Querying a Bioinformatic Data Sources Registry with Concept Lattices

Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, Malika Smaïl-Tabbone

Querying a Bioinformatic Data Sources Registry with Concept Lattices

Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, and Malika Smaïl-Tabbone

UMR 7503 LORIA, BP 239, 54506 Vandœuvre-lès-Nancy, FRANCE
{messai,devignes,napoli,smail}@loria.fr
http://www.loria.fr/equipes/orpailleur

Abstract Bioinformatic data sources available on the web are multiple and heterogenous. The lack of documentation and the difficulty of interaction with these data banks require users competence in both informatics and biological fields for an optimal use of sources contents that remain rather under exploited. In this paper we present an approach based on formal concept analysis to classify and search relevant bioinformatic data sources for a given user query. It consists in building the concept lattice from the binary relation between bioinformatic data sources and their associated metadata. The concept built from a given user query is then merged into the concept lattice. The result is given by the extraction of the set of sources belonging to the extents of the query concept subsumers in the resulting concept lattice. The sources ranking is given by the concept specificity order in the concept lattice. An improvement of the approach consists in automatic refinement of the query thanks to domain ontologies. Two forms of refinement are possible by generalisation and by specialisation.

1 Introduction

Bioinformatics is facing the great challenge of enabling biologists to effectively and efficiently access to data stored in distributed data sources. The large number of sources, their heterogeneity and the complexity of the biological objects they refer to, make it difficult to adequately relate the sources with a user query. The query itself often needs to be processed and distributed over several data sources. Different approaches are being experienced through data warehouses (e.g. GUS [6]), federated databases (e.g. SEMEDA [14]) or mediators (e.g. TAMBIS [11]), all aiming at organizing access to several data sources in order to satisfy user queries. Systems such as TAMBIS or SEMEDA are capable of taking into account semantic processing of user query. However, most available systems only deal with a limited number of data sources that do not satisfy a large proportion of user queries. The work presented here aims at modeling knowledge about bioinformatic data sources in order to propose to users, given a query, the best-suited available bioinformatic data sources. The problem here is not the querying of the data sources themselves but rather the identification and selection among

F. Dau, M.-L. Mugnier, G. Stumme (Eds.): ICCS 2005, LNAI 3596, pp. 323–336, 2005.

all existing data sources of the most appropriate ones given the query. In this paper we propose a solution to a particular information retrieval (IR) problem where data sources instead of documents are searched and indexation is based on metadata reflecting information about sources rather than on data extracted from documents. Formal concept analysis (FCA) is used here for improving the retrieval of relevant data sources thanks to a dynamic and flexible classification of existing sources. In addition domain ontologies have been taken into account for processing the query in a semantic manner.

We will first review in section 2 related works that combine FCA and IR, as well as ontology usage in similar problems. Section 3 presents the BioRegistry project as a new repository for metadata about bioinformatic data sources. The formalisation of our problem using FCA is detailed in section 4 and the querying aspects are developed in section 5 including an original query refinement method. Finally some perspectives of this research work are discussed in section 6.

2 Related Work

2.1 Concept Lattices for Information Retrieval (IR)

The application of concept lattices in information retrieval was originally present at the beginning of FCA [24]. Indeed, an obvious analogy exists between object-attribute and document-term tables. Information retrieval was then mentioned as one application field for concept lattices usage [12]. The formal concepts in the lattice are seen as classes of relevant documents that match a given user query with the subsumption relation (i.e. the partial ordering relation within the concept lattice) between concepts allowing moving from one query to another more general or more specific. A lattice-based information retrieval approach is proposed in [4]. In both propositions [12] and [4], lattice-based information retrieval shows performances that are better than boolean information retrieval. One limitation is the complexity of the lattice (regarding the size and the needed computation) for large contexts. But in the real applications it is estimated that this maximum complexity is not reached [12]. However some works such as multi-level strategies developed in ZooM [19] and iceberg lattices [22] propose solutions for such complex applications either by expanding or refining a subpart of a lattice (ZooM) or by decreasing the overall size of the lattice by limiting the exploration depth of the set of concepts (icebergs).

2.2 Concept Lattice Construction

Several works deal with the problem of generating the set of concepts and the concept lattice of a given formal context. A detailed comparison between performances of algorithms for generating concept lattices and their diagram graphs can be found in [15]. Some of the proposed algorithms allow an incremental construction of concept lattice for a given formal context such as proposed in [13, 5, 23]. This aspect is particulary beneficial for the information retrieval

applications in general and for our bioinformatic problem in particular for two reasons. First, user queries need to be merged into the set of concepts in order to retrieve relevant documents (or bioinformatic data sources) included in these concepts as in [12, 4]. Second, incremental lattice construction allows the insertion of new concepts, that in our case takes into account the availability of new bioinformatic data sources on the web. This kind of insertion is essential for keeping the BioRegistry repository in accordance with the web content as explained in the following.

