Acquisition of generic problem solving knowledge through information extraction and pattern mining

Ahmed Halioui
Département d'informatique
Université du Québec à Montréal
Québec, Canada
halioui.ahmed@uqam.ca

Petko Valtchev
Département d'informatique
Université du Québec à Montréal
Québec, Canada
valtchev.petko@uqam.ca

Abdoulaye Baniré Diallo
Département d'informatique
Université du Québec à Montréal
Québec, Canada
diallo.abdoulaye@uqam.ca

***Abstract*—In many technical domains, the generic problem solving knowledge is scarce even though a large number of concrete resolutions exist and are well documented. This makes the machine learning from resolution traces approach facing a number of challenges, not least among them the complexity of the underlying domain (concepts, relationships, events, processes, etc.) and the machine-readability of the documented resolution. We tackle here the acquisition of expertise in phylogeny, which is a notoriously rich and prolific field where hundreds, if not thousands, concrete cases are reported in the literature, yet tools to assist the phylogenist in analyzing a new dataset are virtually absent. Thus, we propose an approach that amounts to ontology-based workflow mining: Our *T-GROWLer* system abstracts general patterns from event sequences previously extracted from texts. It comprises two modules –a workflow extractor and a pattern miner– both relying on a pair of ontologies (a domain one and a procedural one).**

***Keywords*-Workflow pattern mining; Information extraction; Ontology completion**

I. Introduction

In many technical domains, the resolution of challenging problems unfolds as a complex process comprising multiple steps executed according to loosely defined order. Moreover, a step of the process typically involves methods and tools chosen among a long list of of available items and requires many parameters to be set, often in a mutually dependent fashion. Phylogeny is one such domain where the analysis of a genomic dataset is knowingly delicate task as the domain covers a huge number of relevant concepts, relationships, instances, etc., whereas the resolution is supported by a large variety of software tools implementing concurrent analysis methods. As a consequence, despite the large number –in the thousands– of well-documented solutions to concrete analytical problems, a dare lack of generic problem-solving knowledge is to be observed in the field. This clearly hampers the progress in the field and therefore motivates the application of machine learning/data mining methods, e.g., workflow mining, to the elicitation of analytical expertise from resolution data. Yet such an application will have to face significant challenges, not the least among whom are the complexity of the underlying domain and the degree of formality in the resolution descriptions (mostly free text as published in the specialized literature).

Workflow (aka *process*) *mining* aims at extracting generic process patterns from event logs [1], [2]. Different types of workflow representations are proposed in the literature [1], [3], [4]. Yet the representational bias of each model should be considered while mining patterns: each representation delimits the search space of patterns and circumscribes the expressive power of the underlying models. Moreover, there is a large gap between concrete workflows and generic models and bridging it requires to explicitly represent knowledge at different abstraction levels.

In this paper, we adopt an ontological approach to represent the different levels of expertise. Ontologies offer shared vocabularies to facilitate tasks of knowledge integration and interpretation via triples of concepts and relations. Building ontologies is a proliferating research topic in artificial intelligence, bioinformatics and computational medicine [5], [6]. Today, ontologies are used to represent data via concepts and relations for semantic enrichment. We propose to use ontological representations for two purposes: (1) the extraction of workflow terms and relations from texts, and (2) the extraction of generalized (abstract) workflow patterns. Our approach of pattern mining from scientific texts follows an ontology-based representation of experience within scientific texts. In this study, we distinguish two types of knowledge in order to represent experience and skills: theoretical knowledge and processual knowledge. The former is used to build a domain ontology representing concepts and relations of one domain analysis basics (theory). While a processual ontology is used to represent another dimension of knowledge which is domain practices and software experience. It is specifically used to extract abstract but relevant patterns from texts. Concepts and relations of processual ontology represent explicitly elements of workflows (events and activities). Here, we distinguish two types of workflows: (1) concrete workflows which elements are objects (events) and links between them and (2) abstract workflows representing more general concepts and relations.

Only the processual ontology is used in the extraction steps of these two kinds of workflows.

The rest of the paper is organized as follows: the next section presents related work and background information on workflow pattern-based mining. In section 3, we define the pipeline of our approach and the models representing the different kinds of knowledge. We also describe the different processes of concrete workflow extraction from texts and generalized pattern mining. The proposed model is applied to the phylogenetic workflows where processes describe tools and methods used in the phylogenetic inference. Section 4 shows the phylogenetics domain knowledge. Section 5 presents the experimental evaluation and results.

