Discovering Emerging Graph Patterns from Chemicals

Guillaume Poezevara, Bertrand Cuissart, Bruno Crémilleux

Discovering Emerging Graph Patterns from Chemicals

Guillaume Poezevara, Bertrand Cuissart, and Bruno Crémilleux

Laboratoire GREYC-CNRS UMR 6072
Université de Caen Basse-Normandie, France
Forename.Lastname@info.unicaen.fr
http://www.greyc.unicaen.fr/

Abstract. Emerging patterns are patterns of a great interest for characterizing classes. This task 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. Experiments on a real-world database composed of chemicals show the feasibility and the efficiency of our approach.

Keywords: Data mining, emerging patterns, 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 is to establish relationships between chemicals (or molecules) and a given activity (e.g., toxicity). Such a relationship may be characterized by *patterns* associating atoms and chemical bonds. A difficulty of the 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 [12] provides efficient methods. A very useful constraint is the emerging constraint [5]: emerging patterns (EPs) are patterns whose frequency strongly varies between two classes (the frequency of a pattern P is the number of examples in the database supporting P). EPs enable us to characterize classes (e.g., toxic versus non-toxic chemicals) in a quantitative and qualitative way. EPs are at the origin of various works such as powerful classifiers [9]. From an applicative point of view, we can quote various works on the characterization of biochemical properties or medical data [10].

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 [11] cannot be applied. 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 extract a subset of the whole set of EPs. That it is why some techniques use handlings of borders but it is very expensive [5]. 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 [13,14]. 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 graphs [2,4]. These last two works are based on a combination of monotone and anti-monotone constraints and do not extract the whole collection of EPs. Mining patterns in a graph dataset is a much more challenging task than mining patterns in itemsets.

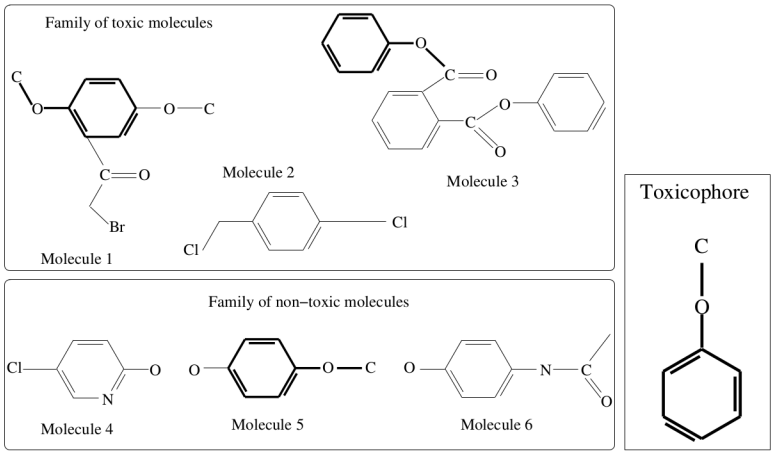
In this paper, we tackle this challenge of mining emerging graph patterns. Our main contribution is to propose a method mining all frequent emerging graph patterns. This result is achieved by a change of the description of the initial problem in order to be able to use efficient algorithmic and data mining methods (see Section 3). In particular, all frequent connected emerging subgraphs are produced; they correspond to the patterns of cardinality 1. These subgraphs are useful because they are the most understandable subgraphs from the chemical point of view. The patterns of cardinality greater than one capture the emerging power of associations of connected subgraphs. 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. Finally, we present a case study on a chemical database provided by the Environnement Protection Agency. This experiment shows the feasibility of our approach and suggests promising chemical investigations on the discovery of toxicophores.

This paper is organized as follows. Section 2 outlines preliminary definitions and related work. Our method for mining all frequent emerging graph patterns is 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



(a) Molecules excerpted from the EPAFHM database [6] (b) Graph G_0

Fig. 1. Examples of molecules

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. Fig. 1(a) displays *molecular graphs*. The graph G_0 (see Fig. 1(b)) is (isomorphic to) a subgraph of molecules 1, 3 and 5 in Fig. 1(a) and therefore its frequency is 3. Assuming now that $\mathcal{D}$ is partitioned into two subsets (or classes) $\mathcal{D}_1$ and $\mathcal{D}_2$. For instance, in Fig. 1(a), its top part $\mathcal{D}_1$ gathers toxic molecules and its bottom part $\mathcal{D}_2$ non-toxic molecules. With a frequency threshold of 2 both for $\mathcal{D}_1$ and $\mathcal{D}_2$, G_0 is a frequent graph in $\mathcal{D}_1$ but it is infrequent in $\mathcal{D}_2$.