2.3 Improvement of FCA-based IR Performance Using Ontologies

Query refinement is an IR mechanism aiming at improving retrieval performance by adding to user's query new terms related to the query terms [2]. Propositions combining ontologies and FCA in the purpose of improving retrieval performance are found in [3, 20, 21]. In the two first works, a thesaurus is included to enhance the retrieval process by enriching the indexation in the lattice. In the last work, domain ontologies are used to build refined lattices according to user preferences thus avoiding complete lattice construction. Both approaches work directly on the lattice either by adding terms or by considering only parts of it.

In our work, domain ontologies are taken into account at early stage of the information retrieval process, i.e. during the BioRegistry construction (see below). This leads us to propose a mechanism for IR improvement based on query rather than lattice modification.

3 The BioRegistry Project

3.1 Bioinformatic Data Sources

Hundreds of biological data sources are known today [8]. Most efforts so far have been devoted to unifying the access to these sources, facilitating query processing and distribution over relevant sources, integrating answers, etc. These tasks involve designing appropriated workflows and require seamless interoperation of resources. Integrated systems are available that rely on data warehouses or mediation architectures. Today solutions are also envisaged in the context of semantic web, involving composition of web services [1, 26, 17].

The maximal efficiency of these solutions is reached when the whole knowledge available about all existing data sources can be exploited. For example the apparently simple query: *"What are the genes from human chromosome X that are preferentially expressed in brain?"* deals with both so-called *mapping data* and *expression data* which may or may not be contained in a single source at a given time. Probably more than one data source can be found for each part of the query. The user may select one of these sources because of given quality criteria (e.g. manual revision of the data or update frequency) or availability information (e.g. access constraints).

The largest existing catalog for bioinformatic data sources is certainly DBCAT[1] [7]. However, this flat file repository contains a rather small metadata set and offers limited query capabilities because most fields domains are open (free text). Registries are being developed for bioinformatic web services such as in the BioMoby[2] and MyGrid[3] projects [16]. Today the proportion of biological information accessible through web services is far too limited and does not properly answer users needs. However this situation may change and the need for modeling and organizing knowledge about web services in order to give access to relevant services for a given query will become as pressing as today for biological data sources. In order to build a specific environment for bioinformatic data source classification and searching and to test our propositions, we have decided to build our own registry called BioRegistry, in which the various metadata attached to biological data sources are organized in a dynamic, flexible and structured manner.

3.2 The BioRegistry Model

A hierarchical model has been designed to organize four categories of metadata attached to a data source: source identification, topics covered by the source, data and data source quality, availability. At present, all these metadata are manually extracted from the documentation associated to the data sources [18]. Topic information is divided in two parts: the subjects covered by the data sources and the organisms concerned. In both domains, existing controlled vocabularies, ontologies are used to valuate metadata fields by choosing the most specific terms and therefore minimizes the redundancy. The BioRegistry model thus includes a sub-hierarchy for describing and referencing the ontologies. To illustrate this point, figure 1 shows the ontology used to represent the phylogeny of model organisms in the purpose of indexing bioinformatic data sources. This ontology has been extracted from the NCBI taxonomy[4] that is used to index the Genbank sequences entries. Model organisms are lying at the leaves of the ontology and only the structuring nodes have been retained. Assuming that each node represents a concept defined by common properties shared by the corresponding group of organisms, the relation between nodes can be considered as a specialization (a partially ordering) relation. The MeSH thesaurus[5] has been used to valuate the subjects metadata field. The BioRegistry has been implemented as an XML schema compatible with semantic web languages such as OWL. Instances of the BioRegistry model organizing metadata relative to certain bioinformatic data sources can be visualized at the BioRegistry home page[6].