II. Background and related work

Given a universe of items (events) O, a database $\mathcal{D}$ of records (processes) that combine items from O, and a frequency threshold σ, pattern mining's goal is to extract the family $\mathcal{F}_\sigma$ of patterns present in at least σ records. Two binary relations are generally used in a pattern mining problem combining hierarchies: the *generality* [1] between patterns $\sqsubseteq_\Gamma$ (where Γ is a pattern language) and the instantiation $\precsim$ between a data record and a pattern. In the simplest settings, both data records and patterns are sets of items (itemsets). Hence, the mining goal is to find all the frequent subsets of a family of sets $\mathcal{D} \subseteq \Delta$ where Δ is the data language. We find in the literature more elaborate data structures such as sequences, episodes and graphs [7]. In the first studies on sequential pattern mining, several work, about including *hierarchies* in the mining process in order to enrich semantically generated patterns, are proposed. This research axis looks for generalized pattern languages built on the top of concepts C and abstracting objects from O. Concepts from a taxonomy H represent *is-a* relationships. In fact, a taxonomy is a simple domain ontology. The second wave of studies on generalized pattern mining uses ontologies as a Directed Acyclic Graphs (DAG). Furthermore, we find that ontologies are also used in the different steps of data mining: (a) in the preprocessing step to filter infrequent items, (b) while generating patterns (in the joining step) or (c) in the postprocessing step in order to prune irrelevant patterns.

Among the first researches on using knowledge representations in pattern mining domain, we find the work of [8]. Authors propose three data structures: HG-trees (generalized hierarchical trees), AR-rules (attribute-relation rules) and constraints on the environment EBC. Inspired from this work, in [9], the authors used hierarchies in pattern mining process. Their algorithm GSP (Generalized Sequential Patterns) searches for $\mathcal{F}_\sigma$ through the pattern space $\langle \Gamma, \sqsubseteq_\Gamma \rangle$ by exploring the monotony in frequency with regard to the generality operator $\sqsubseteq_\Gamma$. Their *Apriori* miner performs a level-wise top-down traversal of the pattern space. On itemsets, it examines patterns at level k - that is, of size k - on two points: the method computes the frequency of candidate patterns by matching them against the records in $\mathcal{D}$ whereas it generates $k+1$ candidates by combining pairs of frequent k-patterns. However, their approach is expensive in terms of memory and computation. Their representation of $\mathcal{D}$ becomes huge while exploring deeper levels on the hierarchy. Han in [10] uses associations in one level at a time. While Fortin *et al.* [11] used an oriented object representation to generate association rules of different levels of abstraction.

Zhou *et al.* in [12] represent apriori information in a DAG in order to calculate numerical dependencies (probabilities) between concepts C and items from O. Furthermore, Philips *et al.* in [13] proposed Prolog ontologies in an inference rule-based system using the first order logic language. Cevspivova *et al.* integrated ontology in the different steps of mining in the model CRIPS-DM. They publish a new miner afterwards called 4ft-Miner used in mining medical processes [14]. In [12], authors proposed a Generalization Impact measure based on support in order to estimate the sufficient level of abstraction to generate interesting generalized patterns.

Finally, in [15], a fully-blown ontology is used. First, the data sequences are enriched with domain properties from the ontology. The resulting structures are akin to labeled digraphs, in which vertices are objects and edges represent links between objects or the induced sequential order. Patterns are generalized ontological digraphs, i.e., made of sets of ontology classes connected by properties (plus sequencing). Matching between data sequences and patterns is defined on-top of the standard ontological instanciation between objects and classes and represents an order preserving injective graph morphism. The *xPMiner* method [15] applies an Apriori-like level-wise mining whereby levels in the pattern space are defined w.r.t. the overall depth of the pattern elements within the ontology. Candidates of level $k+1$ are generated from k-level frequent patterns by a *refinement operator* (see for instance [16]) made of four types of primitives: 1) add (append) a root concept to the sequence, 2) replace a concept from the sequence by a direct specialization thereof, 3) add a root property between two concepts from the sequence, and 4) replace a property by a direct specialization. A major shortage of the proposed approach is the cost of the plain pattern-to-data matching operation. Indeed, as no optimization is proposed, the large number of labeled digraph morphism computations take a significant toll on the method's performances. In our own study, we choose to enhance that approach (see below).

Overall, in our approach, data records are extracted from a unstructured sources, e.g., scientific texts. To the best of our knowledge, no ontology-based pattern mining system exists using data extracted from texts. Only text mining systems based on patterns have been proposed [5]. In what

[1] also know as the raising operation

follows, we present the *T-GROWLer* system which elicits experience in the form of generalized workflows mined from concrete ones, themselves acquired by text analysis of textual sources.

III. T-GROWLER, AN ONTOLOGY-BASED WORKFLOW MINING SYSTEM

We describe below the two main components of *T-GROWLer*: (A) workflow extraction from texts and (B) workflow mining (see Fig. 1). Both rely on a pair of ontologies, a domain and a processual one, as guidance (see Fig. 1).

