is achieved by a change of the description of the initial problem in order to be able to use efficient algorithmic and data mining methods (see Section 3). We formally prove that these two problems are equivalent. Among other results, all frequent connected emerging graphs are produced; they correspond to the patterns of cardinality 1. These graphs are useful because they are the most understandable graphs from the chemical point of view. The patterns of cardinality greater than 1 capture the emerging power of associations of connected graphs. A great feature of our method is to be able to extract *all* frequent emerging graph patterns (given a frequency threshold and an emergence threshold) and not only particular EPs. We also deal with the pattern condensed representation issue (Yan and Han 2003) and we show that the closed graph patterns are a condensed representation of the frequent emerging graph patterns. Moreover, we propose a new condensed representation based on the representative pruned graph patterns: by providing shorter patterns, it is especially dedicated to represent a set of graph patterns. Finally, we present several experiments providing quantitative results on our method and a case study on a chemical database provided by the Environment Protection Agency. This experiment shows the feasibility of our approach and suggests promising chemical investigations on the discovery of toxicophores (Lozano et al. 2010). This paper extends the preliminary version (Poezevara et al. 2009) by further results such as the proposition of the condensed representation of the frequent emerging patterns, the formal proof of the equivalence of the change of the description of the initial problem, the mining method has been improved by testing the subgraph isomorphism during the mining process and further experiments.

This paper is organized as follows. Section 2 outlines preliminary definitions and related work. Our method for mining all frequent emerging graph patterns and results on the condensed representation of the frequent emerging patterns are described in Section 3. Experiments showing the efficiency of our approach and results on the chemical dataset are given in Section 4.

2 Context and motivations

2.1 Notation and definitions

Graph terminology In this text, we consider simple labeled graphs. We recall here some important notions related to these graphs. A *graph* $G(V, E)$ consists of two sets V and E. An element of V is called a *vertex* of G. An element of E is called an *edge* of G, an edge corresponds to a pair of vertices. Two edges are *adjacent* if they share a common vertex. A *walk* is a sequence of edges such that two consecutive edges are adjacent. A graph G is *connected* if any two of its vertices are linked by a walk. Two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are *isomorphic* if there exists a bijection $\psi : V_1 \longrightarrow V_2$ such that for every $u_1, v_1 \in V_1$, $\{u_1, v_1\} \in E_1$ if and only if $\{\psi(u_1), \psi(v_1)\} \in E_2$; ψ is called an *isomorphism*. Given two graphs $G'(V', E')$ and $G(V, E)$, G' is a *subgraph* of G if (a) V' is a subset of V and E' is a subset of E or if (b) G' is isomorphic to a subgraph of G. Given a family of graphs $\mathcal{D}$ and a frequency threshold $f_{\mathcal{D}}$, a graph G is a *frequent subgraph (of $(\mathcal{D}, f_{\mathcal{D}})$)* if G is a subgraph of at least $f_{\mathcal{D}}$ graphs of $\mathcal{D}$; a *frequent connected subgraph* is a frequent subgraph that is connected.

Graphs encountered in the text carry information by the meaning of *labellings* of the vertices and of the edges. The labellings do not affect the previous definitions, except that an isomorphism has to preserve the labels. A *molecular graph* is a labelled graph that depicts a chemical structure: a vertex represents an atom, an edge represents a chemical bond. Figure 1 displays molecular graphs. The graph SG_1 (see Fig. 2) is (isomorphic to) a subgraph of molecule 2 in Fig. 1 and therefore its frequency is 0.16 (1 molecule among 6 supports SG_1). Let us assume that $\mathcal{D}$ is partitioned into two subsets (or classes) $\mathcal{D}_1$ and $\mathcal{D}_2$. For instance, in Fig. 1, its left

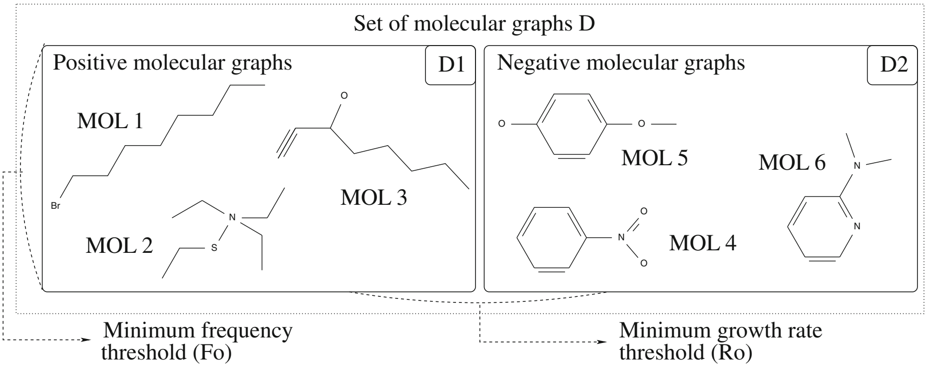

Set of molecular graphs D
Positive molecular graphs
D1
MOL 1
O
Br
MOL 3
N
MOL 2
S
Negative molecular graphs
D2
O
O
MOL 5
MOL 6
N
O
N
O
MOL 4
N
Minimum frequency threshold (Fo)
Minimum growth rate threshold (Ro)

Fig. 1 Molecules excerpted from the EPAFHM database (EPAFHM 2008)

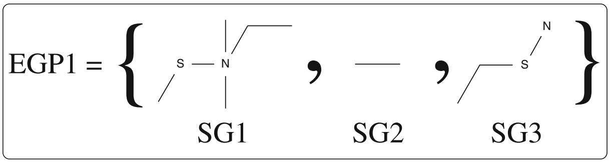
part $\mathcal{D}_1$ gathers positive molecules and its right part $\mathcal{D}_2$ negative molecules. With a minimum frequency threshold of 0.33, SG_1 is a frequent graph in $\mathcal{D}_1$ (1 molecule among 3 supports SG_1 thus its frequency is 1/3) but it is not a frequent graph in $\mathcal{D}_2$ (0 molecule among 3 supports SG_1).

The problem of mining all the frequent connected subgraphs of ($\mathcal{D}$, $f_{\mathcal{D}}$) is called the *discovery of the Frequent Connected SubGraphs (FCSG)*. It relies on multiple subgraph isomorphisms. Given a couple of graphs (G', G), the problem of deciding whether G' is isomorphic to a subgraph of G is named the *Subgraph Isomorphism Problem (SI)*. *SI* is NP-complete (Garey and Johnson 1979, p. 64). The problem remains NP-complete if we restrict the input to connected graphs. Consequently, the discovery of the *FCSGs* is NP-Complete. The labellings do not change the class of complexity of *SI* and the discovery of the *FCSGs*.

In the following, a *graph pattern* denominates a set of connected graphs. Let $\mathcal{G}$ be a graph pattern and $\mathcal{D}$ be a set of graphs. $\mathcal{F}(\mathcal{G}, \mathcal{D})$ denotes the graphs of $\mathcal{D}$ that include every graph of $\mathcal{G}$ as a subgraph ($\mathcal{F}(\mathcal{G}, \mathcal{D}) = \{G_{\mathcal{D}} \in \mathcal{D} : \forall G \in \mathcal{G},\ G \text{ is a subgraph of } G_{\mathcal{D}}\}$). Given a minimum frequency threshold f, the graph pattern $\mathcal{G}$ is *frequent* in $\mathcal{D}$ if $\frac{|\mathcal{F}(\mathcal{G},\mathcal{D})|}{|\mathcal{D}|} \geq f$ (we use here a relative frequency threshold). In Fig. 2, the graph pattern made of SG_1, SG_2 and SG_3 has a frequency of 0.16 in $\mathcal{D}$ (it is included in molecule 2). In this paper, a graph pattern is composed of *connected* graphs.