[1] http://www.infobiogen.fr/services/dbcat/
[2] http://mobycentral.cbr.nrc.ca/cgi-bin/gbrowse_moby
[3] http://mobycentral.cbr.nrc.ca/cgi-bin/gbrowse_moby
[4] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Taxonomy
[5] http://www.nlm.nih.gov/mesh/meshhome.html
[6] http://bioinfo.loria.fr/Members/devignes/Bioregistry/presentationBioregistry/view

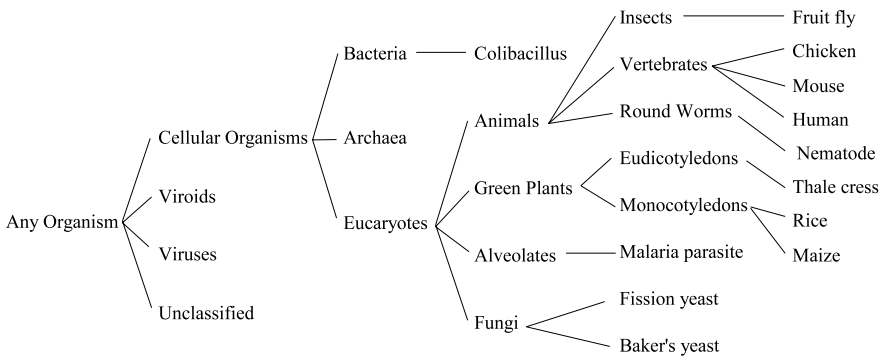

Fig. 1. Ontology of living organisms (defined for the BioRegistry)

3.3 BioRegistry Exploitation and FCA

First exploitation of the BioRegistry is form-based querying, allowing structured information retrieval of the metadata. This should allow the biologist to formulate a multi-criteria query combining various metadata categories and to recover a sorted list of data sources matching the query. For example the query cited in section 3.1 would be composed of the following criteria: subjects concerned = mapping or expression, organism concerned = human, manual revision = yes, update frequency = monthly, and access constraints = free.

However, this approach requires the user to formulate a query which may reveal inefficient without an overall knowledge on the data sources described in the BioRegistry. To overcome this limit, we decided to apply FCA to the BioRegistry content. Indeed this type of formalisation could enable flexible classification of data sources on the basis of metadata sharing as well as querying of the registry. The resulting classification of the sources present in the BioRegistry in the form of a concept lattice enables the user to discover relevant sources simply by browsing the lattice itself. Given a query, it is also possible to recover from the concept lattice data sources sharing all or a subset of metadata with the query. One advantage over classical information retrieval is that in the BioRegistry, the set of data sources is far smaller (about one thousand) than most sets of documents, thus constraining search space and query processing under certain limits. This can be considered as a condition for scalability.

Domain ontologies, used to valuate the BioRegistry metadata, are also intended to help the user to select query terms. In addition, they will be exploited as a mean for query refinement in order to improve the recall (see sections 5.2,5.3, 5.4 and 5.5).

4 Concept Lattices for Classifying BioRegistry Data Sources

4.1 Construction of BioRegistry Concept Lattice

In this section, we show how the FCA framework applies to the formalisation of BioRegistry content. More detailed FCA related definitions can be found in [10].

In the following, the formalisation of the BioRegistry is given by a formal context $\mathcal{K}_{bio} = (G, M, I)$ where G is a set of bioinformatic data sources (e.g. Swissprot, RefSeq,...), M is a set of metadata (e.g. manual revision, human organism,...) and I is a binary relation between G and M called the incidence of $\mathcal{K}_{bio}$ and verifying: $I \subseteq G \times M$ and $(g, m) \in I$ (or gIm) where g, m are such that $g \in G$ and $m \in M$ means that the data source g has the metadata m. An example of formal context is given in table 1 with bioinformatic data sources and metadata full names in table 2 (symbols and abbreviations are used for a better visibility in the lattice). Consider $A \in G$ a set of data sources, then the set of

Table 1. Example of BioRegistry formal context $\mathcal{K}_{bio}$.

Sources\Metadata	NS	PS	AS	AO	An	Ve	Hu	Mo	MR
S1	0	1	0	1	0	0	0	0	1
S2	1	1	1	1	0	0	0	0	1
S3	1	0	0	0	0	0	1	0	0
S4	0	1	0	1	0	0	0	0	1
S5	1	1	1	0	0	0	1	0	0
S6	1	0	0	0	1	0	0	0	0
S7	0	1	0	0	0	0	0	1	0
S8	0	1	0	0	0	1	0	0	0

Table 2. Complete names of bioinformatic data sources and their metadata.