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 isomorphism. Given a couple of graphs (G', G), the problem of deciding if G' is isomorphic to a subgraph of G is named the *Subgraph Isomorphism Problem (SI)*. *SI* is NP-complete [7, 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, we will need to compute the frequency of a set of graphs $\mathcal{G}$ (i.e. a *graph pattern*). $\mathcal{F}(\mathcal{G}, \mathcal{D})$ denotes the graphs of $\mathcal{D}$ that include every graph of $\mathcal{G}$ as a subgraph ($\mathcal{F}(\mathcal{G}, \mathcal{D}) = \{G_{\mathcal{D}} \in \mathcal{D} : \forall G \in \mathcal{G},\ G$ is a subgraph of $G_{\mathcal{D}}\}$).

For example, the graph pattern made of G_0 and the graph c=o named G_1 has a frequency of 2 in $\mathcal{D}$ (it is a subgraph of molecules 1 and 3). 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) 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 partioned 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 can now provide the terms of the problem of mining the whole set of frequent EGPs:

Definition 2 (Frequent Emerging Graph Pattern Extraction (*FEGPE*))

Input: *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*

Output: *the set of the frequent emerging graph patterns with their growth rate from $\mathcal{D}_2$ to $\mathcal{D}_1$ according to $f_{\mathcal{D}_1}$ and ρ.*

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 $\mathcal{D}_1$.

For the sake of simplicity, the definitions are given with only with 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 [5]). Following our example, with $f_{\mathcal{D}_1} = 2$ and $\rho = 2$, the *FEGPE* problem provides 273 intervals [14] (see Section 3) condensing the frequent emerging graph patterns including G_0 ($GR_1(G_0) = 2$) and the graph pattern $\mathcal{G}$ made of G_0 and G_1 ($GR_1(\mathcal{G}) = \infty$).

2.2 Related Work: Extraction of Discriminative Subgraphs

Several methods have been designed for discovering subgraphs that are correlated to a given class.

Molfea [8] uses a levelwise algorithm [11] 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 [1,2] 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. Indeed, mining all discriminative fragments according to the thresholds f_M and f_m do not ensure extracting all EPs having a growth rate higher than f_M/f_m or another given growth rate threshold. At 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* [15]. 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 presents our method for mining frequent emerging graph patterns. We start by giving the key ideas and the three steps of our method.

Outline. Let $\mathcal{D}$ be a set of graphs partioned into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$. Our main idea is to change the description of the initial problem in order to be able to use efficient algorithmic and data mining methods. Briefly speaking, for mining the whole set of the frequent EGPs from $\mathcal{D}_2$ to $\mathcal{D}_1$, we start by only extracting the frequent connected subgraphs in $\mathcal{D}_1$. Then, by using a subgraph isomorphism method, both molecules of $\mathcal{D}_1$ and $\mathcal{D}_2$ are described with the frequent connected subgraphs as new features. This change of description of the problem brings a twofold advantage. First, as EGPs can only stem from these new features, it is enough in $\mathcal{D}_2$ to focus on candidate patterns made of these features. It strongly reduces the number of candidate patterns and this is precious especially in $\mathcal{D}_2$ because we have in this dataset to deal with graphs with very low frequency. Second, it enables us to set the problem in an itemset context from which we can reuse efficient results on the emerging constraint. Finally, we solve the *FEGPE* problem described in Section 2.1.

Main steps of our method. Fig. 2 depicts the three main steps of our method:

1) extracting the frequent connected subgraphs in $\mathcal{D}_1$ according to the frequency threshold $f_{\mathcal{D}_1}$. This is the *FCSG* problem.
2) for each graph $G_{\mathcal{D}}$ of $\mathcal{D}$ and for each connected graph G resulting from 1), we successively test if G is a subgraph of $G_{\mathcal{D}}$. For that purpose, we have to solve multiple *SI* problems. For that task, we use our own implementation of J.R. Ullmann's algorithm [16]. Then the dataset can be recoded such that

