Wave Top-k Random-d Family Search: How to Guide an Expert in a Structured Pattern Space

Etienne Lehembre[1 [0000−0002−1374−5453]], Bruno Cremilleux[1 [1111−2222−3333−4444]], Bertrand Cuissart[1 [2222−−3333−4444−5555]], Abdelkader Ouali[1 [2222−−3333−4444−5555]], and Albrecht Zimmermann[1 [2222−−3333−4444−5555]]

GREYC, CNRS UMR 6072, UNICAEN, Normandy Univ. Caen, France
{firstname.lastname}@unicaen.fr
https://www.greyc.fr/en/home/

Abstract. This paper presents the method Wave Top-k Random-d Family Search (WTRFS) which guides an expert through data mining results according to her interest. The method exploits expert feedback, paired with the relation between patterns to spread the expert's interest. It avoids the typical feature definition step commonly used in interactive data mining which limits the flexibility of the discovery process. We empirically demonstrate that WTRFS returns the most relevant results for the user's subjective interest. Even with imperfect feedback, WTRFS behavior is robust as it stills deliver primarily most interesting results.

Keywords: Interactive Pattern Mining · Structured Patterns · Machine Learning.

1 Introduction

In the pharmaceutical industry, measuring the affinity of a drug-like molecule w.r.t. a biological receptor is both a vital operation to identify drug precursors, and an expensive one, both financially and resource-wise. As such, it is common practice to use computational methods to suggest molecular substructure patterns associated to strong affinity for a targeted receptor by mining data from the previous test series. When these results are a set of salient patterns, a recurring problem is their large number, often impossible to analyze for a human expert. Various approaches address this problem, e.g. *condensed representations* of result sets that synthesize the result space [19], numerous *quality measures* [22], and, more recently, *pattern set mining* techniques [6]. However, even the combination of these methods remains insufficient as the result space remains too large for human experts. Therefore, one proposal is to integrate the expert into the process via a search described as *interactive* [9].

While several interactive pattern mining methods deal with data as itemsets [4, 15], few works focus on interactive pattern mining dedicated to structured data, e.g. graphs [2, 3], topical to process molecular data. Moreover, even in those works, subgraphs are treated as itemsets and the relationships between

them are not fully used. The methods do not relate the size of the subgraphs to a level of specificity in the pattern space. In addition, as the standard approach in interactive mining learns an approximation of user's preferences on patterns by translating them into weights on predefined features, it necessarily narrows down the adaptation to the user's preferences. Finally, most of the current methods either use a binary interaction, a pattern being seen either as a positive one or as a negative one, which may lack nuance to describe the user interest, or require a full user-derived pattern ranking, which quickly becomes challenging.

In this work, we structure the result space to efficiently compute the sequence of subgraph samples successively submitted to an expert for feedback. The next suggestions will be revised after each expert's interaction according to her subjective interest in order to guide her exploration. Our study draws out three crucial points. First, the search has to explore a partially ordered graph (POG) without confusing the expert and has to avoid focusing only on a part of the POG, i.e. a subset of the solution patterns. This point is addressed by the introduction of *exploratory waves* to control the exploration of the POG. Second, the algorithm makes use of the partial order relation derived from graph inclusion to structure the result space and to propagate the expert's subjective interest. This avoids the usage of features and their induced bias. Third, in order to properly spread the subjective interest, the method provides a graduated interaction scheme to the user with more than two options.

Our contribution in this paper is three-fold. First, we propose an interactive pattern selection method: Wave Top-k Random-d Family Search (WTRFS), exploiting the structure of the pattern search space. Notably, we discuss our method in the context of graph mining, both due to the fact that little work exists in this field, and because of our interest in chemoinformatics. But *whenever* one can structure the pattern search space, as is also the case for itemsets, sequences, or trees, our WTRFS can be applied to guide the user through the mining results. Second, we propose more realistic and more complex ways of evaluating interactive mining, as opposed to the perfect oracles that are used in existing work. Instead of mapping a predefined quality function one-to-one to simulated responses, we propose to model experts that can get confused, or are biased or unsure of themselves. Third, we illustrate our method on a concrete use case arising in therapeutic chemistry, using a primary data set of interest: BCR-ABL.

The article is organized as follows. In Section 2 we present the related work and in Section 3 we introduce core notions. Section 4 describes the Wave Top-k Random-d Family Search method. Section 5 defines a novel methodology to evaluate interactive pattern mining processes. In Section 6, we present the experimental evaluation and discuss the results. Finally Section 7 concludes.

2 Related work

Feature construction. The usual approach to interactive pattern mining describes each result pattern in terms of features such as the presence of language elements, presence in data transactions, frequency, quality or surprisingness calculated

from patterns etc. [20, 4, 2, 8]. Those features are either hard-coded or require user definition beforehand. An exception is [3], where the authors propose to learn embedding-like representations of patterns for sequential or graph patterns. Due to the expensiveness of this operation, the representation is only learned once at the beginning, however. In contrast to this, our method does not re-encode patterns but exploits the structure of the search space to diffuse user feedback directly to result patterns, allowing local and dynamic evaluation to impact the global sampling in the result space.