Source name	Symbol
Swissprot	S1
RefSeq	S2
TIGR-HGI	S3
GPCRDB	S4
HUGE	S5
ENSEMBL	S6
Mouse Genome DB	S7
Vega Genome Browser	S8

Metadata (attributes)	Abbreviation	Category
Nucleic Sequences	NS	Subject
Proteic Sequences	PS	Subject
Any Sequence	AS	Subject
Any Organism	AO	Organism
Animals	An	Organism
Vertebrate	Ve	Organism
Human	Hu	Organism
Mouse	Mo	Organism
Manual Revision	MR	Quality

metadata common to all the sources in A is $A' = \{m \in M \mid \forall g \in A,\ gIm\}$. Dually for a set $B \in M$ of metadata, the set of data sources sharing all the metadata in B is $B' = \{g \in G \mid \forall m \in B,\ gIm\}$.

A formal concept in the BioRegistry formalisation $\mathcal{K}_{bio}$ is a data source set sharing a metadata set. It is formally presented by a pair (A, B) where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$; A and B are called the *extent* and the *intent* of the concept, respectively. We denote by $\mathcal{C}$ the set of all formal concepts of $\mathcal{K}_{bio}$. Consider $C_1 = (A_1, B_1)$ and $C_2 = (A_2, B_2)$ in $\mathcal{C}$. C_1 is subsumed by C_2 if $A_1 \subseteq A_2$ or dually $B_2 \subseteq B_1$ (denoted by $C_1 \sqsubseteq C_2$). $(\mathcal{C}, \sqsubseteq)$ is a complete lattice [25] called the concept lattice corresponding to the context $\mathcal{K}_{bio}$. In the following, $(\mathcal{C}, \sqsubseteq)$ will be denoted by $\mathcal{L}(\mathcal{C})$. Figure 2 shows the concept lattice $\mathcal{L}(\mathcal{C})$ corresponding to the BioRegistry formal context example $\mathcal{K}_{bio}$ given in table 1.

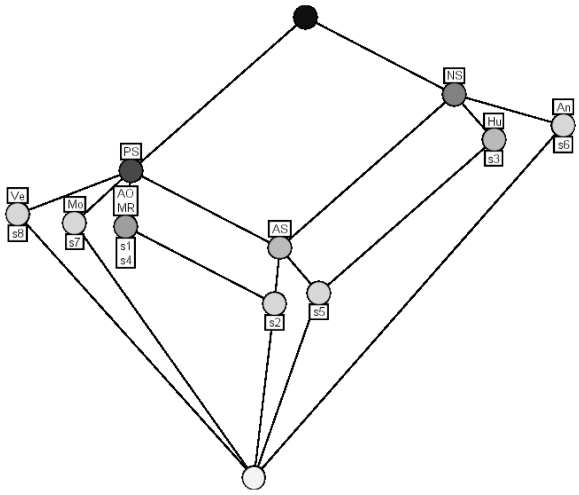

Fig. 2. The concept lattice $\mathcal{L}(\mathcal{C})$ corresponding to $\mathcal{K}_{bio}$

One important characteristic of the formal context $\mathcal{K}_{bio}$ is that the set M of metadata is carefully delineated during the BioRegistry construction so that its cardinality remains small. This particularity led us to choose the Godin algorithm [13] to generate the corresponding concept lattice since the context is small and sparse [15]. In addition, as mentioned in section 2.2 this algorithm allows the addition of new concepts to an existing lattice. This aspect is useful for the querying method described in section 5.

4.2 Flexible BioRegistry Data Sources Classification

Because various sets of data sources and/or various sets of metadata can easily be extracted from the BioRegistry (as a structured document), numerous possibilities can be offered to customize the views on the overall organization of bioinformatic data sources. For example, a user interested in the sharing of *subjects* across the data sources (see section 3.2 for a definition of *subject* in the

BioRegistry) may define a modified formal context where the attribute set is only composed of subject metadata. The object set of this modified formal context would be constituted by all the data sources being indexed with these metadata. Alternatively, a user may wish to visualize the classification of a subset of data sources dealing for instance with human data. A modified formal context may be constructed where the object set is a subset of the data sources retrieved from the BioRegistry on the basis of the metadata *organism* (see section 3.2) valued as *human*. The attribute set of this modified formal context would then be composed by the set of all metadata associated to the selected subset of data sources.

This flexibility in customizing the views over the BioRegistry content is for the moment very different from the solutions [19, 22] discussed above (section 2.1). It simply relies on a new automatic lattice construction every time that a new formal context can be created as an answer to a user need.