Emerging Graph Pattern (EGP) As introduced earlier, an emerging graph pattern $\mathcal{G}$ is a set of graphs whose frequency increases significantly from one subset (or class)

Fig. 2 Example of an Emerging Graph Pattern ($f_{\mathcal{D}_1} = 0.33$, $GR_{\mathcal{D}_1}(EGP1) = \infty$)



to another. The capture of contrast brought by $\mathcal{G}$ from $\mathcal{D}_2$ to $\mathcal{D}_1$ is measured by its *growth rate* $GR_{\mathcal{D}_1}(\mathcal{G})$ defined as:

$$\begin{cases} 0, & \text{if } \mathcal{F}(\mathcal{G}, \mathcal{D}_1) = \emptyset \text{ and } \mathcal{F}(\mathcal{G}, \mathcal{D}_2) = \emptyset \\ \infty, & \text{if } \mathcal{F}(\mathcal{G}, \mathcal{D}_1) \neq \emptyset \text{ and } \mathcal{F}(\mathcal{G}, \mathcal{D}_2) = \emptyset \\ \frac{|\mathcal{D}_2| \times |\mathcal{F}(\mathcal{G}, \mathcal{D}_1)|}{|\mathcal{D}_1| \times |\mathcal{F}(\mathcal{G}, \mathcal{D}_2)|}, & \text{otherwise } (|.| \text{ denotes the cardinality of a set}) \end{cases}$$

Therefore, the definition of an EGP is given by:

Definition 1 (Emerging Graph Pattern) Let $\mathcal{D}$ be a set of graphs partitioned into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$. Given a growth threshold ρ, a set of connected graphs $\mathcal{G}$ is an *emerging graph pattern* from $\mathcal{D}_2$ to $\mathcal{D}_1$ if $GR_{\mathcal{D}_1}(\mathcal{G}) \geq \rho$

We now define the problem of mining the whole set of the frequent EGPs; this definition constitutes the terms of the problem handled by the text.

Definition 2 (Frequent Emerging Graph Pattern Extraction)

Input $\mathcal{D}$ a set of graphs partitioned into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$, $f_{\mathcal{D}_1}$ a frequency threshold in $\mathcal{D}_1$ and ρ a growth threshold

Output the set of the frequent emerging graph patterns from $\mathcal{D}_2$ to $\mathcal{D}_1$ with their growth rate and their frequency according to $f_{\mathcal{D}_1}$ and ρ such that $| \mathcal{F}(\mathcal{G}, \mathcal{D}_1) | \geq f_{\mathcal{D}_1}$ and $GR_{\mathcal{D}_1}(\mathcal{G}) \geq \rho$.

The *length* of a graph pattern denotes its cardinality. Note that the set of frequent EGPs of length 1 from $\mathcal{D}_2$ to $\mathcal{D}_1$ corresponds to the set of frequent emerging connected graphs from $\mathcal{D}_2$ to $\mathcal{D}_1$. On Fig. 2, EGP_1 has a length of 3. For the sake of simplicity, the definitions are given with only two classes but all the results hold with more than two classes (it is enough to consider that $\mathcal{D}_2 = \mathcal{D} \backslash \mathcal{D}_1$, as usual in the EP area (Dong and Li 1999)).

2.2 Related work: extraction of discriminative graphs

Several methods have been designed for discovering graphs that are correlated to a given class. All these algorithms operate on a set of graphs partitioned into two classes called positive graphs and negative graphs.

Molfea (Kramer et al. 2001) uses a level-wise algorithm (Mannila and Toivonen 1997) enabling the extraction of *linear subgraphs (chains)* which are frequent in a set of "positive" graphs and infrequent in a set of "negative" graphs. However, the restriction to linear subgraphs disables a direct extraction of the graphs containing a branching point or a cycle.

Moss (Borgelt and Berthold 2002; Borgelt et al. 2005) is a program dedicated to mine molecular substructure; it can be extended to find the *discriminative fragments*. Given two frequency thresholds f_M and f_m, a discriminative fragment corresponds to a connected subgraph whose frequency is above f_M in a set of "positive" graphs and is below f_m in a set of "negative" graphs. This definition differs from the usual notion of emergence which is based on the growth rate as introduced in the previous section. Note that the set of the discriminative fragments according to the thresholds f_M and f_m does not contain the whole set of the frequent EGPs having a growth rate higher than f_M/f_m or any other given growth rate threshold. Moreover, such fragments only corresponds to EPs of length 1. On the contrary, we will see that our approach follows the usual notion of emergence.

Another work has been dedicated to the discovery of the *contrast subgraphs* (Ting and Bailey 2006). A contrast subgraph is a graph that appears in the set of the "positive" graphs but never in the set of the "negative" graphs. Although this notion is very interesting, it requires a lot of computation. To the best of our knowledge, the calculus is limited to one "positive" graph and the mining of a graph exceeding 20 vertices brings up a significant challenge. Furthermore, contrast subgraphs correspond to *jumping emerging patterns* (i.e., EPs with a growth rate equals ∞) and therefore are a specific case of the general framework of EPs.

3 Mining frequent emerging graph patterns

This section explains our method to extract the set of the frequent emerging graph patterns, as defined in Definition 2. We start by introducing the new context of description of the input dataset and giving the formal proof of the equivalence of the change of the description of the initial problem. Then, our mining method is detailed. Finally, we show that the closed graph patterns are a condensed representation of the frequent emerging graph patterns and we propose a new condensed representation based on the representative pruned graph patterns.

3.1 An equivalent context for extracting the frequent graph patterns

The change of the description of the initial problem is a key idea of our method to mine all the frequent emerging graph patterns. It brings two meaningful advantages. First, the new descriptors, by being frequent subgraphs, strongly reduce the search space and therefore the number of candidate patterns. Second, it enables us to set the problem in an itemset context from which we can reuse efficient results on the emerging constraint. We start by giving the new description of the input graphs.

Let $\mathcal{D} = \{G_1, \ldots, G_n\}$ be a set of graphs, considered as the input dataset. Let $\mathcal{A} = \{a_1, \ldots, a_m\}$ be a set of graphs, considered as the attributes. The binary *description of an element, G_i, based on the occurrences of $\mathcal{A}$* is a sequence of digits $d_i = (d_{(i,j)}, 1 \leq j \leq m : d_{(i,j)} = 1$ if the attribute graph a_j is a subgraph of G_i, $d_{(i,j)} = 0$ otherwise). We extend this notion to a binary *description of a set of graphs $\mathcal{D}$, based on the occurrences of $\mathcal{A}$*; it corresponds to the dataset $\mathcal{D}' = \{d_i, 1 \leq i \leq n\}$ where every d_i is the description of the corresponding G_i based on the occurrences of $\mathcal{A}$.

As an example, we consider the set of molecular graphs depicted in Fig. 1 as being the input dataset and the set of graphs depicted in Fig. 2 as being the set of attributes: $\mathcal{D} = \{MOL_1, \ldots, MOL_6\}$ and $\mathcal{A} = (SG_1, SG_2, SG_3)$. As SG_1, SG_2 and SG_3 are subgraphs of MOL_2, the binary description of MOL_2 based on the occurrences of (SG_1, SG_2, SG_3) is $(1, 1, 1)$. Table 1 gives the binary description of $\mathcal{D}$ based on (SG_1, SG_2, SG_3) .

The datasets $\mathcal{D}$ and $\mathcal{D}'$ are considered as multi-sets: we assume here that every input graph represents one element of $\mathcal{D}$. It means that when a same input graph appears twice in $\mathcal{D}$ (G_i is isomorphic to G_j with $i \neq j$), we consider G_i and G_j as two

Table 1 Binary description of the set of graphs shown on Fig. 1 based on the graphs of Fig. 2

Graphs	SG_1	SG_2	SG_3
MOL_1	0	1	0
MOL_2	1	1	1
MOL_3	0	1	0
MOL_4	0	0	0
MOL_5	0	0	0
MOL_6	0	0	0

elements of $\mathcal{D}$. In a similar way, we consider every description of an input graph as being one element of $\mathcal{D}'$. Consequently, we always have $|\mathcal{D}| = |\mathcal{D}'|$.

Proposition 1 (Equivalent context for mining the Frequent Graph Patterns) *Let $\mathcal{D}$ be a set of graphs partitioned into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$, $f_{\mathcal{D}_1}$ a frequency threshold in $\mathcal{D}_1$ and ρ a growth threshold from $\mathcal{D}_2$ to $\mathcal{D}_1$. Let $\mathcal{A}$ be the set of the frequent subgraphs in $\mathcal{D}_1$: $\mathcal{A} = \{a : \frac{|\mathcal{F}(a,\mathcal{D}_1)|}{|\mathcal{D}_1|} \geq f_{\mathcal{D}_1}\}$.*

A graph pattern $\mathcal{G} = \{g_1, \ldots, g_p\}$ is a frequent emerging graph pattern from $\mathcal{D}_2$ to $\mathcal{D}_1$ if $\{g_1, \ldots, g_p\}$ is an emerging pattern from $\mathcal{D}_2$ to $\mathcal{D}_1$ in the description of $\mathcal{D}$ based on the occurrences of $\mathcal{A}$.