Result space exploration. Interactive mining can explore the result space in two ways: post-processing a (partial) result set, or exploiting user feedback during mining. The majority of existing work [20, 4, 3, 8] does the former, essentially re-ranking or filtering patterns, simply because enforcing learned user preferences remains rather challenging. While we intend to push user feedback into the mining process itself in future work, our approach currently also performs post-processing. Works such as [2] use user feedback to perform direct randomized search space exploration, e.g. MCMC [1, 21] or MCTS sampling [5], which runs the risk, however, of only exploring a subpart of the full pattern space. The work in [7] uses user feedback to directly remove candidates from a *search beam*, which as a heuristic method again risks missing part of the search space. Our method avoids these situations by not restricting its options to the elements directly connected to the last reviewed pattern.

(Subjective) user interestingness. By involving user feedback, interactive mining arguably sounds similar to work on *subjective interestingness*. Works from that field either assume that the user has *fixed*, prior knowledge about the data [16], or that their knowledge gets updated by each mined pattern [10]. Notwithstanding the name, however, such updates typically happen in an *objective* manner, only exploiting prior patterns, and not involving the user, similar to work on non-redundant pattern mining [24, 12]. The updates are also *global* in nature. Yet the expert is sensitive to local parameters and her interest may diverge when studying two distinct regions of the result space. Our method takes this into account by only using relevant information through the concept of pattern families, syntactically related patterns. In interactive mining, the user is instead supposed to give binary (like/dislike) [7, 2] or ternary feedback [4], or rank patterns [20, 3, 8]. The relationship between this feedback and the above-mentioned features is then learned via some (often linear) function. In our work, the user selects from a number of discrete actions and their expressed interest is then diffused to neighboring patterns, instead of learning a preference function. A work in spam web page labeling [11] only exploits one of two possible user feedback options, spreads it only to web pages' descendants, and tends to use it without further refinement. In our work, the spread information is evaluated by a heuristic to assess the potential interest of patterns in the result space.

3 Core notions and notations

Let $\mathcal{D}$ the data set, $\mathcal{L}$ the pattern language, and $\mathbb{G}(\mathbb{V}, \mathbb{E})$ the graph where $\mathbb{V}$ is the set of vertices and $\mathbb{E}$ the set of directed edges. A Partial Order Graph (POG) represents the partially ordered solution pattern space (poset) of $\mathcal{L}$ under a given partial order $<$. For graph patterns, for instance, this partial order can be derived from subgraph inclusion: $g_1 < g_2 : g_1 \subseteq g_2 \Leftrightarrow g_1$ is isomporphic to a subgraph of g_2. In formal concept analysis, the POG is a lattice [13]. For each vertex $v \in \mathbb{V}$, v contains a pattern set $X_v = \{x_i\}, |X_v| \geq 1$. We therefore overload the subgraph relation $<$ to sets of subgraphs: $X_1 < X_2 \implies \exists x_1 \in X_1, \exists x_2 \in X_2, x_1 < x_2$. Each pattern set X has an associated set called support, denoted $Supp(X)$, containing elements of $\mathcal{D}$ in which X occurs: $Supp(X) = \{t \in \mathcal{D} \mid \forall x \in X, x < t\}$. As each vertex contains a pattern set, we base our definition of directed edges on $<$. The set of directed edges is defined as:

$$\mathbb{E} = \{(v_1, v_2) \mid v_1, v_2 \in \mathbb{V}, X_{v_1} < X_{v_2}, \nexists v_3 \in \mathbb{V} : X_{v_1} < X_{v_3} < X_{v_2}\}.$$

We call v_1 *parent* and v_2 *child*. We extend the relation by transitivity to parents of parents and children of children called respectively *ancestors* and *descendants*. The *lineage* of v is the set of its ancestors and descendants. We define the *roots* of $\mathbb{G}$ as the set of vertices $v \in \mathbb{V}$ having no parent. The *distance* is the minimal count of directed edges between two vertices of the POG. A layer L is the set of vertices having the same minimum distance from the roots of the POG, distance which we call *depth* of the layer.

Let L be a layer of $\mathbb{G}$. We call the layer composed of the parents of the vertices of L and the layer formed by their children the *adjacent layers*. Thus, we can build a POG in a *layered view* where the first layer contains the smallest subgraphs with the largest supports and the last layer contains the largest subgraphs with the smallest supports. Basically, the first layer contains the most generic patterns and the last layer contains the most specific patterns.

In order to integrate the user's interest in the graph, we have to introduce several notions. To this end, we define the Interest Partial Order Graph (IPOG) as follows. Let $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{V}^+, \mathbb{V}^-, weight)$ an IPOG where $\mathbb{V}^+ \subseteq \mathbb{V}$ is the subset of vertices prioritized for the exploration, $\mathbb{V}^- \subseteq \mathbb{V}$ is the subset of vertices excluded from the exploration, and $weight : \mathbb{V} \to \mathbb{R}$ is the weight function $weight(v)$ associating each vertex $v \in \mathbb{V}$ to a real number $j \in \mathbb{R}$ called weight. The next section details our method which aims to guide an expert in the pattern result space by building the IPOG.

4 The Wave Top-k Random-d Family Search algorithm

This section presents Wave Top-k Random-d Family Search (WTRFS). Our method aims to emulate the expert's subjective interest in order to guide her in the exploration of the IPOG. WTRFS relies on several components to do so, the first being the *wave exploration*. The second is the *diffusion of the expert's interest* through the structure. The last is the *exploitation of the interest* to modify the pattern search space and to iteratively sample the result space.