5 Querying BioRegistry Concept Lattices

5.1 Relevant Bioinformatic Data Sources Retrieval

Once the concept lattice $\mathcal{L}(\mathcal{C})$ is generated begins the retrieval of relevant sources. In the same way as in [12] and [4], we define a query as a formal concept $Q = (Q_A, Q_B)$ where $Q_A = \{Query\}$, i.e. a name for the extent to be formed (it can also be seen as a name for denoting an empty extent or a virtual class to be instantiated) and Q_B is the set of metadata to be used during the search. Actually, using the name $Query$ is an artifact for allowing the extent of the lattice by classifying the query $Q = (Q_A, Q_B)$. As an example consider a query that searches for data sources with the metadata *Nucleic Sequences*, *Human* and *Manual Revision*. Using the abbreviations given in table 2, the query is given by $Q = (\{Query\}, \{NS, Hu, MR\})$.

Once Q is given, it has to be classified in the concept lattice $\mathcal{L}(\mathcal{C})$ using the incremental classification algorithm of Godin et al. [13]. The resulting concept lattice is noted $(\mathcal{C} \oplus Q, \sqsubseteq)$ where $\mathcal{C} \oplus Q$ denotes the new set of concepts once the query has been added. In the following the concept lattice $(\mathcal{C} \oplus Q, \sqsubseteq)$ will be denoted by $\mathcal{L}(\mathcal{C} \oplus Q)$. For the given example the modified concept lattice $\mathcal{L}(\mathcal{C} \oplus Q)$ is shown on figure 3. Dashed circles point out new or modified concepts due to the insertion of the query. Only these concepts share properties with the query and could thus be interesting for the user.

The query concept is denoted by Q either in $\mathcal{L}(\mathcal{C})$ or in $\mathcal{L}(\mathcal{C} \oplus Q)$. If there exists in the lattice $\mathcal{L}(\mathcal{C})$ a concept of the form $(A, Q_B \cup B)$, then the classification of Q in $\mathcal{L}(\mathcal{C})$ will produce a subsumer concept of the form $(\{Query, A\}, Q_B)$ that will be the new query concept to be considered. For the sake of simplicity, we continue to denote by Q the query concept in $\mathcal{L}(\mathcal{C} \oplus Q)$ whatever the case.

Definition 1. *A data source is relevant for a given query if and only if it shares at least one metadata mentioned in the query. The degree of relevance is given by the number of metadata shared with the query and by the stage during which the data source is added to the result.*

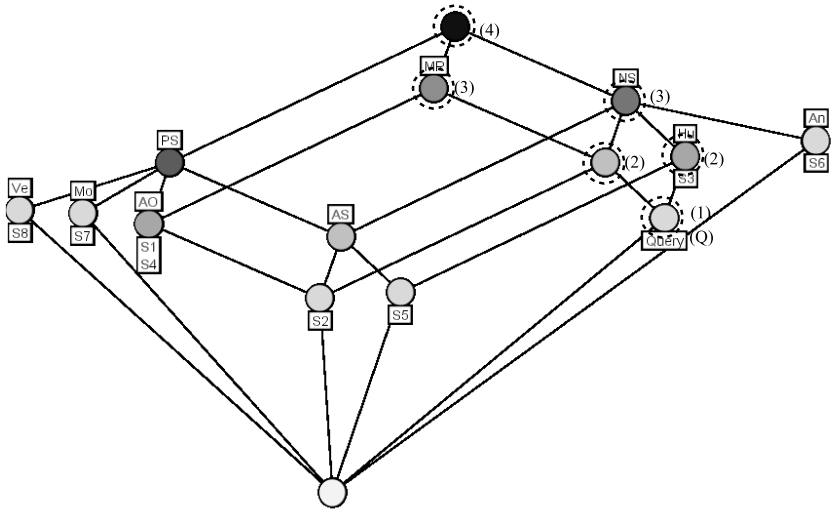

Fig. 3. The concept lattice $\mathcal{L}(\mathcal{C} \oplus Q)$

This definition of relevance is the basis of the retrieval process in the lattice and differs from the neighborhood notion used in [4]. The latter can lead to retrieved documents lacking any query term which is acceptable in document retrieval but not suitable to our needs. The above definition of relevance is sufficient to explain the retrieval algorithm detailed hereafter.

Considering the above definition, all the relevant sources are in the extents of Q and its subsumers in the concept lattice (indicated by dashed circles other than the top concept in figure 3) since the intent of each one of these concepts is a subset of Q_B (the intent of the query concept). In the following we will denote by $\mathcal{R}_{sources}$ the set of relevant data sources for the considered query. It is important to mention here that all the sources in $\mathcal{R}_{sources}$ do not have the same relevance. In fact, they are ranked according to the number of shared metadata with the query and according to the stage during which they have been added to $\mathcal{R}_{sources}$.