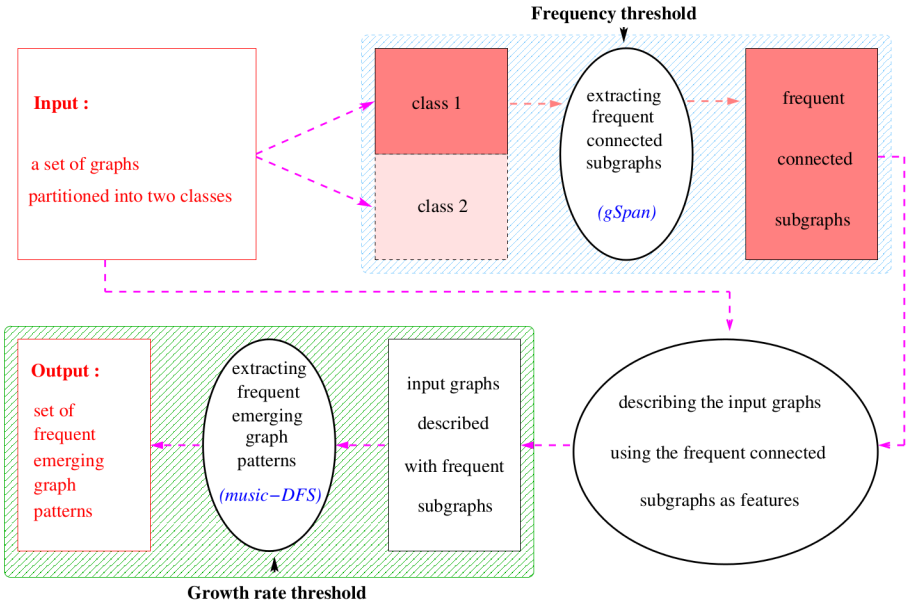


Fig. 2. The three steps of the process: extracting FCSGs (*up*), describing the input graphs using the FCSGs as features (*down-right*) and extracting frequent EGPs (*down-left*)

each row is a graph G of $\mathcal{D}$ and each column indicates if a frequent connected graph issued from 1) is present or not in G.

3) 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.

There are several methods to solve the *FCSG* problem. 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 [3]: 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 [18]. For mining a set of chemical graphs, gSpan [19] consumes less memory and runs faster than the other ones. For these reasons, we have chosen gSpan for extracting frequent connected subgraphs. Moreover, gSpan is available on `http://illimine.cs.uiuc.edu/` under GNU GPL License Version 2[1].

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 [14]. Then MUSIC-DFS mines soundly and completely all the patterns satisfying a given set of input

[1] `http://www.gnu.org/licenses/gpl/html`
[2] `http://www.info.univ-tours.fr/~soulet/music-dfs/music-dfs.html`

Table 1. Excerpt from the EPAFHM database: 395 molecules partitioned into two subsets according to the measure of *LC50*

Class	Subset	Toxicity	*LC50* measure	Number of molecules
1	toxic	$\mathcal{D}_1$	$LC50 \leq 10mg/l$	223
2	non-toxic	$\mathcal{D}_2$	$100mg/l \leq LC50$	172

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 pruning conditions are based on intervals. Here, an *interval* denominates a set of patterns that include a same prefix-free pattern P and that are included in the prefix-closure of P (see [14] 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 [14]. 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.

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.

4 Experiments on Chemical Data

Experiments are presented according to the three steps of our method. They show the feasibility of our approach and provide quantitative results.

The dataset gathers molecules stored in *EPA Fathead Minnow Acute Toxicity Database* [6] (EPAFHM). It has been generated by the Environment Protection Agency (EPA) of the United-States, it has been used to elaborate expert systems predicting the toxicity of chemicals [17]. 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 1) 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 1.9 Gio.

Extraction of the FCSGs. As already said, we use gSpan to extract the set of FCSGs. The input set of graphs is the set $\mathcal{D}_1$. The frequency threshold $f_{\mathcal{D}_1}$ varies from 1 % to 10 % with a step of 1 %. For each calculation, we measure the number of frequent connected subgraphs extracted and the computing time. Results are displayed in Fig 3.

First, as expected, the number of extracted subgraphs decreases exponentially as the frequency increases. It takes 8 seconds to extract 49438 subgraphs ($f_{\mathcal{D}_1} =$