Table 1. Expert's interactions and their respective consequences.

Interaction	Consequences	Color
Rejected	Exclusion areas & Weights diminution	Red
Not-interested	Weights diminution	Orange
Unsure	no changes	Purple
Interested	Weights augmentation	Blue
Accepted	Prioritized areas & Weights augmentation	Green

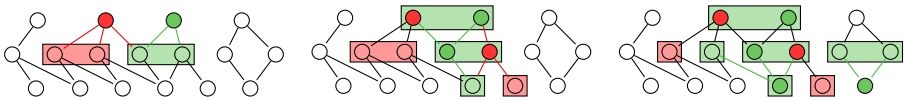


Fig. 1. Defining preferred and excluded research areas in the IPOG.

Exploration motion. The *wave motion* guides the expert from the most generic patterns to the most specific ones before rolling back, forcing her to confront her understanding of the studied patterns by comparing them with their ancestors and descendants. It also allows to diffuse the expert's interest without sampling elements lacking connections to the pattern space the expert observed. Philosophically, this is similar to the EM, or similar iterated optimization, algorithm(s) that update values then exploit them in the next iteration to inform the process.

Expert interactions and propagation of the interest. To achieve these two goals, we need a communication medium: an *interaction*. In Table 1, we list the proposed *interactions* and their *consequences*. The most radical consequence in the table is the designation of prioritized and excluded areas for the exploration. Patterns are sampled first in prioritized area, then sampled in unmarked ones and *never* in excluded areas. Therefore those consequences have to be restricted to parents and children of the reviewed vertices. Let v_1 a sampled pattern for the expert:
If the expert *accepts* v_1 then:

$$\mathbb{V}^+ \leftarrow \mathbb{V}^+ \cup \{v_2 \in \mathbb{V} | \exists (v_1, v_2) \in \mathbb{E} \text{ or } \exists (v_2, v_1) \in \mathbb{E}\} \tag{1}$$

If the expert *rejects* v_1 then:

$$\mathbb{V}^- \leftarrow \mathbb{V}^- \cup \{v_2 \in \mathbb{V} | v_2 \notin \mathbb{V}^+ \text{ and } \exists (v_1, v_2) \in \mathbb{E} \text{ or } \exists (v_2, v_1) \in \mathbb{E}\} \tag{2}$$

Figure 1 illustrates the area modifications as respectively defined in Equations 1 and 2 as consequences of *Acceptance* and *Rejection*. Figure *(a)* places the algorithm in the highest layer of the result space, i.e. the layer containing the most generic subgraphs. A sample of two patterns is shown to the expert who rejects one (in red) and accepts the other (in green). As a result, the method defines an exclusion area in the bottom layer (in red) as well as a prioritized

area (in green). If there is a conflict in the definition of the areas, it is settled in favor of the prioritized area, to keep future evaluation possible. Figure *(b)* places the algorithm in the middle layer. The method samples two patterns from the prioritized area. The interaction defines areas in the layers above and below. Figure *(c)* places the algorithm in the lowest layer containing the most specific subgraphs. This time it samples the sole prioritized patterns and a pattern in the unmarked area. The interactions designate prioritized areas in the above layer. In the next step, the algorithm will ascent to the middle layer and sample from these areas. Notably, the conflicts and confirmations induced by the expert's interest can be exploited to achieve a better comprehension of the result space.

Our second component is more subtle. It modifies the weights of the lineage of the reviewed vertices in order to impact future samples. Let v a vertex, A an expert's interaction and λ a modifier.

$$weighting(v, A, \lambda) = \begin{cases} weight(v) + \lambda & \text{if } A \in \{\text{accepted, interested}\} \\ weight(v) - \lambda & \text{if } A \in \{\text{rejected, uninterested}\} \end{cases} \tag{3}$$

However, the bigger the distance between two vertices, the more different contained patterns are. Therefore we have to consider this variation in the application of Equation (3). Hence the definition of Equation (4) where i is the distance between two vertices v_1 and v_2 as:

$$\forall v_2 \in lineage(v_1) \cup \{v_1\},\ weighting(v_2, A, |weight(v_1)| * \frac{1}{2^i}) \tag{4}$$

We note that the λ used in Equation (3) is in fact the weight of the vertices carrying the novel interaction discounted by the distance between the vertices. The factor $\frac{1}{2^i}$ embodies the decreasing correlation between the sampled subgraph and its generalization and specification. We chose this discount factor after also evaluating multiplicative and additive weights proposed in the literature [14], which lead however to similar yet less convincing results, something also found in [11]. Moreover, the modification of the vertices' weights also modify their impact when the expert interacts with them. The higher the absolute value of a weight is, the higher the impact that is obtained from the interaction. Therefore, the wider the vertex's lineage is, the stronger an impact its interaction has.

Pattern sampling. Now that we possess an efficient way to diffuse the expert's interest into the POG, we have to exploit this information in order to sample patterns from our result space. To achieve this, the method uses the IPOG weights to compute the potential interest of each vertex. A patterns' probability to be sampled increases

- with the cumulative weight of its *accepted* and *interesting* ancestors and descendants, inducing a higher chance to be interesting or accepted (*exploitation heuristic*).
- with the cumulative weight of its *unexplored* ancestors and descendants, guiding the expert towards exploring the unexplored space (*exploration heuristic*).