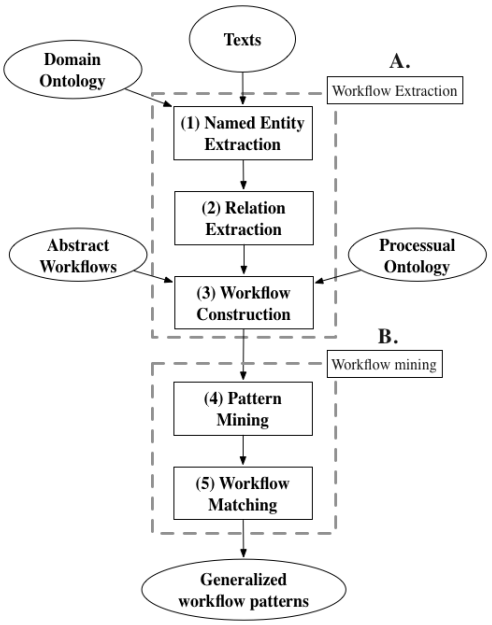

Figure 1. Generalized-workflow mining from texts.

A. Workflow extraction from texts

Component A extracts automatically information from texts and organizes them in two ontologies: a domain ontology and a processual ontology. The first database is used in order to extract terms (named entities and relations) from texts. The second one reorganizes extracted data based on a processual schema to generate workflows. Formally, an ontology is a four-tuple $\Omega = \langle C, \mathcal{R}, \sqsubseteq_\Omega, \rho \rangle$. C is a set of concepts, R is the set of domain relations. The generality order $\sqsubseteq_\Omega$ holds both for concepts and relations. ρ is a ternary relation $C_1 \times R \times C_2$ which connects two concepts with one relation. C_1 is called the *domain* concept and C_2 is the *range* (for example: Fig. 7). Our workflow extraction method takes place in 3 different steps: named entity extraction, relation extraction and workflow reconstruction. We recognize and annotate automatically terms in texts with a hybrid model based on Perceptron and SVM [17].

We use morpho-syntactic techniques in Natural Language Processing (NLP) domain: tokenization, stemming and POS (Part Of Speech) tags to annotate texts. We have developed a supervised approach based on context features in order to recognize and learn concepts from texts [18]. A handwriting specific grammar is described (in JAPE [2] language) for each ontological concept and relation to extract and filter the right instances. For example, when a program's term is a noun (such as IDEA, NETWORK or PHASE) or a verb (such as aLIGN), categorial and word sens semantic ambiguities are presented. We solve this problem by modeling rules to verify if the concerned word is not written in lowercase. Generally program's names are written with similar typographies (with upper-case letters). We use other specific handwritten patterns for each concept based on their syntactic morphologies [19].

After dealing with semantic ambiguities, a learning dataset is built. Two sets of features have been calculated on terms in order to learn them. These features represent term's contexts in POS annotations and surrounded concepts in a window of five next and previous concepts and tokens.

For example, from the following sentence: "*The 16S gene sequenced in this work was aligned to homologous sequences obtained from the Genbank using the ClustalX program.*", we construct the set of features:

{_NPOSC_VBN[-3] _NPOSC_IN[-2] _NPOSC_DT[-1] _NPOSC_NNP[0] _NPOSC_VBG[1] _NPOSC_DT[2] _NPOSC_NNP[3] _CLASSC_[-3] _CLASSC_[-2] _CLASSC_GENE[-1] _CLASSC_[0] _CLASSC_ALIGNP[1] _CLASSC_[2] _CLASSC_[3]]},

where the current term [0] is "*Genbank*". _NPOSC_VBN[-3] for example indicates that the third term preceding *Genbank* is a verb in past participle. _CLASSC_ALIGNP[1] indicates that the next concept after *Genbank* is an alignment program term.

Furthermore, a PAUM model [17] is used to learn terms with the extracted features. PAUM (Perceptron algorithm with uneven margins) was designed especially for imbalanced data and has successfully been applied to various named entity recognition problems [17]. For the relation extraction module, we propose a distant supervision approach [19] taking tuples from the domain ontology as input and annotate text in a semi-supervised way [20]. We adopt this hypothesis for the relation extraction process: a link (an instance of a relation) between two concepts should exist in the same sentence evoking instances of these concepts. Such link is represented in the verb located between concepts' terms. In order to recognize tuples in texts, we designed morpho-syntactic rules for each relation. For example the relation $isAlignedWithHomologsIn$ is defined with specific terms in the context of the verb *align*. This relation