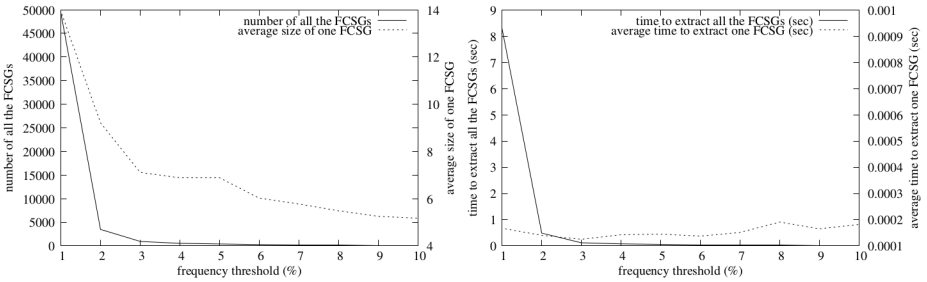


Fig. 3. Extraction of FCSGs according to the frequency threshold

Table 2. Measures on SI according to frequency threshold

frequency threshold	1	2	3	4	5	6	7	8	9	10
number of descriptors	49438	3492	956	558	416	291	198	157	121	110
avg time to describe a graph (sec)	139	7.49	2.04	1.18	0.870	0.612	0.420	0.329	0.255	0.250
avg time per isomorphism (sec)	$2.27\ 10^{-3}$	$2.11\ 10^{-3}$	$2.09\ 10^{-3}$	$2.12\ 10^{-3}$	$2.10\ 10^{-3}$	$2.09\ 10^{-3}$	$2.12\ 10^{-3}$	$2.13\ 10^{-3}$	$2.14\ 10^{-3}$	$2.82\ 10^{-3}$

1%) and 1 second to extract 110 subgraphs ($f_{\mathcal{D}_1} = 10\%$). Second, the computing time is strongly related to the number of extracted subgraphs: the average time to extract one FCSG varies from $1\ 10^{-4}$ second to $2\ 10^{-4}$ second. Third, the average size of an extracted subgraph decreases as frequency increases: from a average size of 14 vertices ($f_{\mathcal{D}_1} = 1\%$) to an average size of 5 vertices ($f_{\mathcal{D}_1} = 10\%$).

Description of the Graphs Using Subgraphs as Features. As in the previous experiment, $f_{\mathcal{D}_1}$ varies from 1% to 10% with a step of 1%. For each frequency threshold, the set of descriptors is the set of FCSGs extracted from $\mathcal{D}_1$. Each graph of $\mathcal{D}$ is then recoded according to these descriptors, one SI is required for recoding each descriptor. We count the number of successful SIs and we measure the computing time. Results are displayed on Table 2.

Obviously, the number of descriptors decreases as the frequency threshold increases (see the previous experiment). The average computing time to describe a graph decreases as the the number of descriptors decreases: it takes 140 seconds with 49438 descriptors ($f_{\mathcal{D}_1} = 1\%$) and 0.250 second with 110 descriptors ($f_{\mathcal{D}_1} = 10\%$). As the size of each descriptor remains small, the average time per isomorphism is stable (around $2\ 10^{-3}$ second). Consequently, the average computing time to describe a graph is strongly correlated to the number of descriptors.

Extraction of Frequent EGPs. The input dataset is the set $\mathcal{D}$ recoded with the FCSGs extracted from $\mathcal{D}_1$ under a frequency threshold of 5% (416 descriptors). The growth rate threshold ρ varies from 2 to 20 with a step of 1. For each

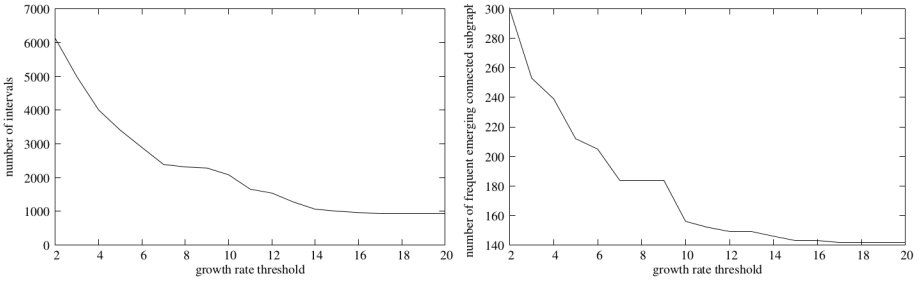


Fig. 4. Extraction of the frequent EGPs according to the growth rate threshold

value of ρ, we indicate the number of intervals mined by MUSIC-DFS. We also indicate the number of frequent emerging connected subgraphs as these graphs are of particular interest in chemoinformatics. Recall that the set of frequent emerging connected subgraphs corresponds to the set of frequent EGPs of length 1. Computing time is not provided because it is always very fast (MUSIC-DFS always extracts all emerging graph patterns in less than one second). Results are displayed on Fig 4.