– if the cumulative weights of its positive and negative ancestors and descendants are *similar*, which indicates a contradiction which requires further exploration (*ambiguity heuristic*).

Following this line of argument, we compute the potential interest of v as follows:

$$I_p(v,\mathbb{G}) = \sum_{i=0}^{k}(weight(L_i^+(v,\mathbb{G})) + weight(L_i^?(v,\mathbb{G}))+ \\ (weight(L_i^U(v,\mathbb{G})) + 2 * min(weight(L_i^+(v,\mathbb{G})), weight(L_i^-(v,\mathbb{G})))) * \tfrac{1}{2^i} \quad (5)$$

In Equation (5), $L_i(v,\mathbb{G})$ is the lineage of v at depth i in $\mathbb{G}$ with:

– $weight(L_i^+(v,\mathbb{G}))$ the sum of weights of relatives with a *positive* interaction,
– $weight(L_i^?(v,\mathbb{G}))$ the sum of weights of those *without* interaction,
– $weight(L_i^U(v,\mathbb{G}))$ the sum of weights of those with an*unsure* interaction,
– $weight(L_i^-(v,\mathbb{G}))$ the sum of weights of those with a *negative* interaction,
– the factor $\frac{1}{2^i}$ reducing the influence of vertices depending on their distance.

Note that $2 * min(weight(L_i^+(v,\mathbb{G})), weight(L_i^-(v,\mathbb{G})))$ corresponds to the logical AND operator, which means that the expression rewards the likeness of $weight(L_i^+(v,\mathbb{G}))$ with $weight(L_i^-(v,\mathbb{G}))$, putting forward the lineage's ambiguity by the same occasion.

The potential interest is exploited in two types of sampling. Firstly, to encourage exploitation, the sampling selects the top k patterns ranked by I_p. Secondly, to encourage exploration, the sampling pseudo-randomly draws d patterns with the selection probability computed with I_p. Let $L \in \mathbb{G}$ a graph layer, sampling probabilities are defined as follows:

$$\forall v \in L, P(v) = \frac{I_p(v)}{\sum\limits_{\forall v' \in L} I_p(v')} \quad (6)$$

The first samples drawn from the IPOG favor vertices with numerous parents, children, and close relatives. As no interactions happened yet, the only non-zero sum of I_p is $weight(L_i^?(v,\mathbb{G}))$. Furthermore, as each vertex is initialized with the same weight, vertices with strong connectivity have the highest potential interest. This behavior is intended as it favors vertices with a wider impact, and thus stronger, on the IPOG at the beginning of the expert's exploration. This behavior more quickly leads to a good discrimination of vertices.

The Wave Top-k Random-d Family Search algorithm. Algorithm 1 takes as input an IPOG $\mathbb{G}$, an exploitation factor k setting the number of top-scoring samples, an exploration factor d setting the number of random samples, and the first and last layers to explore. As long as the user does not end the process and there are patterns to explore, the loop, lines 1 to 21, continues. In the loop, lines 2 to 19, the value i starts at depth f and ends at depth l, i is incremented at each iteration if $f < l$ or decremented if $f > l$. At line 3, we set L to the layer of $\mathbb{G}$

Algorithm 1 Wave Top-k Random-d Family Search

Require: $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{V}^+, \mathbb{V}^-, weight)$ a graph, k the number of top draws, d the number of random draws, f the first explored layer, l the last explored layer.
Ensure: $\mathbb{G}$ the graph shaped by the expert's interactions.

1: **while** $\exists v \in \mathbb{V}$ & $v \notin \mathbb{V}^-$ & v not explored **do**
2: **for** $i : f$ to l **do**
3: $L \leftarrow Layer_i(\mathbb{G})$
4: **for** $v \in L$ **do**
5: $Update_{I_p}(v, I_p(v, \mathbb{G}))$ (Equation (5))
6: **end for**
7: $\mathbb{S} = \{\text{Top } k \ v' \text{ with the highest } I_p(v', \mathbb{G})\}$
8: $\mathbb{S} = \mathbb{S} \cup \{d \ v'' \text{random patterns following Equation (6)}\}$
9: **for** $v \in \mathbb{S}$ **do**
10: $A \leftarrow Interaction(v)$
11: $weighting\_lineage(\mathbb{G}, v, A)$ (Equation (4))
12: **if** $A =$ accepted **then**
13: $\mathbb{V}^+ \leftarrow \mathbb{V}^+ \cup \{\forall v_2 \in \mathbb{V} | \exists (v, v_2) \in \mathbb{E} \text{ or } \exists (v_2, v) \in \mathbb{E}\}$
14: **end if**
15: **if** $A =$ rejected **then**
16: $\mathbb{V}^- \leftarrow \mathbb{V}^- \cup \{\forall v_2 \in \mathbb{V} | v_2 \notin \mathbb{V}^+ \text{ and } \exists (v, v_2) \in \mathbb{E} \text{ or } \exists (v_2, v) \in \mathbb{E}\}$
17: **end if**
18: **end for**
19: **end for**
20: $t \leftarrow f \ ; \ f \leftarrow l \ ; \ l \leftarrow t$ (*Reversing the tide's direction*)
21: **end while**

of depth i. We assign each vertex v in L its potential interest through the loop, lines 4 to 6, by using Equation (5). At line 7, we assign to $\mathbb{S}$ a sample of L by drawing k top patterns and d random patterns based on their potential interest at line 8. The loop, lines 9 to 18, modifies the graph $\mathbb{G}$ for each vertex v and its interaction A obtained at line 10. The vertex lineage's weights are modified at line 11 using Equation (4) and the prioritized areas $\mathbb{V}^+$ and excluded areas $\mathbb{V}^-$ are respectively updated at line 13 and 16 by using Equations 1 and 2. After a full descent or ascent, we swap f and l at line 20 in order to explore layers in the opposite direction. This allows to direct the search from the most general patterns to the most specific ones before going backwards, giving the exploration its wave-shape. Once the expert is satisfied or there are no patterns left to explore, we return the updated graph at line 21.