Proof of the proposition As a graph pattern occurs in a graph G if all its elements are subgraphs of G, we have the following lemma.

Lemma 1 (A frequent graph pattern is constituted of frequent graphs) *Let $\mathcal{D}$ be a dataset of graphs and $\mathcal{G} = \{g_1, \ldots, g_p\}$ be a graph pattern. We have : $\mathcal{F}(\mathcal{G}, \mathcal{D}) \subseteq \mathcal{F}(g_i, \mathcal{D}), \forall i \in 1, \ldots p$.*

Consequently, a frequent emerging graph pattern is constituted only by frequent connected graphs. We go on with the notations of Proposition 1 (i.e., $\mathcal{A}$ is the set of the frequent graphs in $\mathcal{D}_1$) and we assume that $\mathcal{D}'$ corresponds to the description of the set of graphs $\mathcal{D}$ based on the occurrences of $\mathcal{A}$. We also note $\mathcal{D}'_1$ (resp. $\mathcal{D}'_2$) the description of $\mathcal{D}_1$ (resp. $\mathcal{D}_2$) based on the occurrences of $\mathcal{A}$. Thanks to the lemma, $\mathcal{G} = \{g_1, \ldots, g_p\}$ being a frequent emerging graph pattern implies that g_i is a frequent graph in $\mathcal{D}_1$, $\forall 1 \leq i \leq p$. Consequently, $\{g_1, \ldots, g_p\}$ is a frequent pattern of $\mathcal{D}'_1$. By construction, we have: $\mathcal{F}(\mathcal{G}, \mathcal{D}) = \mathcal{F}(\{g_1, \ldots, g_p\}, \mathcal{D}')$, $\mathcal{F}(\mathcal{G}, \mathcal{D}_1) = \mathcal{F}(\{g_1, \ldots, g_p\}, \mathcal{D}'_1)$ and $\mathcal{F}(\mathcal{G}, \mathcal{D}_2) = \mathcal{F}(\{g_1, \ldots, g_p\}, \mathcal{D}'_2)$. The proposition is an immediate consequence of these equalities.

Consequences The description of $\mathcal{D}$ based on the set of the frequent subgraphs in $\mathcal{D}_1$ together with the previous proposition lead to an efficient method of computation. The relation of inclusion defined between two sets can naturally be used as a specialization relation in the context of the graph patterns. Let $\mathcal{G}$ and $\mathcal{G}'$ be two graph patterns. $\mathcal{G}'$ is included in $\mathcal{G}$ if for any element g' of $\mathcal{G}'$, there exists an element g of $\mathcal{G}$ such that g' is isomorphic to g and $g \in \mathcal{G}$. We note the inclusion of $\mathcal{G}'$ in $\mathcal{G}$ by $\mathcal{G}' \subseteq \mathcal{G}$. With this relation, the frequency satisfies the anti-monotone property (Mannila and Toivonen 1997) (i.e., $\mathcal{G}' \subseteq \mathcal{G}$ implies that $\mathcal{F}(\mathcal{G}, \mathcal{D}) \subseteq \mathcal{F}(\mathcal{G}', \mathcal{D})$) whereas the emergence does not satisfy it. Consequently, as the pruning only relies on the frequency, the successive generation of every candidate graph pattern that is required in order to check if a candidate satisfies the two constraints cannot be applied due to the huge number of candidates.

Proposition 1 ensures that the frequent emerging graph patterns may be derived from the set from the frequent subgraphs in $\mathcal{D}_1$. As these descriptors must satisfy a frequency threshold, we can benefit from the pruning properties coming from the frequency early in the calculation. It leads to our method described in the following section.

3.2 The method and its implementation

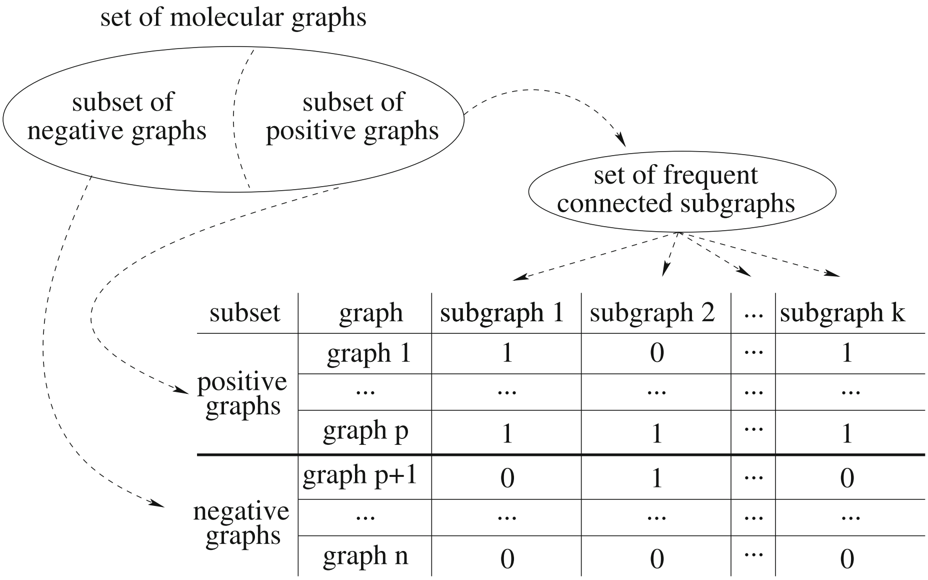
Our method consists in a succession of the three following steps:

- extracting the frequent connected subgraphs in $\mathcal{D}_1$ according to the frequency threshold $f_{\mathcal{D}_1}$. This is the *FCSG* problem defined in Section 2.1.
- for each graph $G_\mathcal{D}$ of $\mathcal{D}$, we recode $G_\mathcal{D}$ according to the set of connected graphs resulting from the previous step: each row of the dataset is a graph G of $\mathcal{D}$ and each column indicates whether a frequent connected graph extracted at the first step is present in G or not.
- the problem is then described by items (presence or absence of each frequent connected graph) and we are able to use an efficient method (i.e., MUSIC-DFS) based on itemsets to discover the frequent emerging graph patterns.

Extraction of the frequent subgraphs of $\mathcal{D}$ Graph mining tools for extracting frequent connected subgraphs generate the candidate graphs according to a specialization relation (such as the inclusion relation presented above). The search space is then pruned thanks to the anti-monotone property of the of the frequency. Based on this principle, there are several methods to solve the *FCSG* problem and the algorithms are classified into two families: the *Apriori-Based* algorithms and the *Pattern-Growth-Based* algorithms. The two families have been compared for mining sets of chemical graphs (Cook and Holder 2006): the Apriori-Based algorithms spend less time while the Pattern-Growth-Based algorithms consume less memory. A comparison of four Pattern-Growth-Based algorithms has been conducted in Wörlein et al. (2005). For mining a set of chemical graphs, Gaston (Nijssen and Kok 2004) runs faster than the other ones. The efficiency of Gaston mainly relies on the adoption of the *quick-start principle*. Gaston first extracts the frequent paths (the connected graphs made only with vertices of degree 1 or 2), then it extracts the frequent trees (the connected acyclic graphs), finally it extracts the frequent graphs. Each step of the process uses the results of the previous extraction to perform an efficient pruning. Moreover, Gaston is available on http://www.liacs.nl/home/snijssen/gaston/ under GNU GPL License Version 2.[1] For these reasons, we have chosen Gaston for extracting frequent connected subgraphs.

Determining the support of the frequent connected subgraphs Figure 3 illustrates the change of the description from the input graph dataset to its description based on

[1] http://www.gnu.org/licenses/gpl/html


set of molecular graphs
subset of negative graphs
subset of positive graphs
set of frequent connected subgraphs

subset	graph	subgraph 1	subgraph 2	···	subgraph k
positive graphs	graph 1	1	0	···	1
	...	...	...	...	...
	graph p	1	1	···	1
negative graphs	graph p+1	0	1	···	0
	...	...	...	...	...
	graph n	0	0	···	0

Fig. 3 Converting a graph dataset into its description based on the frequent connected subgraphs

the frequent connected subgraphs of $\mathcal{D}_1$. For every extracted frequent subgraph, this change of representation requires to know the graphs of $\mathcal{D}$ that include it. Once we have extracted the frequent connected subgraphs in $\mathcal{D}_1$, we may compute the description of $\mathcal{D}$ based on the set of the extracted subgraphs. This may correspond to the computation of a huge number of subgraph isomorphism ($|\mathcal{D}| \times |\{\text{extracted subgraph}\}|$). As it is, this calculation turns very hard to achieve even if the input graphs are small and even if we use an up-to-date tool for solving the subgraph isomorphism problem such as VFLib (Cordella et al. 1999). We now explain how we circumvent this difficulty.