[2]Java Annotation Patterns Engine, https://gate.ac.uk/sale/tao/splitch8.html

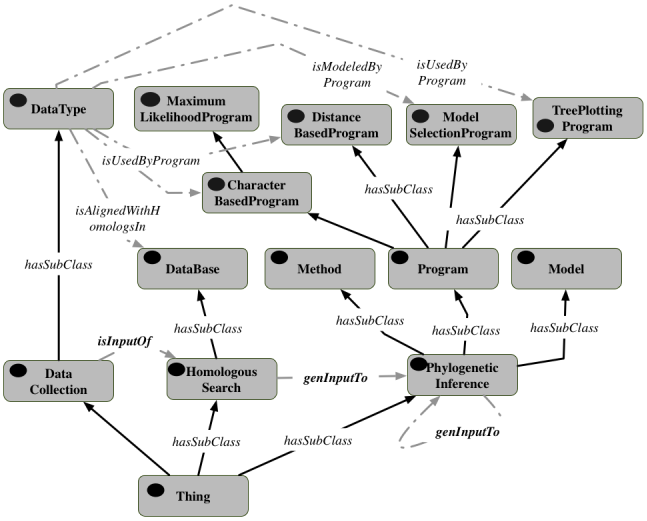

Figure 2. A portion from the phylogenetic processual ontology (PhylOntology). On the bottom, we see abstract concepts ordered by the *isInputOf* and the *genInputTo* properties. We describe the phylogenetic inference in 3 big steps: $DataCollection$, $HomologousSearch$ and $PhylogeneticInference$. Each class/concept contains different subclasses related to each other with different levels of relations. For example, a $DataType$ (*e.g.* DNA sequence) is aligned in a $Database$ used by a $CharacterBasedProgram$, a $TreePlottingProgram$ and modeled by a $ModelSelectionProgram$ (see section IV for more details).

should have as a domain a $DataType$ term and as a range a $Database$ term (see Fig. 2). A homologous alignment relation verb (for instance *align*) should be followed or preceded by the term *homologs* or *homologous* in the same sentence. From recognized links between named entities in text, we build a concrete workflow knowledge database where items are named entities ordered by the *isInputOf* and *genInputTo* properties (relations). Workflows as traceable graphs represent sequences of events. Based on the most abstract workflows, we ordered these events in order to construct a workflow database. For example, from the tuples: (ρ_1) (16sGene, "isInputOf", ClustalW), (ρ_2) (ClustalW, "genInputTo", PHYLIP) and (ρ_3) (PHYLIP, "genInputTo", TreeView), we construct the workflow sequence $\langle$ 16sGene, ClustalW, PHYLIP, TreeView $\rangle$. This latter is enhanced with ontological properties and concepts from the processual ontology to construct generalized workflows.

B. Mining generalized workflows

Component B implements a level-wise approach for the discovery of generalized workflows that is an extension of the method repsented in [15]. Thus, given a workflow database $\mathcal{D}$ and an ontology Ω, it extracts the family $\mathcal{F}$ of all structural patterns which match at least δ records.

Our approach relies on two descriptive languages to represent the pattern space. A data language Δ_Ω expresses data records that are sequences of inter-related events and represent concrete workflows. These are typically extracted from texts whereby the identified phylogeny objects are assigned resource IDs from Ω. As an illustration, Fig. 3 provides a set of workflow object sequences. Furthermore, each record s from Δ_Ω is made of a sequence $s.\zeta$ of objects (concrete programs and datatypes in phylogeny) and a set $s.\theta$ of links between the objects from that induce a DAG structure on top of $s.\zeta$. Hereafter, the sequence $s.\zeta$ will be assimilated to the entire record s that will be termed itself sequence.

A second language Γ_Ω represents the generic patterns. Again, a pattern S is a pair made of a concept sequence $S.\zeta$ and a set of relations $S.\theta$. Generalized pattern are built on top of a set of concepts C abstracting objects from the ontology Ω. For example, in Fig. 3, S is a generalized pattern describing a workflow abstractions of programs used in a phylogenetic analysis. Patterns are related to data sequences through a matching relationship that is akin to sub-graph isomorphism. Moreover, more general patterns are located higher in the hierarchical structure of the pattern space $\langle \Gamma, \sqsubseteq_\Gamma \rangle$. The goal is to search for all frequent generalized workflows $\mathcal{F}$ across that space by exploring the anti-monotony of pattern support. Apriori is the prototypical pattern miner that performs a level-wise top-down traversal of the pattern which relies on generation of $k+1$-level patterns from k-level ones.

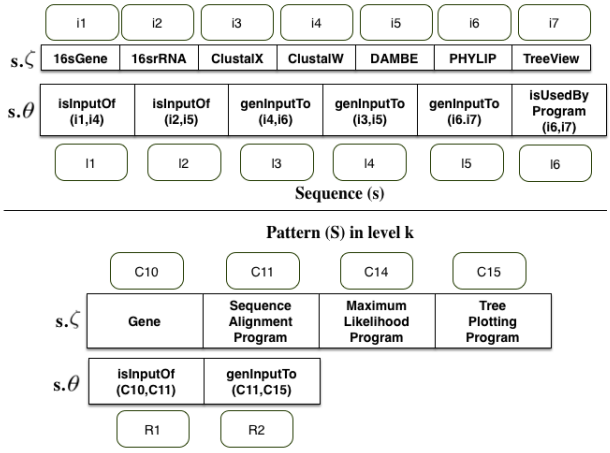

Figure 3. Workflow sequence S and pattern S representations.