WTRFS produces two results. The first one is the pattern set sampled and labeled by the expert. The second is the relational graph $\mathbb{G}$ shaped by the expert's interest. This graph, through the labeled areas, the weights, the potential interest values and the labels, is a structured representation of her interest. A representation which can be observed and studied, offering a *global picture* of the search space based on *local* interactions. Through the IPOG, the expert can explore her own interest, study the relation between the patterns she chose to

Table 2. Answers' probability distributions for the probabilistic oracle

	Rejected	Uninteresting	Unsure	Interesting	Accepted
Rejected	80%	15%	5%	0%	0%
Uninteresting	10%	75%	10%	5%	0%
Unsure	5%	10%	70%	10%	5%
Interesting	0%	10%	70%	10%	0%
Accepted	0%	0%	5%	15%	80%

put forward and those that have been discarded. It also guides the expert in her evaluation of unobserved elements.

5 Designing oracles for the evaluation

One difficulty regarding interactive mining methods is their empiric evaluation. Such an evaluation requires numerous experts evaluating the method's process numerous times in order to obtain significant results which requires a costly time investment. The protocol adopted in the literature is therefore to simulate user feedback through an *omniscient oracle* that exploits an objective quality measure in order to label patterns objectively [20, 4, 2, 3, 8].

The use of such an oracle could lead to an overly optimistic evaluation, especially for an exploratory method like ours. In order to overcome this shortcoming, we test our method with *five types of oracles* simulating different possible expert behaviors. To the best of our knowledge, this is the first time that an interactive mining method has been evaluated in this way.

We assign each vertex a *hidden label* determined by the quality of the contained pattern set. The oracle assigns a *discovered label* to the reviewed vertex determined by the combination of the oracle's type and the pattern's quality.

The quality scores are converted into hidden labels so that the lowest values are *rejected*, the next-lowest labeled *uninteresting*, and so on. The quality thresholds are computed for each pattern result space in order to respect as much as possible the following distribution: 2.00% of *Rejected* labels, 18.00% of *Uninteresting* labels, 60.00% of *Unsure* labels, 18.00% of *Interesting* labels, and 2.00% of *Accepted* labels. This distribution is intended to represent the fact that an expert is not interested in the whole set of results, she is less likely to use actions affecting many patterns and more often the others.

The five oracles are:

1. *the omniscient one*: assigns to each presented vertex its hidden label.
2. *the probabilistic one*: has for each label a response probability vector inducing a fixed error percentage. In order to avoid improbable answers, each vector contains choice probabilities for each label. The idea is to give the highest probability to the right label and positive probabilities to similar labels. The more impact a choice has, the less likely that the experts is wrong, for we

Table 3. TUDatasets data sets, their size, the number of extracted subgraphs and the equivalence classes composing the POG.

data set	Graphs	Frequency	subgraphs	Equivalence classes
AIDS	2,000	10%	192	192
BZR_MD	306	10%	3,249	2,147
MUTAG	188	10%	603	110
MCF-7	27,770	10%	1,024	1,024
Mutagenicity	4,337	10%	1,904	1,880
NCI-H23	40,353	10%	1,001	1,001

consider these choices are made when the expert feels sure of herself. Probabilities are indicated in Table 2 where each row corresponds to a probability vector in which each column contains a label's probability of being chosen.

3. *the biased one*: models the expert's *prior* coming from her knowledge concerning data sets studied in the past. The prior is determined by a second quality measure whose behavior diverges from the one which determines the hidden labels, yet errors made by the oracle remain consistent with the pattern support in the target class. The measure chosen to represent the bias is *confidence*. We arbitrarily set the margin of error to 20%, i.e. to give 20% chance to the oracle to choose its answer according to the quality value of the bias rather than the quality value of the ground truth.
4. *the locally subjective one*: models the expert's behavior if she mainly focuses on the samples, leading her to label the sample by considering its top pattern as *at least* interesting and the lowest-scoring as *at least* uninteresting.
5. *the subjectively surprised one*: models an expert exploring patterns that surprise her, whether due to high quality or not. In order to coherently compute this surprisingness we use the *Outstanding Pattern Selector* introduced in [17]; patterns selected by it are labeled as *accepted* by the oracle.