A general subgraph isomorphism method treats the problem with two graphs as input (a graph and a target graph) and answers the question: is the graph a subgraph of the target graph? When it processes a subgraph isomorphism, a graph mining tool takes as input a graph and a target graph as well as all the embeddings of a large subgraph of the graph into the target graph. This supplementary knowledge drastically simplifies the problem and we use it in our work. The following example illustrates this improvement.

To decide whether a studied graph is a subgraph of a target graph, an algorithm dedicated of subgraph isomorphisms tries to embed every vertex and every edge of the studied graph into the target graph, preserving the adjacency relationship. Such an algorithm does not have to memorize its last solved problems. A graph mining tool, like Gaston, traverses the space of graphs in a rigourous manner like the pattern growth approach. It takes advantage of the last subgraph isomorphisms solved: the latters facilitate the next isomorphisms to proceed. For example, suppose that a graph mining tool has to sucessively determine whether the graphs 1, 2 and 3 depicted on Fig. 4 are subgraphs of the target graph. When it processes graph1 and the target

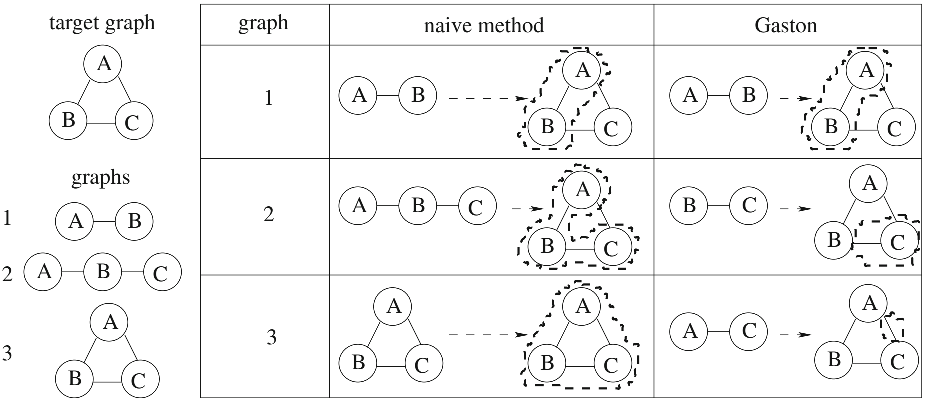


Fig. 4 A single call to a graph mining tool is more efficient than multiple, independent calls to a subgraph isomorphism tool

graph, such a tool memorizes all the embeddings it finds. When it will have to process graph2 and the target graph (suppose graph2 comes just after graph1 in the traversal of the space of graphs), the graph mining tool will just have to check whether each of the memorized embeddings between graph1 and the target graph can lead to an embedding between graph2 and the target graph. The embeddings between graph2 and the target graph will then be memorized and, later, they will be used to process graph3 and the target graph. As a conclusion, to get the embeddings of a "well organized" family of graphs into a database of graphs, one call of a graph mining tool may be, by far, more efficient than multiple, independent calls of a subgraph isomorphism tool.

In order to solve an instance of Frequent Emerging Graph Pattern Extraction, we first call a graph mining tool to extract the set of the frequent connected subgraphs in $\mathcal{D}_1$. As the whole database of input graphs (both $\mathcal{D}_1$ and $\mathcal{D}_2$.) will have to be described based on this set of frequent connected subgraphs, we also use this call of a graph mining tool to determine the occurrences of each frequent connected subgraphs in every graph of $\mathcal{D}_2$. That way, we take advantage of the fact that one call to a graph mining tool is more efficient than multiple, independent calls to a subgraph isomorphism tool.

Extracting the frequent emerging (graph) patterns Frequent emerging graph patterns are mined by using Music-dfs.[2] This tool offers a set of syntactic and aggregate primitives to specify a broad spectrum of constraints in a flexible way, for data described by items (Soulet et al. 2007). Then Music-dfs mines soundly and completely all the patterns satisfying a given set of input constraints. The efficiency of Music-dfs lies in its depth-first search strategy and a safe pruning of the pattern space by pushing the constraints. The constraints are applied as early as possible. The

[2] http://www.info.univ-tours.fr/~soulet/music-dfs/music-dfs.html

pruning conditions are based on intervals. Here, an *interval* denominates a set of patterns that include a same prefix-free pattern and that are included in the prefix-closure of this pattern (see Soulet et al. 2007, for more details). Whenever it is computed that all the patterns included in an interval simultaneously satisfy (or not) the constraint, the interval is positively (negatively) pruned without enumerating all its patterns (Soulet et al. 2007). The output of Music-dfs enumerates the intervals satisfying the constraint. Such an interval condensed representation improves the output legibility and each pattern appears in only one interval. In our context, this tool enables us to use the emerging and frequency constraints.

Finally, our approach ensures to produce the whole set of frequent EGPs because the *FCSG* step extracts all the connected subgraphs and Music-dfs is complete and correct for the pattern mining step.

3.3 A condensed representation of the frequent emerging graph patterns

As said earlier, the number of extracted patterns may be large and many works propose methods to reduce the collection of patterns, such as the constraint-based paradigm previously introduced or the so-called condensed representations (Calders et al. 2005). Indeed, even if constraints such as the emergence and the frequency reduce the number of resulting patterns, we can go further thanks to the notion of pattern condensed representations. The key principle of the pattern condensed representations with respect to a constraint is to mine a set of patterns as concise as possible from which the whole set of patterns satisfying the constraint can be efficiently derived. Whereas there are many propositions for data described by items (Calders et al. 2005), including the constraint of emergence (Soulet et al. 2005), condensed representations on sequences or graphs mainly address the frequency constraint and are based on the closed patterns (Plantevit and Crémilleux 2009; Yan and Han 2003). In this section, we show that the closed graph patterns are a condensed representation of the frequent emerging graph patterns and we propose a new condensed representation based on the *representative pruned graph patterns*.

Let $\mathcal{D}$ be a dataset of graphs. The description of $\mathcal{D}$ is based on the occurrences of a *finite* set of graphs $\mathcal{A}$ (see Section 3.1). A graph pattern denominates any subset of $\mathcal{A}$. Recalling the inclusion relation: a graph pattern $\mathcal{G}'$ is included in a graph pattern $\mathcal{G}$ if every element of $\mathcal{G}'$ is (isomorphic to) an element of $\mathcal{G}$: $\mathcal{G}' \subseteq \mathcal{G}$ if $\forall g' \in \mathcal{G}', \ \exists g \in \mathcal{G}$ such that g' is isomorphic to g.

Definition 3 (Closed graph pattern) A graph pattern $\mathcal{G}$ is a *closed graph pattern* in $\mathcal{D}$ if $\forall \mathcal{G}'$ a graph pattern, $\mathcal{F}(\mathcal{G}, \mathcal{D}) = \mathcal{F}(\mathcal{G}', \mathcal{D})$ implies that $\mathcal{G}' \subseteq \mathcal{G}$.

Proposition 2 follows:

Proposition 2 (Existence and uniqueness of a closure) *Let $\mathcal{G}'$ be a graph pattern. There exists a unique closed graph pattern $\mathcal{G}$ such that $\mathcal{F}(\mathcal{G}', \mathcal{D}) = \mathcal{F}(\mathcal{G}, \mathcal{D})$. $\mathcal{G}$ is the* closure *of $\mathcal{G}'$, denoted by $\overline{\mathcal{G}'} = \mathcal{G}$.*

Figure 5 displays the graph pattern CGP1 which is the closure of the emerging graph pattern EGP1 displayed on Fig. 2.