Intuitively, the relevant data source retrieval algorithm consists first in classifying the query concept in the lattice, operation that instantiates the extension $\{Query\}$ (actually, $Query$ could be considered as a variable to be instantiated). Then, the set of data sources that are inherited from the subsumers of the query concept Q in the lattice are gathered in the result $\mathcal{R}_{sources}$. The rank of the returned data sources may be memorized according to the distance of the sources to the query concept. Consider $\mathcal{C}_1$ the direct (most specific) subsumers of Q in the concept lattice. The set of data sources in the extents of the concepts in $\mathcal{C}_1$ and not already in the result are added to the result. The next step consists in considering the direct subsumers of concepts in $\mathcal{C}_1$ (subsumers of distance two of Q) and adding new emerging data sources to the result set $\mathcal{R}_{sources}$. Then we

continue in the same way for $\mathcal{C}_2$, $\mathcal{C}_3$ etc until we reach an empty set $\mathcal{C}_n$. In each step, the data sources in the extent of concept with an empty intent are ignored since they may not share metadata with the query.

In figure 3, the numbers near the concepts show the iterations of the explained algorithm. In the first iteration, Q is considered. In this case there is no data source in the extent of the query Q. In the second iteration the data sources S_3, S_5 and S_2 are added to the result. In the third iteration the data sources S_6, S_1 and S_4 are added to the result.

Finally the set $\mathcal{R}_{sources}$ includes the data sources ranked as follows:

1. S3 (TIGR-HGI) and S5 (HUGE) share *Nucleic Sequences* and *Human* with Q

1. S2 (RefSeq) shares *Nucleic Sequences* and *Manual Revision* with Q

2. S6 (ENSEMBL) shares *Nucleic Sequences* with Q

2. S1 (Swissprot) and S4 (GPCRDB) share *Manual Revision* with Q

Additional ranking criteria can be defined according to given preferences (e.g. user preferences).

5.2 Ontology-Based Query Refinement

The query concept may not be filled with any result. For example in the BioRegistry formal context presented in table 1 a user searching data sources containing data relative to the organism *Chicken* will not get any answer. However, there may be data sources relevant to the query described by metadata that do not directly map the query metadata. To help further the user we propose a query refinement procedure based on domain ontologies.

Contrasting with the propositions [3, 20, 21] mentioned in section 2.3, we modify the query instead of the lattice. In fact, we preserve the whole lattice structure and we modify the query by inserting metadata related to metadata of the query in a given ontology. This strategy, which can be automated, avoids introducing redundancy in the lattice.

Added metadata are either more specific or more general than those initially in the query. This leads to two types of query refinement: refinement by generalisation and refinement by specialisation. It is important to recall here that we are not facing any synonymy problem between metadata in the query and terms in the ontology since metadata valuation in the BioRegistry involves terms extracted from domain ontologies.

The generalisation refinement w.r.t. a metadata adds more general metadata represented by its ancestors in the ontology. In the example cited above (metadata *Chicken*), considering the ontology shown in figure 1, the metadata that can be added to the query are *Vertebrates, Animals, Eucaryotes, Cellular Organisms*, and *Any Organism*. However some of these metadata (*Eucaryotes* and *Cellular Organisms*) are not in the formal context $\mathcal{K}_{bio}$ given in the table 1. This means that these metadata are not shared by any source in this context so adding them to the query will not lead to any result enrichment. Only new metadata already present in $\mathcal{K}_{bio}$ are considered during the generalisation refinement process.

In a dual way the specialisation refinement w.r.t. a metadata adds semantically more specific metadata represented by its descendants in the ontology. In the given example the metadata *Chicken* has no descendant and thus could not be specialised. A better example would be a query composed by the metadata *Eucaryotes* which does not retrieve any answer since this metadata is absent from the formal context $\mathcal{K}_{bio}$. Specialisation refinement leads to inspect all descendants of *Eucaryotes* in the ontology and select only those that appear in the formal context (*Animals, Vertebrate, Human*, and *Mouse*) to add them to the query.

It is possible to combine both types of query refinement. This means, for a given query metadata, adding both its ancestors and its descendants in the corresponding domain ontology. In all cases of query refinement the number of added metadata can be controlled by considering only the nearest ancestors to the considered metadata in the ontology (generalisation refinement) or its nearest descendants (specialisation refinement).

Once the ontology-based query refinement done, the refined query has to be inserted into the original lattice $\mathcal{L}(\mathcal{C})$ and the algorithm detailed above can be applied to the new resulting lattice $\mathcal{L}(\mathcal{C} \oplus Q)$. The next section presents the ontology-based query refinement.