The number of intervals decreases as ρ increases: it varies from 6000 ($\rho = 2$) to 1000 ($\rho = 20$). The number of emerging graphs decreases as growth rate threshold ρ increases: it varies from 300 ($\rho = 2$) to 140 ($\rho = 20$). Interestingly, such experiments depict the speed of decreasing of the number of patterns according to the growth rate threshold. We are also able to extract the jumping emerging patterns (see Section 2.2, page 49). An EGP is a jumping emerging pattern if it is always extracted whatever the value of the growth rate is. For this experiment, there are 635 jumping emerging patterns, 69 of them are of length 1. These latters are of particular interest for the toxicologist.

An overall result of these experiments is to show that mining emerging graph patterns from real-world chemical dataset is feasible. About use of EGPs in chemoinformatics, these patterns are currently used by the toxicologists in the search for discovering toxicophores because these latter are strongly present in toxic molecules and may be responsible of their toxicity. However, toxicity does not rely on the sole presence of a toxicophore (a pattern of length one). Indeed many toxicophores could be inhibited by a neighboring fragment. Although our tool underlines these interactions, we still don't know how to explain the patterns of length greater than one.

We are now processing a bigger dataset excerpted from the *Registry of Toxic Effects of Chemical Substances* (`http://www.cchst.ca/products/rtecs/`). This set contains more than 10 000 molecular graphs, along with their toxicity measures. The EGPs resulting from this experiment will provide valuable information for toxicologists.

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 subgraphs 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. 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.

Acknowledgments. 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 project Bingo2 ANR-07-MDCO-014 and the Region Basse-Normandie (INNOTOX2 project).

References

1. Borgelt, C., Berthold, M.R.: Mining molecular fragments: Finding relevant substructures of molecules. In: Proceedings of the IEEE International Conference on Data Mining (ICDM 2002), pp. 51–58 (2002)
2. Borgelt, C., Meinl, T., Berthold, M.: Moss: a program for molecular substructure mining. In: Workshop Open Source Data Mining Software, pp. 6–15. ACM Press, New York (2005)
3. Cook, D.J., Holder, L.B.: Mining Graph Data. John Wiley & Sons, Chichester (2006)
4. De Raedt, L., Kramer, S.: The levelwise version space algorithm and its application to molecular fragment finding. In: IJCAI 2001, pp. 853–862 (2001)
5. Dong, G., Li, J.: Efficient mining of emerging patterns: discovering trends and differences. In: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 1999), pp. 43–52. ACM Press, New York (1999)
6. EPAFHM. Mid continent ecology division (environement protection agency), fathead minnow, `http://www.epa.gov/med/Prods_Pubs/fathead_minnow.htm`
7. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman and Company, New York (1979)
8. Kramer, S., Raedt, L.D., Helma, C.: Molecular feature mining in HIV data. In: KDD, pp. 136–143 (2001)
9. Li, J., Dong, G., Ramamohanarao, K.: Making use of the most expressive jumping emerging patterns for classification. Knowledge and Information Systems 3(2), 131–145 (2001)
10. Li, J., Wong, L.: Emerging patterns and gene expression data. Genome Informatics 12, 3–13 (2001)
11. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 1(3), 241–258 (1997)

12. Ng, R.T., Lakshmanan, V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. In: Proceedings of ACM SIGMOD 1998, pp. 13–24. ACM Press, New York (1998)
13. Soulet, A., Crémilleux, B.: Mining constraint-based patterns using automatic relaxation. Intelligent Data Analysis 13(1), 1–25 (2009)
14. Soulet, A., Kléma, J., Crémilleux, B.: Efficient Mining under Rich Constraints Derived from Various Datasets. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 223–239. Springer, Heidelberg (2007)
15. Ting, R.M.H., Bailey, J.: Mining minimal contrast subgraph patterns. In: Ghosh, J., Lambert, D., Skillicorn, D.B., Srivastava, J. (eds.) SDM, pp. 638–642. SIAM, Philadelphia (2006)
16. Ullman, J.: An algorithm for subgraph isomorphism. Journal of the ACM 23, 31–42 (1976)
17. Veith, G., Greenwood, B., Hunter, R., Niemi, G., Regal, R.: On the intrinsic dimensionality of chemical structure space. Chemosphere 17(8), 1617–1644 (1988)
18. Wörlein, M., Meinl, T., Fischer, I., Philippsen, M.: A quantitative comparison of the subgraph miners mofa, gspan, FFSM, and gaston. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 392–403. Springer, Heidelberg (2005)
19. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: ICDM. LNCS, vol. 2394, pp. 721–724. IEEE Computer Society Press, Los Alamitos (2002)