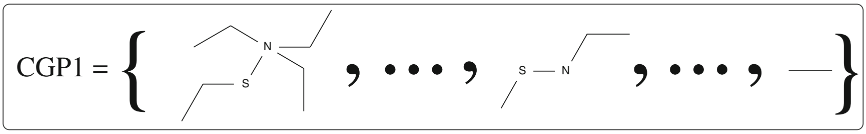


Fig. 5 Example of a Closed Graph Pattern

Sketch of proof

(existence) Let $\mathcal{G}'$ be a graph pattern. One of the two mutually exclusive situations happens:

(i) For all graph patterns $\mathcal{G}$, either $\mathcal{G} \subseteq \mathcal{G}'$ or $\mathcal{F}(\mathcal{G}', \mathcal{D}) \neq \mathcal{F}(\mathcal{G}, \mathcal{D})$. In this situation, $\mathcal{F}(\mathcal{G}', \mathcal{D}) = \mathcal{F}(\mathcal{G}, \mathcal{D})$ implies $\mathcal{G} \subseteq \mathcal{G}'$. $\mathcal{G}'$ is a closed graph pattern.

(ii) (negation of i)) There exists a graph pattern $\mathcal{G}$ (different from $\mathcal{G}'$) such that $\mathcal{G}' \subsetneq \mathcal{G}$ and $\mathcal{F}(\mathcal{G}', \mathcal{D}) = \mathcal{F}(\mathcal{G}, \mathcal{D})$. In this situation, the existence of a closed graph pattern representing $\mathcal{G}'$ depends on the existence of a closed graph pattern representing $\mathcal{G}$. We iterate the process by substituting $\mathcal{G}$ for $\mathcal{G}'$. As the length $\mathcal{G}$ is strictly greater than the length of $\mathcal{G}'$, the process terminates in situation i) after a finite number of iterations. As a conclusion, there exists a closed graph pattern $\mathcal{G}$ such that $\mathcal{F}(\mathcal{G}', \mathcal{D}) = \mathcal{F}(\mathcal{G}, \mathcal{D})$.

(uniqueness) Definition 3 indicates that if $\mathcal{G}_1$ and $\mathcal{G}_2$ are two distinct closed graph patterns then $\mathcal{F}(\mathcal{G}_1, \mathcal{D}) \neq \mathcal{F}(\mathcal{G}_2, \mathcal{D})$ (either $\mathcal{F}(\mathcal{G}_1, \mathcal{D}) \subseteq \mathcal{F}(\mathcal{G}_2, \mathcal{D})$ or $\mathcal{F}(\mathcal{G}_2, \mathcal{D}) \subseteq \mathcal{F}(\mathcal{G}_1, \mathcal{D})$ does not hold). The uniqueness is immediate.

Consequently, a set of graph patterns – a family of sets over $\mathcal{A}$– can be partitioned according to their closures in $\mathcal{D}$, as soon as the set contains the closure of all its elements. We name the subsets given by this partition as the *subsets induced by the closed graph patterns*. We now define the notion of a proper set for a representation by its closed elements. Basically, a proper set is a set that can be represented by its closed patterns. We will show that a proper set is suitable to condense the frequent emerging graph patterns.

Definition 4 (A proper set for a representation by its closed elements) Let $\mathcal{P}$ be a set of graph patterns. $\mathcal{P}$ is a *proper set for a representation by its closed elements* if for any pair of graph patterns $\mathcal{G}$ and $\mathcal{G}'$ such that $\mathcal{F}(\mathcal{G}', \mathcal{D}) = \mathcal{F}(\mathcal{G}, \mathcal{D})$, $\mathcal{G} \in \mathcal{P}$ is equivalent to $\mathcal{G}' \in \mathcal{P}$.

A consequence of Definition 4 is the following: with a proper set for a representation by its closed elements $\mathcal{P}$, as soon as one graph pattern belongs to $\mathcal{P}$, any graph pattern that shares its extension (its support) in $\mathcal{D}$ also belongs to $\mathcal{P}$. By definition of $\mathcal{P}$, for all pair of graph patterns $\mathcal{G}$ and $\mathcal{G}'$ such that $\mathcal{F}(\mathcal{G}', \mathcal{D}) = \mathcal{F}(\mathcal{G}, \mathcal{D})\}$, we have $\mathcal{F}(\mathcal{G} \cup \mathcal{G}', \mathcal{D}) = \mathcal{F}(\mathcal{G}', \mathcal{D}) = \mathcal{F}(\mathcal{G}, \mathcal{D})$. The following proposition is an immediate consequence of this remark:

Proposition 3 *Let $\mathcal{P}$ be a set of graph patterns. If $\mathcal{P}$ is a proper set for a representation by its closed elements then the union is a closed operation on the subsets of $\mathcal{P}$ induced by its closed graph patterns.*

As a consequence of Proposition 3, a proper set for a representation by its closed elements may be summarized by its closed elements without loss of information. For any pair of graph patterns $\mathcal{G}$ and $\mathcal{G}'$ such that $\mathcal{F}(\mathcal{G}', \mathcal{D}) = \mathcal{F}(\mathcal{G}, \mathcal{D})$, the proposition "$\mathcal{G}$ is a frequent emerging graph pattern" is equivalent to "$\mathcal{G}'$ is a frequent emerging graph pattern". Consequently, the set of the frequent emerging graph patterns is always a proper set for a representation by its closed elements and thus *the set of the frequent emerging graph patterns is condensed by its subset of closed patterns.*

The previous results of this section can be seen as an extension to the graph patterns of the results on the condensed representations on itemsets or sequences (Calders et al. 2005; Plantevit and Crémilleux 2009). In the following, we show that with graph patterns, the condensed representation based on the closed patterns can be improved by providing shorter patterns that we call representative pruned graph patterns.

The relation "being a subgraph" satisfies the anti-monotone property with respect to the inclusion of support: if a graph g' is a subgraph of a graph g then $\mathcal{F}(\{g\}, \mathcal{D}) \subseteq \mathcal{F}(\{g'\}, \mathcal{D})$. The following proposition is an immediate consequence of this remark:

Proposition 4 (Constraint brought by the addition of a graph to a graph pattern) *Let $\mathcal{G}$ be a graph pattern and g' be a connected graph. If g' is a subgraph of an element of $\mathcal{G}$ then $\mathcal{F}(\mathcal{G} \cup \{g'\}, \mathcal{D}) = \mathcal{F}(\mathcal{G}, \mathcal{D})$.*

It states that adding graphs to a graph pattern $\mathcal{G}$ maintains its support as soon as the added graphs are subgraphs of $\mathcal{G}$. Combining Proposition 4 and the fact that a closed graph pattern can not be included in a distinct graph pattern with the same support, we straightforwardly have the proposition:

Proposition 5 (Composition of a closed graph pattern) *Let $\mathcal{G}$ be a graph pattern and g' be a connected graph. If $\mathcal{G}$ is a closed graph pattern, then the following property is true: for any couple of graphs (g,g'), if $g \in \mathcal{G}$ and g' is a connected subgraph of g then $g' \in \mathcal{G}$.*

We now define the notion of a *representative pruned graph pattern*:

Definition 5 (A representative pruned graph pattern) A graph pattern $\mathcal{G}$ is a *representative pruned graph pattern* if (i) no element of $\mathcal{G}$ is a subgraph of another element of $\mathcal{G}$ and (ii) the graph pattern obtained by adding all the connected subgraphs of every element of $\mathcal{G}$ is a closed graph pattern.

It is important to note that it is always possible to construct a representative pruned graph pattern from a closed graph pattern $\mathcal{G}$: it is enough to remove every element of $\mathcal{G}$ that is a subgraph of another element of $\mathcal{G}$.

We now show that a representative pruned graph pattern has the same support as its corresponding closed pattern. Let $\mathcal{G}$ be a closed graph pattern and $\mathcal{G}'$ be its corresponding representative pruned graph pattern. As $\mathcal{G}' \subseteq \mathcal{G}, \mathcal{F}(\mathcal{G}, \mathcal{D}) \subseteq \mathcal{F}(\mathcal{G}', \mathcal{D})$. On the opposite, as any element of $\mathcal{G}$ is a subgraph of an element of $\mathcal{G}'$, we have: $\mathcal{F}(\mathcal{G}', \mathcal{D}) \subseteq \mathcal{F}(\mathcal{G}, \mathcal{D})$.

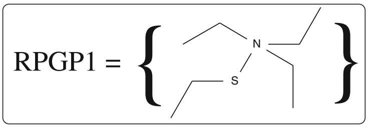


Fig. 6 Example of a Representative Pruned Graph Pattern

As two distinct representative pruned graph patterns cannot generate the same closed graph pattern, we can state that there is a *one to one mapping between the set of the closed graph patterns and the set of the representative pruned graph patterns*.