Following , the mining method in [15] moves downwards in the pattern space by applying a refinement operator made of four *canonical* operations whose immediate effect is to generate a set of candidates patterns: (1) AddCLS - adds a concept to a pattern, (2) AddREL - adds a relation between two concepts, (3) SplCLS - instantiates a concept and (4) SplREL - instantiates a relation in a pattern. For instance, AddREL adds a property relating two concepts (a domain and a range) in a workflow pattern (see Fig. 5). At the following step, the support for each candidate is computed, using the matching primitive, i.e., confronting a candidate to

each data sequence. Such a procedure is redundancy-prone and hence would harm the efficiency of the global mining process.

To that end, we introduce a novel data structure which factors out repeated pattern matchings while calculating their supports. Thus, instead of calculating *all* concept and relation matchings in a pattern S for each level k, our method reuses the results of the previous matching (on the parent from the space). It therefore starts from the latest operator on the concept/relation in the pattern. For example (see Fig. 4 and Fig 5), in the iteration k, S is matched with s in ζ: $i1 \sqsubseteq C10$, $i4 \sqsubseteq C11$, $i6 \sqsubseteq C14$ and $i7 \sqsubseteq C15$. If the next operation in the $k + 1^{th}$ iteration is to add the relation, e.g., $AddRELR3(C14, C15)$, then the matching data structure of S should put the relation in the proper position without scanning all previous concepts and relations. Our matching structure adds the domain and the range of a link by connecting the corresponding concepts and guaranties that the matching of the whole sequence is valid.

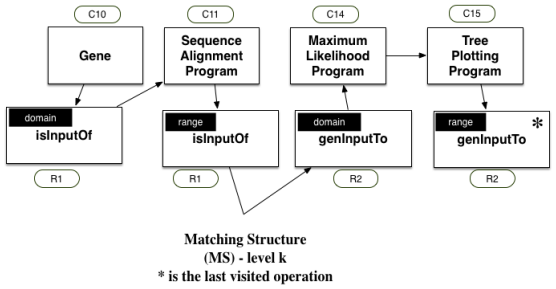

Figure 4. Matching structure of a pattern S.

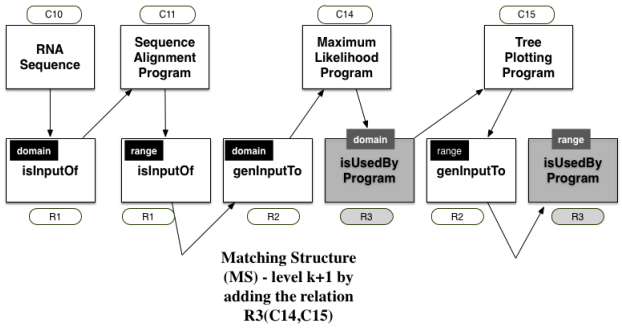

Figure 5. Extension of a pattern S by adding a relation $R3$ (grey boxes).

In fact, the proposed matching structure MS is a "stack" comprising the results of the canonical operations (adding/specializing concepts/relations) on a pattern. For example, the arrows in Fig.5 illustrate the way we obtain a pattern S in each candidate generation level. Adding the relation $R3$ between the concepts $C14$ and $C15$ amounts to verify if the matching between S and s is still valid. The matching verification begins from the last operation $AddR(R3)$. Our matching data structure MS saves all previous matching IDs of a workflow pattern and searches/verifies if the last added operation still satisfies matchings constraints.

IV. Case study: phylogenetic workflows

Phylogeny represents the evolutionary history of a group of organisms having a common ancestor. Inferred from nucleic acid (DNA or RNA) or protein sequences, phylogenetic relations are rediscovered with tools detecting different evolutionary phenomena such as duplications, insertions and deletions [6]. During phylogenetic tasks, multiple bioinformatics tools and methods are used [21] which complicates the comparison of different studies even for the same set of species. Platforms like Phylogeny.fr, Taverna [6] or Armadillo [22] propose different tools and web services. However each platform represents a completely different level of phylogenetic knowledge: from workflow based-systems to web services and programming language packages. However, the complex nature of bioinformatics data (as heterogeneous, spread and dynamic data) makes the knowledge discovery task very difficult.

Phylogenetic workflows follow a general pipeline of processes: (i) data collection, (ii) homology search, (iii) phylogenetic inference (iv) statistical testing and data resampling and (v) tree visualization and annotation. In the first step, a molecular data (sequences or alignments) of a set of species (taxa) is collected. In the second step, sequences are aligned with homologs ("similar sequences"). In the third step, phylogenetic trees describing evolutionary history of the concerned taxa are inferred. The fourth step is about testing the evolutionary hypothesis of the phylogenetic analysis and the last step is to visualize results and annotate sequences. This general pipeline is used in the construction of workflows from texts. Extracted relation instances are ordered following this general workflow. We redisgned the latter in the processual ontology so the concepts are ordered by their $genInputO$ and $isInputOf$ relations (see Fig. 2). For thi study, we downloaded scientific articles from the digital repository PubMed Central (PMC) [3] database (articles published from 2013 until April 2015) in a XHTML format. Over 2,000 texts have been extracted. In addition, we constructed a terminology gazetteer from different well-known databases in the phylogenetic literature such as NCBI Unigene, Uniprot, Taxonomy Browser and KEGG Disease [4]. The terminology is then reorganized in a domain ontology in order to recognize terms and relations in texts like presented in module A of the proposed model. Next, we built a processual ontology to discover workflow patterns within annotated terms in texts.