5.3 The Generalisation Query Refinement

Consider the query with the metadata *Chicken* represented by the formal concept $Q = (\{Query\}, \{Ch\})$. The result for this query is empty since the metadata it contains is not in the context. Applying the generalisation query refinement, we obtain the following result as response to the refined query:

1. S6 (ENSEMBL) shares *Animals* with the refined query
1. S8 (Vega Genome Browser) shares *Vertebrate* with the refined query
1. S1 (Swissprot), S2 (RefSeq) and S4 (GPCRDB) share *Any Organism* with the refined query

Each source of the result has a part satisfying the query and a part that does not (e.g. S8 is concerned with *Chicken* but with *Mouse* and *Human* as well). Furthermore the shorter the distance between the query metadata and the added metadata the more relevant the resulting sources (S8 is preferable to S6). This aspect motivates the possibility of controlling the added metadata during the generalisation refinement process mentioned above. Hence to avoid introducing less relevant (or irrelevant) sources in the result we have to consider only the nearest ancestors of the considered metadata in the domain ontology.

5.4 The Specialisation Query Refinement

Consider the query with the metadata *Eucaryotes* represented by the formal concept $Q = (\{Query\}, \{Eu\})$. The result for this query is empty since the metadata it contains is not in the context. Applying the specialisation query refinement, we obtain the following result for the refined query:

1. S6 (ENSEMBL) shares *Animals* with the refined query
1. S8 (Vega Genome Browser) shares *Vertebrate* with the refined query
1. S5 (ENSEMBL) shares *Human* with the refined query
1. S7 (Mouse Genome DB) shares *Mouse* with the refined query

In this case each source of the result gives a partial answer to the query and a composition of these data sources could provide a complete answer to the query if each descendant (of the query metadata) indexes one data source. Similarly as in the generalisation refinement the distance between the original metadata and the added ones explains the difference of sources relevance. In fact sources dealing with a far descendant of the query metadata give precise information that is not always needed by the user. The level of specialisation can be controlled by considering the nearest descendants of the metadata in the domain ontology that constitute the best coverage of the query.

5.5 Choice Between Generalisation and Specialisation Query Refinement

When the considered metadata is a leaf or is the root of the domain ontology there is no problem of choice since in both case only one type of refinement is possible (generalisation in the first case and specialisation in the second). But when the metadata is neither a leaf nor the root the two types of refinement are possible. The choice can be done with relation to the user preferences. In fact if the user accepts to get data sources a part of which corresponds to his need then the generalisation refinement is adopted. If he accepts to get data sources that correspond to a part of his need the specialisation refinement is used. In both cases it is useful to have a post ranking of the new selected data sources reflecting the similarity between their indexing metadata and the query one [9].

6 Conclusion and Future Work

In this paper, we have presented an approach combining formal concept analysis and domain ontologies for an information retrieval problem in bioinformatics. The BioRegistry as a structured repository of metadata relative to bioinformatic data sources (including data quality information) constitutes a well-suited application domain for the FCA theory allowing scalability and flexibility. The approach is intended for the problem of relevant bioinformatic data sources selection for a further interrogation. Indeed concept lattices appear as a mean to provide customised views about bioinformatic data sources and to organize knowledge about these sources. This in turn can help the user in the process of data sources retrieval to answer his query. Furthermore ontology-based query refinement mechanisms have been proposed to improve this retrieval process.

An implementation of our proposition is currently underway. Preliminary testing have shown the usefulness of a post-processing mechanism to improve the ranking of retrieved data sources.

Acknowledgments

We would like to thank Shazia Osman for her contribution to the conception of the BioRegistry model, its construction and indexation of the data sources it contains. This work was supported by the "PRST Intelligence Logicielle" from the Région Lorraine.

References

[1] D. Buttler, M. Coleman, T. Critchlow, R. Fileto, W. Han, C. Pu, D. Rocco, and L. Xiong. Querying Multiple Bioinformatics Information Sources: Can Semantic Web Research Help? *SIGMOD Record*, 31(4):59–64, December 2002.

[2] D. Carmel, E. Farchi, Y. Petruschka, and A. Soffer. Automatic query refinement using lexical affinities with maximal information gain. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 283–290. ACM Press, August 2002.

[3] C. Carpineto and G. Romano. A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, 24(2):95–122, August 1996.

[4] C. Carpineto and G. Romano. Order-theoretical ranking. *Journal of the American Society for Information Science*, 51(7):587–601, May 2000.