Figure 6 displays the representative pruned graph pattern that represents the closed graph pattern CGP1 shown on Fig. 5.

Consequently, we can state the following result which is necessary to obtain the exact values of the frequency and the emergence of each graph pattern.

Proposition 6 (Pruning a closed graph pattern maintains its support) *Let $\mathcal{G}$ be a closed graph pattern. Let $\mathcal{G}'$ be a graph pattern. If $\mathcal{G}'$ is the representative pruned graph pattern corresponding to $\mathcal{G}$ then $\mathcal{F}(\mathcal{G}', \mathcal{D}) = \mathcal{F}(\mathcal{G}, \mathcal{D})$.*

It means that the whole set of the frequent emerging graph patterns is condensed by its set of representative pruned graph patterns. Moreover, the exact values of the frequency and the emergence of each graph pattern can be inferred from the condensed representation since a representative pruned graph pattern has the same support as its closed graph pattern (Proposition 6). Note that the condensed representation based on the representative pruned graph patterns and the condensed representation based on the closed patterns have the same size (i.e., the same number of graph patterns). But, the great interest of the representative pruned graph patterns is to provide shorter patterns, thus more understandable patterns.

4 Experiments on chemical data

4.1 Motivations, materials and methods

The aims of the experiments is to study: (i) the feasibility of our approach by providing quantitative results on the computation method, (ii) the condensed representation of the Frequent Emerging Graph Patterns (FEGPs) by their Representative Pruned Graph Patterns (RPGPs), (iii) the relationships between the FEGPs and the discriminative fragments and (iv) the interest of the FEGPs of length greater than 1.

The dataset gathers molecules stored in *EPA Fathead Minnow Acute Toxicity Database* (EPAFHM 2008). It has been generated by the Environment Protection Agency of the United-States and it has been used to elaborate expert systems predicting the toxicity of chemicals (Veith et al. 1988). From EPAFHM, we have selected the molecules classified as toxic and non-toxic, toxicity being established according to the measure of *LC50*. The resulting set $\mathcal{D}$ contains 395 molecules (Table 2) and it is partitioned into two subsets: $\mathcal{D}_1$ contains toxic molecules (223 molecules) and $\mathcal{D}_2$ contains non-toxic molecules (172 molecules). Experiments were conducted on a computer running *Linux* operating system with a dual processor at 2.83 GHz and a RAM of 3.8 GB.

Table 2 Excerpt from the EPAFHM database: 395 molecules partitioned into two subsets according to the measure of *LC50*

Set	Toxicity	*LC50* measure	Number of molecules	Size Min	Size Max	Size Average
$\mathcal{D}_1$	Toxic	*LC50* ≤ 10 mg/l	223	3	34	12.8
$\mathcal{D}_2$	Non-toxic	100 mg/l ≤ *LC50*	172	2	19	8.2
$\mathcal{D}$	–	–	395	2	34	10.8

Methods used throughout the experimental part When we have to assess the FEGPs in a context of classification we adopt a *cross-validation scheme* (Section 4.3). In a n-folds cross-validation scheme, both $\mathcal{D}_1$ and $\mathcal{D}_2$ are split into n samples of equal size. Each of these n samples constitutes successively the *testing set*, the gathering of the other ones constituting the *learning set*. The resulting classification model leaves a large testing set on which we can measure the model's average performance, and it is almost as powerful as the full model which is trained on 100% of the dataset (Hassan et al. 1996). As our dataset is constituted of 395 graphs and as we need one hundred graphs in each testing set, we have chosen a *four-folds* cross-validation scheme.

The Frequent Connected SubGraphs (FCSGs) are extracted according to a minimum frequency threshold in $\mathcal{D}_1$. Except for the Section 4.2.1, this threshold is set thanks to a Chi-Square Test of Independance; it corresponds to the minimum frequency an attribute has to exceed to be able to be considered as dependant of the classification (Schervish 1995). This threshold is always determined at a level of confidence of 99% for the statistical test, assuming the attribute never occurs in $\mathcal{D}_2$. It is named the *Chi-square frequency threshold*. For example, considering the entire dataset $\mathcal{D}$ partitioned into $\mathcal{D}_1$ (223 graphs) and $\mathcal{D}_2$ (172 graphs), the correponsding Chi-square frequency threshold is 4.5%.

4.2 Extraction and representation of the emerging graph patterns

This section focuses on both the feasibility of the method and the study of the condensed representation of the FEGPs based on the RPGPs.

4.2.1 Extraction of the frequent connected subgraphs

The first experiment studies the first step of our mining method, the extraction of the FCSGs. It focuses on the number of the FCSGs and on their average size. The related runtimes also are of a particular interest.

The minimum frequency threshold $f_{\mathcal{D}_1}$ varies from 1 to 10% with a step of 1%. For each value of $f_{\mathcal{D}_1}$, we measure (i) the number of FCSGs, (ii) the average size (number of vertices) of the FCSGs, (iii) the runtime of the extraction and (iv) the number of FCSGs extracted per second. Results are displayed in Table 3.

As explained in Section 3.2, we have modified the implementation of Gaston in order to obtain not only the FCSGs but also their supports for the whole dataset. As the difference between the original version of Gastion and our modified version never exceeds 0.004 second, we only provide the runtimes related to the modified version of Gaston without loss of generality. Measured on the modified version of Gaston, the runtime dedicated to extract the FCSGs depends on the minimum frequency threshold; it varies from 2.74 s ($f_{\mathcal{D}_1} = 1\%$) to 0.02 second ($f_{\mathcal{D}_1} = 10\%$).

Table 3 Measures related to the extraction of the Frequent Connected SubGraphs according to the minimum frequency threshold

Frequency threshold (%)	Number of FCSGs	Average size	Computing runtime (s)	Number of FCSGs/sec
1	49,428	13.9	2.74	18,039
2	3,487	9.22	0.17	20,511
3	958	7.11	0.09	10,644
4	561	6.90	0.06	9,350
5	414	6.91	0.05	8,280
6	288	6.05	0.04	7,200
7	195	5.82	0.04	4,875
8	155	5.52	0.03	5,166
9	120	5.28	0.03	4,000
10	109	5.20	0.02	5,450

The number of extracted FCSGs is strongly correlated to the minimum frequency threshold; it varies from 49 428 FCSGs ($f_{\mathcal{D}_1} = 1\%$) to 109 FCSGs ($f_{\mathcal{D}_1} = 10\%$). The average size of the extracted FCSGs also depends on the minimum frequency threshold; it varies from 13.9 vertices ($f_{\mathcal{D}_1} = 1\%$) to 5.2 vertices ($f_{\mathcal{D}_1} = 10\%$). The number of extracted FCSGs per second relies on the minimum frequency threshold; it varies from 18 039 FCSGs per second ($f_{\mathcal{D}_1} = 1\%$) to 5 450 FCSGs per second ($f_{\mathcal{D}_1} = 10\%$).

Even if this experiment has been conducted on a medium sized dataset, the results point out the feasibility of the extraction of the FCSGs (the first step of our method) in this context. Lowering the minimum frequency threshold increases the number of extracted FCSGs and their average size.

4.2.2 The emerging graph patterns: their mining step and their condensed representation based on the representative pruned graph patterns

This section details the mining of the FEGPs and it assesses the condensed representation of the FEGPs based on the RPGPs. We assume that the FCSGs have already been extracted with a Chi-square frequency threshold calculated on the whole dataset (4.5%). The set of the FCSGs contains 477 subgraphs with an average size of 6.87 vertices.

The first experiment evaluates the efficiency of the condensed representation of the FEGPs based on the Closed Graph Patterns (CGPs). The minimum growth rate threshold varies from 1 to 10 with a step of 1. For each threshold we measure (i) the number of FEGPs and CGPs, (ii) the average lengths of the FEGPs and the CGPs, (iii) the runtime for the extraction of the FEGPs and (iv) the number of FEGPs represented by one CGP. Results are displayed in Table 4. We also give the values of these measures when the minimum growth rate is equal to ∞; this rate corresponds to the patterns that are not present in $\mathcal{D}_2$ (see Section 2.2).