[3]http://www.ncbi.nlm.nih.gov/pmc/

[4]More than 1,800,000 terms have been extracted.

V. Experiments and results

A. Named entity learning results

We used a PAUM model to recognize the different phylogenetic concepts in texts. The Gate implementation of PAUM[5] (with default parameters) is used to assess our supervised learning annotation system. The learning set is built from the gazetteer terminology and the specific grammar for each concept. A 10 folds cross validation of 10 folds presents results in terms precision and recall (see table I). Our proposed model with context features has well learned and applied 7 different domain concepts: DataType, Taxon, Gene, Protein, Disease, Method and Program. DataType is a concept including different data type sequences: DNA, RNA, protein, alignment, etc. 9366 terms have been extracted from the phylogenetic analyses sections (with 95% of F-measure). The overall F-measure for both sections (abstracts and phylogenetic analyses) is 96%. Indeed, with only context POS (Part Of Speech) annotations and concept neighbors, we could recognize the 7 different classes. Specific vocabularies and gazetteers are able to distinguish different concepts in the texts.

Table I
Evaluation results of the automatic annotation system on phylogenetic analyses sections - extracted from articles published from 2013 to April 2015

Concepts	N.Instances	Precision	Recall
DataType	9366	0.97	0.92
Taxon	8867	0.98	0.94
Gene	922	1.00	0.95
Protein	485	0.99	0.97
Disease	6	0.30	0.30
Method	2671	0.99	0.96
Program	2132	0.99	0.90
Overall	24449	0.98	0.93

A gold standard list of terms and a corresponding acquisition corpus were built. An expert has extracted manually about 1,000 terms (and classified them into the different concepts) from 100 phylogenetic sections to build a reference terminology. Each extracted term has been compared to the reference entities in order to validate its domain's precision and recall (see figure 6).

Evaluating extracted terms with the expert reference terminology shows that only 6 classes have been well extracted with our PAUM model. Extracted diseases terms weren't well identified by the expert. This is expected, since diseases terms aren't frequent in phylogenetic analyses sections. Only 6 terms have been extracted with our model and 3 are recognized by the expert. However, for the other concept's instances, our model did recognize terms with 73.95 % of F-measure.

High precision and recall of phylogenetic concepts is due to the quality of our input (various well-known domain-terminologies) and the grammar used for the disambiguation (as heuristic rules). However, while the ground truth is generated with a semi-automatic way (from the terminologies and the specific disambiguation grammar), a gold standard validation shows lower precision and recall. Other domains evaluation depends hence on the quality of the terminology vocab and the expert gold standard validation.

[5]https://gate.ac.uk

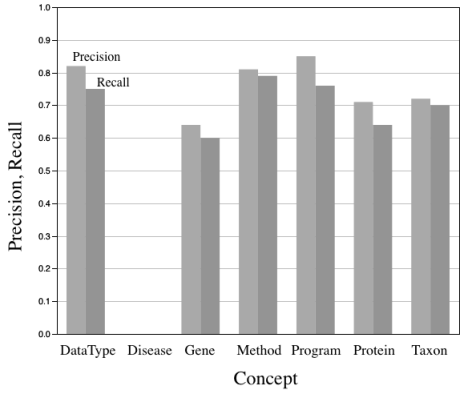

Figure 6. Gold standard evaluation results.

Next, we have manually built 50 workflows in order to execute the generalized pattern mining phase. Concepts and properties terms are then organized in the processual ontology. This latter presents 13 different object properties (links), 68 concepts (with 3 concept roots) and 560 individuals (terms). Our workflow database (PhyloFlows) is composed from 50 workflows with an average of 11 items (terms) and 9 links per workflow. Both, PhyloFlows and the phylogenetic processual ontology represent the input of the next module (B) of the proposed model (see Fig. 1). We present in the next section the different results of the workflow-pattern mining process.

B. Generalized-pattern results

In order to evaluate our pattern mining system *T-GROWLer*, we present different experimentations. We tested the new matching algorithm in terms of speed (mining time) and scalability (see Fig. 7). *T-GROWLer* outperforms the *xPMiner* system [15] with 2 times order of magnitude in high threshold supports (>0.8). With 60% of minimum support, our approach executes 10 times faster the mining algorithm than the *xPMiner* algorithm.