6 Experiments and results

6.1 Results on publicly available data

Data set descriptions We chose six data sets with varying characteristics from the TUDataset repository[1]. Each data set contains two classes, to render the experimental setting easier. The frequent subgraphs are extracted by *quickSpan*[2] with a minimal frequency of 10%. We keep the subgraphs' order under seven vertices, following the assumption that larger subgraphs would be hard to interpret. Table 3 list the data sets' names, their size, the number of extracted subgraphs and the number of equivalence classes. Equivalence classes are computed as follows: if two subgraphs p and q have the same support $Supp(p) = Supp(q)$ and they are linked in the POG by a path passing only through subgraphs p_i having

[1] https://chrsmrrs.github.io/datasets/docs/datasets/
[2] https://gitlab.inria.fr/Quickspan/quickspan

the same support $Supp(p) = Supp(p_i)$ then they belong to the same equivalence class. In the following, we equate equivalence classes and pattern sets so each vertex of the POG contains an equivalence class (i.e. pattern set).

Result space sizes range from about 200 patterns to a few thousand. This variation helps to observe the variable or non-variable behaviors of WTRFS with respect to its application space and to get an idea about its adaptability.

In our work, we use the well-known Weighted Relative Accuracy (WRAcc) [23] quality measure based on the graph data classes defined as:

$$WRAcc(X, \mathcal{D}) = \frac{Supp(X)}{|\mathcal{D}|} * (\frac{Supp(X)^+}{Supp(X)} - \frac{|\mathcal{D}^+|}{|\mathcal{D}|}),$$

where $\mathcal{D}^+$ is a subset of $\mathcal{D}$ which contains data graphs from a target class and $Supp(X)^+$ the support of X in this subset. As patterns in the same equivalence class have the same support, they will also have the same score, the pattern presented to an expert would be one of the most *general*, or *generator*, patterns (also known as free sets in itemset mining).

Experimental protocol. In order to evaluate the effectiveness of WTRFS, each equivalence class in the IPOG is labeled with one of five interactions: Rejected, Uninterested, Uncertain, Interested, Accepted. We submit 100 samples of 3 patterns to each of the oracles described earlier. We choose to divide the 3 sampled patterns into 2 exploited patterns ($k = 2$) and 1 explored pattern ($d = 1$). As results include random elements, each data set - oracle pair is evaluated 100 times and the observed results averaged.

In order to interpret our results we study the oracles' discovered labels and the hidden labels. Let A a label type, we define the *Recall* as :

$$Recall(A) = \frac{\text{Discovered}(A)}{\text{Hidden}(A)}$$

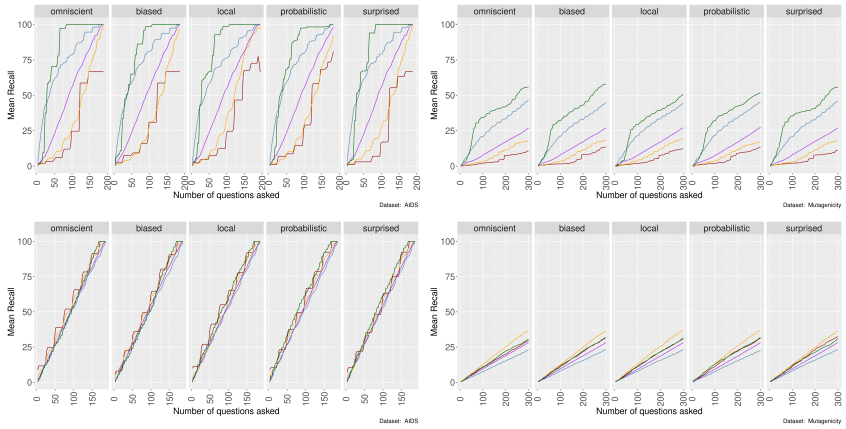
The only existing works on interactive graph mining are [2, 3] but after a number of attempts, we have not been able to acquire an implementation of either those methods, which unfortunately precludes us from a direct comparison.

Results. Evaluating 300 patterns is a tedious task. Hence the need to help the expert to discover as many accepted patterns as possible and a small portion of rejected ones in the shortest amount of time. If the presented pattern is neither accepted nor rejected, it should be at least interesting.

In Figure 2, we show results for two data sets, *AIDS* and *Mutagenicity*. Complementary studies, data sets, and program executable are available at: https://github.com/wtrfs/Wave-Top-k-Random-d-Family-Search/.

For each illustration, the x-axis indicates the number of patterns proposed to the oracle, and the y-axis indicates the *Recall*. Each column corresponds to an oracle type. The colors correspond to the types of labels (see Table 1).[3]

[3] Taken independently, recall curves are strictly increasing. However, as each run is not identical, all curves are not considered at the same time in the same layer. This is why the average curve of the results is not always strictly increasing.



AIDS *Mutagenicity*

Fig. 2. Comparison of WTRFS (top) and waved random sampling (bottom) for AIDS and Mutagenicity.

The results show the almost constant progression of the percentage of found *accepted* labels, which is particular fast in the first 50 queries. Even if in the densest networks these labels are not always all found, we notice that their percentage of discovery remains higher than those of the other labels whatever the oracle. Comparing the curves of a *wave* run where the patterns are randomly sampled (lower figures) with the curves with WTRFS where consequences are applied, we note that the results of WTRFS are clearly better. We note that for *AIDS*, in the case of random sampling, the progression of the recall is linear and quasi-equivalent for each label type. This means that for each layer, the distribution of labels is equivalent. For *Mutagenicity*, purely random sampling recovers *not interesting* patterns faster than *accepted* or *interesting* ones.

Although the omniscient oracle always gets the best results, the other oracles do not clearly degrade the result quality. In addition, the curve for the *interesting* patterns generally increases more rapidly than the others. The curves of the *rejected* patterns remain low, either for the whole experiment, or for a long period until many of the other types near exhaustion.