The number of extracted FEGPs varies from $5.21.10^6$ ($\rho = 1$) to $1.15.10^6$ ($\rho = \infty$). The number of extracted CGPs varies from 677 ($\rho = 1$) to 87 ($\rho = \infty$). Consequently the condensed representation of the FEGPs based on the CGPs reduces significantly the number of patterns without loosing any information. The number of FEGPs embedded into one CGPs increases as the growth rate threshold increases: it varies from 7,703 ($\rho = 1$) to 13,273 ($\rho = \infty$). The condensed representation of the FEGPs based on the CGPs seems to be more efficient when the growth rate threshold is high.

Table 4 Extraction and summarization of the Frequent Emerging Graph Patterns according to the minimum growth rate threshold

Growth rate threshold	Number of FEGPs (.10^6)	Number of CGPs	Extraction runtime (s)	Number of FEGPs in one CGP
1	5.21	677	352	7,703
2	5.02	548	335	9,168
3	4.16	438	255	9,505
4	3.52	345	164	10,216
5	3.15	286	129	11,022
6	2.94	255	96	11,555
7	2.52	212	75	11,903
8	2.03	172	52	11,830
9	1.83	154	23	11,895
10	1.77	142	41	12,527
∞	1.15	87	22	13,273

The runtime for extracting the FEGPs decreases when the growth rate threshold increases: it varies from 352 s for $\rho = 1$ to 22 s for $\rho = \infty$. As the extraction of the FCSGs never exceeds 3 s, the runtime for the whole process (extraction of FCSGs and extraction of FEGPs) is close to the runtime of the extraction of the FEGPs. We may conclude that the whole method is applicable on medium-sized dataset.

The second experiment focuses on the condensed representation of the CGPs based on the RPGPs. The minimum growth rate threshold varies from 1 to 10 with a step of 1. For each threshold we measure (i) the average length of the FEGPs and the RPGPs. Results are displayed in Table 5. We also give the values of the measures when the minimum growth rate is equal to ∞.

The average length of the CGPs varies from 15.1 ($\rho = 1$) to 18.1 ($\rho = \infty$). The average length of the RPGPs from 2.23 ($\rho = 1$) to 2.55 ($\rho = \infty$). In average the RPGPs are by far smaller than the CGPs. The condensed representation of the FEGPs based on the RPGPs reduces significantly the length of the FEGPs without loosing any information.

Even if the experiments were conducted on a medium-sized dataset, the extraction of the EGPs from a chemical dataset is feasible. When we have to memorize the

Table 5 Measures on the condensed representation of the Frequent Emerging Graph Patterns based on the Representative Pruned Graph Patterns according to the minimum growth rate threshold

Growth rate threshold	Average length of the CGPs	Average length of the RPGPs	Average Number of FCSGs deleted
1	15.1	2.23	12.8
2	16.1	2.20	13.8
3	16.8	2.37	14.4
4	16.4	2.39	14
5	17.1	2.44	14.6
6	17.3	2.43	14.8
7	16.6	2.42	14.2
8	17	2.45	14.4
9	17.1	2.44	14.6
10	16.6	2.43	14.2
∞	18.1	2.55	15.6

EGPs, their condensed representation by their RPGPs appears to be very efficient: it drastically reduces both the number of patterns and their average length.

4.3 Evaluation of the frequent emerging graph patterns in a classification context

Quantitative results have shown the feasibility and the efficiency of the method. We now evaluate the interest of the FEGPs in a context of classification.

4.3.1 Relationship between the discriminative fragments and the frequent emerging graph patterns

First we recall the notion of discriminative fragment. Given two thresholds of frequency f_M and f_m, a pattern of length 1 is a *discriminative fragment* if its frequency in $\mathcal{D}_1$ exceeds f_M and its frequency in $\mathcal{D}_2$ is above f_m (Borgelt and Berthold 2002; Borgelt et al. 2005). As the notion of FEGP relies on a minimum frequency threshold in $\mathcal{D}_1$, $f_{\mathcal{D}_1}$, and on a minimum growth rate threshold from $\mathcal{D}_2$ to $\mathcal{D}_1$, ρ, we naturally relate the notion of discriminative fragments with the notion of the FEGPs by terming: *a FEGP is a discriminative pattern if its frequency threshold in $\mathcal{D}_2$ is above $\frac{f_{\mathcal{D}_1}}{\rho}$*. Under this definition, a discriminative pattern is always a FEGP (*the discriminative fragments constitute a subset of the FEGPs*). As a RPGP has exactly the same support than any FEGP it represents, the following property is immediate: a RPGP is a discriminative pattern if any of its represented FEGP is a discriminative pattern. Consequently we are able to compare the notion of FEGPs and the notion of discriminative patterns by comparing the RPGPs that are discriminative with the RPGPs that are not discriminative.

During the following experiment, the frequency threshold $f_{\mathcal{D}_1}$ is the corresponding Chi-square frequency threshold ($f_{\mathcal{D}_1} = 5.9\%$). The *coverage rate* of a set of patterns $\mathcal{P}$ into a set of graphs $\mathcal{D}$ corresponds to the proportion of the elements of $\mathcal{D}$ that contain at least one pattern of $\mathcal{P}$.

For each fold of the cross-validation scheme, the minimum growth rate threshold varies from 1 to 10 with a step of 1. The RPGPs are extracted from the learning set and they are partitioned into two subsets: the discriminative ones (*D-RPGPs*) and the non discriminative ones (*ND-RPGPs*). For the three resulting subsets we measure (i) the number of patterns (*Nb*), (ii) their coverage rate (*CR*) into the positives graphs of the testing set (*PT*) and (iii) their coverage rate into the negatives graphs of the testing set (*NT*). Results are displayed on Table 6.

Measured on the testing set, the coverage rate of the RPGPs varies from 95 to 26.5% for the positive graphs and it varies from 83.5% ($\rho = 1$) to 4% ($\rho = \infty$) for the negatives graphs. The coverage rate of the D-RPGPs does not vary when the growth rate threshold increases : from 26.5% ($\rho = 1$) to 26.5% ($\rho = \infty$) for the positive and from 4.2% ($\rho = 1$) to 4% ($\rho = \infty$) for the negatives graphs. These results show that there exist non discriminative RPGPs that are of interest within a classification context.

4.3.2 The interest of the emerging graph patterns of length greater than 1

This experiment shows that there exists FEGPs of length greater than 1 that are of interest in a context of classification. When a RPGP has a length greater than 1, its corresponding CGP has also a length greater than 1 and, consequently, they both

Table 6 Average number and coverage rates of the Representative Pruned Graph Patterns into the testing sets according to the minimum growth rate threshold

Growth	D-RPGPs			ND-RPGPs			RPGPs		
rate	Nb	CR (%)		Nb	CR (%)		Nb	CR (%)	
threshold		PT	NT		PT	NT		PT	NT
1	10.7	26.5	4.2	417.7	95	83.5	428.5	95.5	83.5
2	10.2	26.5	2.5	333.7	89.7	53	344	89.7	53
3	10	24.2	3.7	25.6	82.5	38.5	246	83	39
4	10	24	2	181	76.2	29.7	191	77.5	32.7
5	11.5	28.2	3	149.7	75.5	21	161.2	76.7	21.7
6	10.5	26	2	125.5	63.7	17.2	136	65	17.2
7	11.2	27.7	2	100.7	63.7	17.2	112	66.7	17.5
8	10.7	27.2	1.5	88.5	59.5	17.2	115	61.2	17.2
9	12.7	24.2	2.5	78.7	56.5	14	91.5	58	14
10	12.2	28.2	4.5	54.7	53.7	8	67	55	9.2
∞	12.5	26.5	4	0	0	0	12.5	26.5	4

represent at least one FEGP of length greater than 1. If we exhibit RPGPs of length greater than 1 that are of interest in a classification context, we will be allowed to say that there exist interesting FEGPs of length greater than 1.

A jumping emerging pattern is a special type of EP: a pattern is a jumping pattern when it never appears in the negative class. A jumping pattern corresponds to a highly discriminative set of characteristics (Li et al. 2001). We now focus on the RPGPs that are jumping patterns; these patterns are extracted with a growth rate threshold equal to ∞.

During the following experiment, the frequency threshold $f_{\mathcal{D}_1}$ is the corresponding Chi-square frequency threshold (5.9%). For each fold of the cross-validation scheme, the *jumping RPGPs* are extracted from the learning set and they are partitioned into two subsets: the patterns of length 1 and the patterns of length greater than 1. For these two resulting subsets, we measure (i) the number of extracted patterns, (ii) the number of patterns that are still jumping patterns in the testing set and (iii) the number of jumping RPGPs that are still jumping patterns in the testing set and that composed only with FCSGs that are not emerging fragment.