We also tested our method with a different database. *eTP-tourism* (Electronic Tourism Platform) worfklow-base is a portion of a log file describing web pages navigation of 56 different users in a Web travel site [15]. This database has an average of 19 items and 6 links per sequence and enhanced by an ontology of 157 concepts and 35 object properties. Fig. 7- B shows the difference in mining time between PhyloFlows and eTP-tourism. Running time does eventually increase when more concepts and links are used. Our system

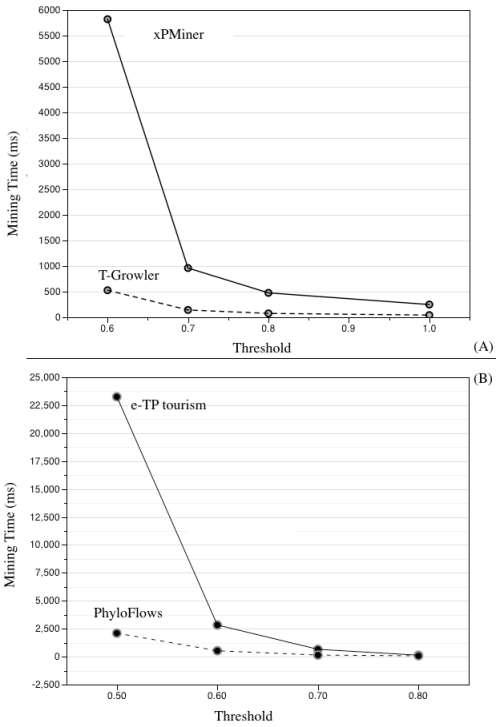

Figure 7. (A) Comparing the mining time of *T-GROWLer* and *xPMiner*. (B) Varying support for PhyloFlows and e-TP tourism databases.

spends about 10 times the mining time to generate patterns with 500 items and 100 links more. While *eTP-tourism* database presents more dense graphs, matching sub-graphs and mining processes become heavier to handle. However, taking 23 seconds to do the whole process of matching and generating about 6,000 graph patterns is still a reasonable time.

We present next, workflow mining results obtained from the *PhyloFlows* database. We remind that a pattern S is a set of concepts and relations. Relations are couples of domain and range concepts. For instance:$(1, 2) = [isAlignedIn]$ represents the relation $isAlignedIn$ between the 1^{st} and the 2^{nd} concept in the pattern S. We show in table II a sample of 3 different generated patterns ordered by their threshold support *Th*.

All generated patterns present different levels of concepts and relations.. For example, in 80% of the cases, we find that a phylogenetic analysis should have at least three ordered steps: data collection, multiple sequence alignment and a phylogenetic inference program. However, these kinds of patterns are generally well known by the community. While decreasing the threshold support, we find more interesting patterns like when selecting DNA sequences, a homology search is done with a GTR model for a character based inference method and a visualization tool (pattern 2). This sequence of concepts is enhanced with different kinds of

Table II
EXAMPLES OF GENERATED PATTERNS FROM THE PHYLOFLOWS WORKFLOW DATABASE

	Pattern		**Th**
	Sequence of concepts	***Set of relations***	
1	⟨DataCollection, MultipleSequenceAlignment, PhylogeneticInference⟩	{(1,3)=[isInferredBy], (1,2)=[genInputTo], (2,3)=[genInputTo]}	80%
2	⟨DNASequence, HomologySearch, GTRModel, CharacterBasedProgram, TreePlottingProgram⟩	{(1,2)=[isAlignedIn], (1,3)=[isModeledBy], (1,4)=[isUsedByProgram], (1,5)=[isUsedByProgram], (1,2)=[isInputOf], (2,3)=[genInputTo], (3,4)=[genInputTo], (4,5)=[genInputTo]}	40%
3	⟨ProteinSequence, HomologySearch, JTTModel, ParsimonyProgram⟩	{(1, 2)=[isAlignedIn], (1,3)=[isModeledBy], (1,4)=[isUsedByProgram], (1,5)=[isUsedByProgram], (1,2)=[isInputOf], (2,3)=[genInputTo], (3,4)=[genInputTo], (4,5)=[genInputTo]}	35%

links like which alignment program is used or which kind of inference method has been adopted in these kinds of studies (table II - third column).

These kinds of patterns could be used for a better recommendation considering their abstract but semantically richer meanings. For example, if we know that the $16s\ rRNA$ gene is used by the $ClustalW$ alignment program in 10% of the cases, then a concrete recommendation could offer $MrBayes$ as phylogenetic reconstruction program. However, offering any $Character-basedInferenceProgram$ is a more interesting recommendation due to its higher frequency (40%) in the database as it is a more "general" concept.

VI. CONCLUSION

Our proposed approach of mining generalized patterns from texts proves that different levels of knowledge could be extracted from the phylogenetics literature. Expertise and domain knowledge from articles show interesting patterns to mine. Different ontologies have been used to represent the different types of knowledge into concepts and relations. Extracted workflows are enhanced with ontologies' annotations to produce generalized frequent patterns. These patterns, covering a wider range of values then specific terms, could be used to assist practitioners with a richer vocabulary during their tasks. The elicitation of analytical expertise from resolution data proves that richer workflow patterns could be extracted. Our proposed model could also be applied to different domain-based processes when expertise is confronted to domain knowledge. Video games, for example, is a good domain application to our approach when gamers are constantly faced to cognitive abstract gestures (*i.e* expertise). Workflows of events could be extracted from games sessions

and mined in order to predict and recommend generalized patterns like what type of event is going to happen during a game or what kind of item a player needs to buy to accomplish a mission.