6.2 Results on experimental data

In the following section, we apply the WTRFS algorithm to a chemical data-set. As a data set $\mathcal{D}$ we use BCR-ABL from CHEMBL23[4], a chemical graph data set containing 1,485 molecules. The graph pattern set $\mathcal{L}$ called pharmacophores extracted from it is composed of 112,363 frequent labeled graphs the orders of which go from 1 to 7 extracted with *Norns*[5] [18].

[4] https://chembl.gitbook.io/chembl-interface-documentation/downloads
[5] https://valorisation.greyc.fr/catalog/logiciel?identifier=norns

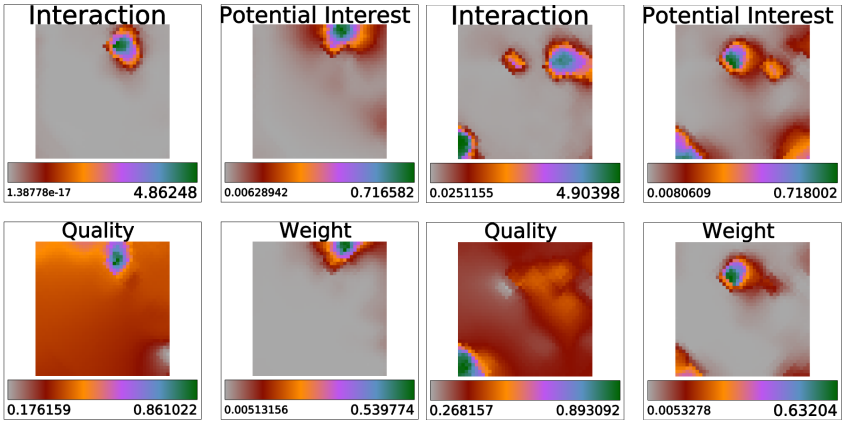


Fig. 3. Correlation between ground truth and normalized metrics used in WTRFS.

Pharmacophores are complete graphs, with pharmacophoric features as vertices, mapping given descriptors of chemical molecules in order to study their biological behavior. Each pharmacophore's support is composed of molecules which can be either labeled as active or inactive. The pharmacophores have been grouped into 1,533 equivalence classes based on identical support and structural connection in the relational POG. We note that even if we narrow down the set of patterns in the search space, the number of patterns to study still forms a combinatorial explosion. In this dataset, our class are defined with a receptor, the first class consists of active molecules and the second of inactive ones.

We generated an IPOG using WTRFS and the omniscient oracle 100 times, each vertex is described by *Potential Interest* — the mean value of I_p, *Interaction* — the mean oracle feedback where 0 indicates that the vertex has not been sampled, *Quality* — the mean WRAcc of the vertex, and *Weight* — its mean weight. Then we clustered the IPOG's vertices, using a SOM (Self Organizing Map) in Tulip[6] with a grid of 40x40, and 10 epochs. Figure 3.*(a)* shows how strongly the four measures correlate through its mountain shape with the best clusters at the center (in green) decreasing towards the worst clusters in the outer layer (in red). Especially *Weight* and *Quality*, except for the lower half of each subfigure, which corresponds to a part of the IPOG not explored. The correlation is even more pronounced in Fig. 3 (b) where we can observe a clear link between the *Potential Interest* value and *Quality*. We note that *Weight* transcribes the quality by exacerbating the minimal and maximal values, while *Potential Interest* produces a more nuanced representation of the *Quality*. We also see the effects of conflicting interactions, where *accepted* patterns surrounded by *unsure* ones inhibit interest propagation, leading to lower *Potential Interest*.

[6] https://tulip.labri.fr/site/

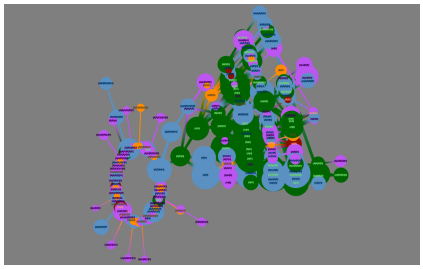


Fig. 4. Visualization of the IPOG explored by an omniscient oracle.

In Figure 4, also done with Tulip and showing a visualization of the final IPOG, vertex colors correspond to the legend described in Table 1. The size of the vertices are mapped on their weight and their label indicates the most present set of pharmacophoric features of their pharmacophores. Most visible vertices are green (i.e. accepted patterns) and blue (i.e. interesting patterns) which indicates that most of the patterns proposed to the omniscient oracle are indeed interesting ones. It also confirms that our weight maps correctly the user raw interest. From this type of figure, the expert can study the relation between the elements she finds interesting and the others. If she takes a step back and add elements she didn't already study, it can also help her in a second exploration, more static, where she uses the result weights and evaluations of her interactive exploration to infer interestingness upon unexplored patterns. All of which should help her understand her own subjective interest.

7 Conclusion

In this paper, we present an algorithm whose goal it is to guide an expert during her exploration of a result space. Our work focuses on structured patterns and spaces, in particular graph patterns and partial order graphs. The algorithm is divided into three components: the structure exploration, the sampling, and the graduated response. We determined five interactions and their consequences. Each interaction-consequence pair influences the accessible search space and the patterns' sampling.