By averaging the results obtained with the different folds of the cross-validation scheme, there is 7 jumping RPGPs of length equal to 1 and 38 of length greater than 1.65% of the jumping RPGPs of length greater than 1 are still jumping patterns in the testing set: these patterns (of length greater than 1) appears to be of interest in a context of classification.

If we focus on the jumping RPGPs of length greater than 1 that still jump in the testing set, 73% of theses jumping RPGPs are constituted only with FCSGs that are not emerging alone. We can say that there exists FEGPs of length greater than one that are of interest in a context of classification. Moreover, these patterns are not always made of emerging graphs. These facts justify the extraction of FEGPs of length greater than one. As a FEGP may have an interest in a context of classification whatever its length and its constitution are, the whole set of RPGPs has to be extracted.

These experiments have shown that mining emerging graph patterns from real-world chemical dataset is feasible. Moreover, these experiments indicate that the condensed representation of the FEGPs based on the RPGPs is very efficient : it drastically reduces both the number of patterns to memorize and their lengths. Furthermore these patterns may have an interest in a context of classification whatever their length is.

5 Conclusion and future work

In this paper, we have investigated the notion of emerging graphs and we have proposed a method to mine emerging graph patterns. A strength of our approach is to extract *all* frequent emerging graph patterns (given thresholds of frequency and emerging) and not only particular emerging patterns. In the particular case of patterns of length 1, all frequent connected emerging graphs are produced. Our results are achieved thanks to a change of the description of the initial problem so that we are able to design a process combining efficient algorithmic and data mining methods. Moreover, we show that the closed graph patterns are a condensed representation of the frequent emerging graph patterns and we propose a new condensed representation based on the representative pruned graph patterns: by providing shorter patterns, it is especially dedicated to represent a set of frequent emerging graph patterns. Experiments on a real-world database composed of chemicals have shown the feasibility and the efficiency of our approach. Further work is to better investigate the use of such patterns in chemoinformatics, especially for discovering toxicophores. A lot of data can be modeled by graphs and, obviously, emerging graph patterns may be used for instance in text mining or gene regulation networks.

Acknowledgements The authors would like to thank Arnaud Soulet for very fruitful discussions and the Music-dfs prototype and the CERMN lab for its invaluable help about the data and the chemical knowledge. This work is partly supported by the ANR (French Research National Agency) funded Innotox, Bingo2 projects and the Region Basse-Normandie (Innotox2 project).

References

Borgelt, C., & Berthold, M. R. (2002). Mining molecular fragments: Finding relevant substructures of molecules. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'02)* (pp. 51–58).

Borgelt, C., Meinl, T., & Berthold, M. (2005). Moss: a program for molecular substructure mining. In *Workshop Open Source Data Mining Software* (pp. 6–15). ACM Press.

Calders, T., Rigotti, C., & Boulicaut, J.-F. (2005). A survey on condensed representations for frequent sets. In J.-F. Boulicaut, L. De Raedt, & H. Mannila (Eds.), *Constraint-based mining and inductive databases*. *Lecture notes in computer science* (Vol. 3848, pp. 64–80). Springer.

Cook, D. J., & Holder, L. B. (2006). *Mining graph data*. Wiley.

Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (1999). Performance evaluation of the vf graph matching algorithm. In *ICIAP '99: Proceedings of the 10th international conference on image analysis and processing* (p. 1172). Washington, DC, USA: IEEE Computer Society.

De Raedt, L., & Kramer, S. (2001). The levelwise version space algorithm and its application to molecular fragment finding. In *IJCAI'01* (pp. 853–862).

Dong, G., & Li, J. (1999). Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the fifth international conference on knowledge discovery and data mining (ACM SIGKDD'99)* (pp. 43–52). San Diego, CA: ACM Press.

EPAFHM (2008). Mid continent ecology division (environement protection agency), fathead minnow. http://www.epa.gov/med/Prods_Pubs/fathead_minnow.htm.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability*. Freeman and Company.

Hassan, M., Bielawski, J., Hempel, J., & Waldman, M. (1996). Optimization and visualization of molecular diversity of combinatorial libraries. *Molecular Diversity, 2*(1), 64–74.

Kramer, S., Raedt, L. D., & Helma, C. (2001). Molecular feature mining in HIV data. In *KDD* (pp. 136–143).

Li, J., Dong, G., & Ramamohanarao, K. (2001). Making use of the most expressive jumping emerging patterns for classification. *Knowledge and Information Systems, 3*(2), 131–145.

Li, J., & Wong, L. (2001). Emerging patterns and gene expression data. *Genome Informatics, 12*, 3–13.

Lozano, S., Poezevara, G., Halm-Lemeille, M.-P., Lescot-Fontaine, E., Lepailleur, A., Bissell-Siders, R., et al. (2010). Introduction of jumping fragments in combination with qsars for the assessment of classification in ecotoxicology. *Journal of Chemical Information and Modeling, 50*(8), 1330–1339.

Mannila, H., & Toivonen, H. (1997). Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery, 1*(3), 241–258.

Morishita, S., Sese, J., & Ward, B. (2000). Traversing itemset lattices with statistical metric pruning. In *In Proc. of the 19th ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems* (pp. 226–236). ACM.

Ng, R. T., Lakshmanan, V. S., Han,J.,& Pang, A. (1998). Exploratory mining and pruning optimizations of constrained associations rules. In *proceedings of ACM SIGMOD'98* (pp. 13–24). ACM Press.

Nijssen, S., & Kok, J. N. (2004). A quickstart in frequent structure mining can make a difference. In W. Kim, R. Kohavi, J. Gehrke, & W. DuMouchel, (Eds.), *KDD* (pp. 647–652). ACM.

Plantevit, M. & Crémilleux, B. (2009). Condensed representation of sequential patterns according to frequency-based measures. In *8th international symposium on Intelligent Data Analysis (IDA'09), Lecture Notes in Computer Science* (Vol. 5772, pp. 155–166). Lyon, France: Springer.

Poezevara, G., Cuissart, B., & Crémilleux, B. (2009). Discovering emerging graph patterns from chemicals. In *18th International Symposium on Methodologies for Intelligent Systems (ISMIS'09). Lecture Notes in Artificial Intelligence* (Vol. 5522, pp. 45–55). Prague, Czech Republic: Springer.

Schervish, M. J. (1995). *Theory of statisitics* (chapter 7). *Large sample theory* (p. 467). Springer series in statisitics. Springer.

Soulet, A., & Crémilleux, B. (2009). Mining constraint-based patterns using automatic relaxation. *Intelligent Data Analysis, 13*(1), 1–25.

Soulet, A., Crémilleux, B., & Rioult, F. (2005). *Knowledge discovery in inductive databases: KDID 2004. Lecture notes in computer science, chapter Condensed representation of EPs and patterns quantified by frequency-based measures*, (Vol. 3377, pp. 173–190). Springer.

Soulet, A., Kléma, J., & Crémilleux, B. (2007). *Post-proceedings of the 5th international workshop on Knowledge Discovery in Inductive Databases in conjunction with ECML/PKDD 2006 (KDID'06). Lecture notes in computer science, chapter efficient mining under rich constraints derived from various datasets* (Vol. 4747, pp. 223–239). Springer.

Ting, R. M. H., & Bailey, J. (2006). Mining minimal contrast subgraph patterns. In J. Ghosh, D. Lambert, D. B. Skillicorn, & J. Srivastava, (Eds.), *SDM*, (pp. 638–642). SIAM.

Veith, G., Greenwood, B., Hunter, R., Niemi, G., & Regal, R. (1988). On the intrinsic dimensionality of chemical structure space. *Chemosphere, 17*(8), 1617–1644

Wörlein, M., Meinl, T., Fischer, I., & Philippsen, M. (2005). A quantitative comparison of the subgraph miners mofa, gspan, FFSM, and gaston. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, & J. Gama (Eds.), *Knowledge Discovery in Databases: PKDD 2005. Lecture notes in computer science* (Vol. 3721, pp. 392–403). Springer.

Yan, X., & Han, J. (2003). Closegraph: Mining closed frequent graph patterns. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'03)* (pp. 286–295). New York, NY, USA: ACM.