ACKNOWLEDGMENT

This work has been parially supported by the NSERC Dicovery Grants of Canada of the 2nd and 3rd authors.

REFERENCES

[1] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Trans. on Knowl. and Data Eng.*, vol. 16, no. 9, pp. 1128–1142, Sep. 2004.

[2] C. A. Ellis, A. J. Rembert, K.-H. Kim, and J. Wainer, "Beyond workflow mining," in *Business Process Management*, ser. Lecture Notes in Computer Science, S. Dustdar, J. L. Fiadeiro, and A. P. Sheth, Eds., vol. 4102. Springer, 2006, pp. 49–64.

[3] W. M. P. van der Aalst and A. Weijters, "Process mining: a research agenda," *Computers in Industry*, vol. 53, pp. 231–244, 2004.

[4] N. Hashmi, "Abstracting workflows: unifying bioinformatics task conceptualization and specification through semantic web services," *on Semantic Web for Life.*, no. October, pp. 27–28, 2004.

[5] Y. Tsuruoka, Y. Tateishi, J.-D. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii, "Developing a robust part-of-speech tagger for biomedical text," in *Advances in informatics*. Springer, 2005, pp. 382–392.

[6] M. Anisimova, D. a. Liberles, H. Philippe, J. Provan, T. Pupko, and A. von Haeseler, "State-of the art methodologies dictate new standards for phylogenetic analysis." *BMC evolutionary biology*, vol. 13, p. 161, Jan. 2013.

[7] C. C. Aggarwal and J. Han, Eds., *Frequent Pattern Mining*. Springer, 2014.

[8] B. Mobasher and S. S. Anand, "The role of domain knowledge in data mining," in *Proceedings of the 4th int. conference on Information and knowledge management*. ACM, 1995, pp. 37–43.

[9] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Proceedings of the 5th int. Conference on Extending Database Technology: Advances in Database Technology*, ser. EDBT '96. London, UK, UK: Springer-Verlag, 1996, pp. 3–17. [Online]. Available: http://dl.acm.org/citation.cfm?id=645337.650382

[10] J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," in *VLDB*, vol. 95, 1995, pp. 420–431.

[11] S. Fortin and L. Liu, "An object-oriented approach to multi-level association rule mining," in *Proceedings of the fifth int. conf. on Information and knowledge management*. ACM, 1996, pp. 65–72.

[12] X. Zhou and J. Geller, "Raising, to enhance rule mining in web marketing with the use of an ontology," *Data Mining with Ontologies: Implementations, Findings and Frameworks*, pp. 18–36, 2007.

[13] J. Phillips and B. G. Buchanan, "Ontology-guided knowledge discovery in databases," in *Proceedings of the 1st int. conf. on Knowledge capture*. ACM, 2001, pp. 123–130.

[14] H. Cešpivová, J. Rauch, V. Svatek, M. Kejkula, and M. Tomeckova, "Roles of medical ontology in association mining crisp-dm cycle," in *ECML/PKDD04 Workshop on Knowledge Discovery and Ontologies (KDO04), Pisa*, 2004.

[15] M. Adda, P. Valtchev, R. Missaoui, and C. Djeraba, "Toward recommendation based on ontology-powered web-usage mining," *IEEE Internet Computing*, vol. 11, no. 4, pp. 45–52, 2007.

[16] L. De Raedt, "Logical and relational learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5249 LNAI, 2008, p. 1.

[17] Y. Li, K. Bontcheva, and H. Cunningham, "Using uneven margins svm and perceptron for information extraction," in *Proceedings of the Ninth Conference on Computational Natural Language Learning*, ser. CONLL '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 72–79.

[18] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[19] M. Poesio, E. Barbu, C. Giuliano, and L. Romano, "Supervised relation extraction for ontology learning from text based on a cognitively plausible model of relations," in *ECAI 2008 3rd Workshop on Ontology Learning and Population*, 2008, pp. 1–5.

[20] N. Bach and S. Badaskar, "A survey on relation extraction," *Language Technologies Institute, Carnegie Mellon University*, 2007.

[21] J. W. Leigh, F. J. Lapointe, P. Lopez, and E. Bapteste, "Evaluating phylogenetic congruence in the post-genomic era," *Genome Biology and Evolution*, vol. 3, pp. 571–587, 2011.

[22] E. Lord, M. Leclercq, A. Boc, A. B. Diallo, and V. Makarenkov, "Armadillo 1.1: An Original Workflow Platform for Designing and Conducting Phylogenetic Analysis and Simulations," *PloS one*, vol. 7, no. 1, p. e29903, Jan. 2012.