[5] C. Carpineto and G. Romano. *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, 2004.

[6] S. B. Davidson, J. Crabtree, B. P. Brunk, J. Schug, V. Tannen, G. C. Overton, and C. J. Stoeckert. K2/Kleisli and GUS : experiments in integrated access to genomic data sources. *IBM systems journal*, 40(2):512–531, 2001.

[7] C. Discala, X. Benigni, E. Barillot, and G. Vaysseix. DBCAT: a catalog of 500 biological databases. *Nucleic Acids Research*, 28(1):8–9, January 2000.

[8] M. Y. Galperin. The Molecular Biology Database Collection: 2004 update. *Nucleic Acids Research*, 32:D4–D22, 2004.

[9] P. Ganesan, H. Garcia-Molina, and J. Widom. Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems (TOIS)*, 21(1):64–93, January 2003.

[10] B. Ganter and R. Wille. *Formal Concept Analysis*. Mathematical Foundations, Springer-Verlag, 1999.

[11] C. A. Goble, R. Stevens, G. Ng, S. Bechhofer, N. W. Paton, P. G. Baker, M. Peim, and A. Brass. Transparent Access to Multiple Bioinformatics Information Sources. *IBM Systems Journal*, 40(2):532–551, 2001.

[12] R. Godin, G. W. Mineau, and R. Missaoui. Méthodes de classification conceptuelle basées sur les treillis de Galois et applications. *Revue d'intelligence artificielle*, 9(2):105–137, 1995.

[13] R. Godin, R. Missaoui, and H. Alaoui. Incremental Concept Formation Algorithms Based on Galois (Concept) Lattices. *Computational Intelligence*, 11:246–267, 1995.

[14] J. Kohler, S. Philippi, and M. Lange. SEMEDA : ontology based semantic integration of biological databases. *Bioinformatics*, 19(18):2420–2427, December 2003.

[15] S. O. Kuznetsov and S. A. Obiedkov. Comparing Performance of Algorithms for Generating Concept Lattices. *Journal of Experimental & Theoretical Artificial Intelligence*, 14:189–216, 2002.

[16] P. Lord, S. Bechhofer, M. D. Wilkinson, G. Schiltz, D. Gessler, D. Hull, C. Goble, and L. Stein. Applying semantic web services to Bioinformatics: Experiences gained, lessons learnt. In F. v. H. Sheila A. McIlraith, Dimitris Plexousakis, editor, *The Semantic Web ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings*, volume 3298, pages 350–364. Springer-Verlag GmbH, 2004.

[17] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Greenwood, T. Carver, Matthew, Pocock, A. Wipat, and P. Li. Taverna : a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20:3045–3054, 2004.

[18] S. Osman. Réalisation d'un annuaire de sources de données génomiques en vue de la collecte et de l'intégration de données sur le web. Rapport de master professionnel sciences et techniques mention informatique, spécialité bio-informatique, Université Bordeaux I, Université Victor Segalen, Bordeaux II, Septembre 2004.

[19] N. Pernelle, M.-C. Rousset, H. Soldano, and V. Ventos. ZooM: a nested Galois lattices-based system for conceptual clustering. *Journal of Experimental and Theoretical Artifial Intelligence (JETAI)*, 14(2):157–187, September 2002.

[20] U. Priss. Lattice-based Information Retrieval. *Knowledge Organization*, 27(3):132–142, 2000.

[21] B. Safar, H. Kefi, and C. Reynaud. OntoRefiner, a user query refinement interface usable for Semantic Web Portals. In *Proceedings of Application of Semantic Web technologies to Web Communities, Workshop ECAI'04*, pages 65–79, Valencia, Spain, August 2004.

[22] G. Stumme, R. Taouil, Y. Bastide, and L. Lakhal. Conceptual Clustering with Iceberg Concept Lattices. In *Proceeding GI-Fachgruppentreffen Maschinelles Lernen (FGML'01)*, Universitat Dortmund 763, Oktober 2001.

[23] D. van der Merwe, S. A. Obiedkov, and D. G. Kourie. AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In P. W. Eklund, editor, *ICFCA Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings*, volume 2961, pages 372–385. Springer, 2004.

[24] R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. *Ordered sets*, pages 445–470, 1982.

[25] R. Wille. Line diagrams of hierarchical concept systems. *International Classification*, 2:77–86, 1984.

[26] C. Wroe, R. Stevens, C. Goble, A. Roberts, and M. Greenwood. A suite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data. *International Journal of Cooperative Information Systems*, 12(2):197–224, March 2003.