The structure is explored through waves going forth and back from the most general patterns to the most specific ones. The goal is to accompany the expert through her understanding of the studied objects and to avoid confusing her. The interactions either modify directly the reachable result space or modify the patterns' emulated interest which is latter used to asses patterns' potential interest through heuristics and thus affect the future samples.

To evaluate our method, we simulated expert feedback using oracles based on an objective quality measure. We show that the method recovers a large number of high quality patterns, depending on the number of interactions with the oracle, even when the oracle's returns are noisy.

References

1. Al Hasan, M., Zaki, M.J.: Output space sampling for graph patterns. Proceedings of the VLDB Endowment **2**(1), 730–741 (2009)
2. Bhuiyan, M., Hasan, M.A.: Interactive knowledge discovery from hidden data through sampling of frequent patterns. Statistical Analysis and Data Mining: The ASA Data Science Journal **9**(4), 205–229 (2016)
3. Bhuiyan, M.A., Al Hasan, M.: Priime: A generic framework for interactive personalized interesting pattern discovery. In: 2016 IEEE International Conference on Big Data (Big Data). pp. 606–615. IEEE (2016)
4. Boley, M., Mampaey, M., Kang, B., Tokmakov, P., Wrobel, S.: One click mining: Interactive local pattern discovery through implicit preference and performance learning. In: Proceedings of the ACM SIGKDD workshop on interactive data exploration and analytics. pp. 27–35 (2013)
5. Bosc, G., Boulicaut, J.F., Raïssi, C., Kaytoue, M.: Anytime discovery of a diverse set of patterns with monte carlo tree search. Data mining and knowledge discovery **32**, 604–650 (2018)
6. De Raedt, L., Zimmermann, A.: Constraint-based pattern set mining. In: Proceedings of the Seventh SIAM International Conference on Data Mining. SIAM (2007)
7. Dzyuba, V., van Leeuwen, M.: Interactive discovery of interesting subgroup sets. In: Advances in Intelligent Data Analysis XII: 12th International Symposium, IDA 2013, London, UK, October 17-19, 2013. Proceedings 12. pp. 150–161. Springer (2013)
8. Dzyuba, V., van Leeuwen, M.: Learning what matters - sampling interesting patterns. In: PAKDD 2017, Proceedings, Part I. pp. 534–546 (2017)
9. Fournier-Viger, P., Gan, W., Wu, Y., Nouioua, M., Song, W., Truong, T., Duong, H.: Pattern mining: Current challenges and opportunities. In: International Conference on Database Systems for Advanced Applications. pp. 34–49. Springer (2022)
10. Gallo, A., De Bie, T., Cristianini, N.: Mini: Mining informative non-redundant itemsets. In: Knowledge Discovery in Databases: PKDD 2007: 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings 11. pp. 438–445. Springer (2007)
11. Gyongyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In: Proceedings of the 30th international conference on very large data bases (VLDB) (2004)
12. Hien, A., Loudni, S., Aribi, N., Lebbah, Y., Laghzaoui, M.E.A., Ouali, A., Zimmermann, A.: A relaxation-based approach for mining diverse closed patterns. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I. pp. 36–54. Springer (2021)
13. Kuznetsov, S.O., Obiedkov, S.A.: Algorithms for the construction of concept lattices and their diagram graphs. In: PKDD. pp. 289–300. Springer (2001)
14. Lavrac, N., Flach, P., Todorovski, L.: Subgroup discovery with cn2-sd. J. Mach. Learn. Res. **5**(2), 153–188 (2004)
15. van Leeuwen, M.: Interactive data exploration using pattern mining. In: Interactive knowledge discovery and data mining in biomedical informatics, pp. 169–182. Springer (2014)
16. van Leeuwen, M., De Bie, T., Spyropoulou, E., Mesnage, C.: Subjective interestingness of subgraph patterns. Machine Learning **105**(1), 41–75 (2016)

17. Lehembre, E., Bureau, R., Crémilleux, B., Cuissart, B., Lamotte, J.L., Lepailleur, A., Ouali, A., Zimmermann, A.: Selecting outstanding patterns based on their neighbourhood. In: IDA. pp. 185–198. Springer (2022)
18. Métivier, J.P., Cuissart, B., Bureau, R., Lepailleur, A.: The pharmacophore network: a computational method for exploring structure–activity relationships from a large chemical data set. Journal of Medicinal Chemistry **61**(8), 3551–3564 (2018)
19. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: ICDT. pp. 398–416. Springer (1999)
20. Rueping, S.: Ranking interesting subgroups. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 913–920 (2009)
21. Saha, T.K., Al Hasan, M.: Fs3: A sampling based method for top-k frequent subgraph mining. Statistical Analysis and Data Mining: The ASA Data Science Journal **8**(4), 245–261 (2015)
22. Tan, P., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. Inf. Syst. **29**(4), 293–313 (2004)
23. Todorovski, L., Flach, P., Lavrač, N.: Predictive performance of weighted relative accuracy. In: PKDD. pp. 255–264. Springer (2000)
24. Xin, D., Cheng, H., Yan, X., Han, J.: Extracting redundancy-aware top-k patterns. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 444–453. ACM, Philadelphia PA USA (Aug 2006). https://doi.org/10.1145/1150402.1150452, https://dl.acm.org/doi/10.1145/1150402.1150452