Mining Linked Open Data: A Case Study with Genes Responsible for Intellectual Disability

Gabin Personeni, Simon Daget, Céline Bonnet, Philippe Jonveaux, Marie-Dominique Devignes, Malika Smaïl-Tabbone, Adrien Coulet

Mining Linked Open Data: a Case Study with Genes Responsible for Intellectual Disability

Gabin Personeni[1], Simon Daget[1], Céline Bonnet[2,3], Philippe Jonveaux[2,3], Marie-Dominique Devignes[1], Malika Smaïl-Tabbone[1], and Adrien Coulet[1]

[1] LORIA (CNRS, Inria NGE, Université de Lorraine), Campus scientifique, Vandoeuvre-lès-Nancy, F-54506, France
[2] Laboratoire de Génétique Médicale, Centre Hospitalier Universitaire de Nancy, Vandoeuvre-lès-Nancy, France
[3] INSERM U-954, Université de Lorraine, Rue du Morvan, Vandoeuvre-lès-Nancy, France

Abstract. Linked Open Data (LOD) constitute a unique dataset that is in a standard format, partially integrated, and facilitates connections with domain knowledge represented within semantic web ontologies. Increasing amounts of biomedical data provided as LOD consequently offer novel opportunities for knowledge discovery in biomedicine. However, most data mining methods are neither adapted to LOD format, nor adapted to consider domain knowledge. We propose in this paper an approach for selecting, integrating, and mining LOD with the goal of discovering genes responsible for a disease. Selection step relies on a set of choices made by a domain expert to isolate relevant pieces of LOD. Because these pieces are potentially not linked, an integration step is required to connect unlinked pieces. Resulting graph is subsequently mined using Inductive Logic Programming (ILP) that presents two main advantages. First, the input format compliant with ILP is close to the format of LOD. Second, domain knowledge can be added to this input and be considered by ILP. We have implemented and applied this approach to the characterisation of genes responsible for intellectual disability. On the basis of this real world use case, we present an evaluation of our mining approach and discuss its advantages and drawbacks for the mining of biomedical LOD.

1 Introduction

Linked Open Data (LOD) is part of a community effort to build a semantic web, where web resources can be interpreted both by humans and machines. LOD is a large and growing collection of datasets represented in a standard format (that includes the use of RDF and URI*s*), partially connected to each others and to domain knowledge represented within semantic web ontologies [1]. For these reasons, LOD offers novel opportunities for the development of successful data integration and knowledge discovery approaches.

It can be particularly beneficial to the life sciences, where relevant data are spread over various data resources with no agreement on a unique representation of biological entities [2]. Consequently, data integration is an initial challenge one faces if he wants to mine life science data considering several data sources. Various initiatives such as Bio2RDF, LOD drug data, PDBj or the EBI platform aim at pushing life sciences data into the LOD with the idea of facilitating their integration [3, 4, 5, 6]. It results from these initiatives a large collection of life-science data unequally connected but in a standard format and free of use for mining. In addition to their integrated dimension, LOD may be connected to domain knowledge represented within ontologies such as the Gene Ontology [7]. Ontologies provide a formal representation of a particular domain that can be used to support automatic reasoning. We have investigated that ontologies and their associated reasoning mechanisms can be coupled with data mining to facilitate the process of knowledge discovery [8, 9]. We would like to extend this investigation to the context of LOD. Despite good will and emerging standard practices for publishing data as LOD, several drawbacks make their use still challenging [10, 11]. Among existing difficulties we can list the limited amount of links between datasets, the lack of update on published datasets, the unequal rationales of systems that enable querying LOD.

We propose here an approach that is schematized in Figure 1 and that enables to *(1)* select, *(2)* integrate and then *(3)* mine LOD with Inductive Logic Programming (ILP). *(1)* The selection of LOD is achieved with respect to a conceptualization of data related to a biomedical question. This conceptualization is driven by a biomedical expert and, in this paper, is motivated by our will to characterize genes responsible for intellectual disability. *(2)* The integration of LOD is made possible both by the use of existing links from the LOD and by the manual definition of mappings between our conceptualisation and LOD. Links and mappings enable to automatically build SPARQL queries and subsequently to retrieve, from the LOD, triples to mine. In addition, our mappings enable the generation of unpublished links between LOD entities, consequently contributing to the community effort. *(3)* Finally, triples are mined using ILP that is particularly adapted to the format of LOD and capable of taking into account domain knowledge defined in ontologies.

The next section presents a state of the art of data mining applied to LOD and then presents ILP. The third section presents the LOD selection and integration made in preparation for the mining. The fourth section reports about mining experiments with ILP on selected triples. Last section concludes on experiments and presents perspectives of this initial work.

2 State of the Art

2.1 Preparing LOD for Mining

The complexity of LOD has motivated several studies about the preparation (*i.e.,* selection, integration, formatting) of data before mining. For instance, we proposed a system that guides the selection of LOD by structuring data within

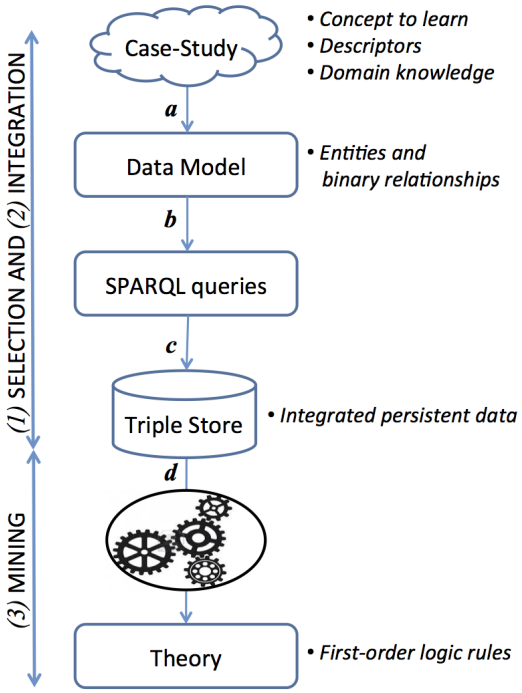

Fig. 1. Outline of the methodology used for preparing and mining Linked Open Data (LOD). *a*: Conceptualization in term of entities and binary relationships; *b*: Mapping onto various LOD datasets; *c*: Retrieval of triples using SPARQL queries; *d*: Relational learning with Inductive Logic Programming (ILP).

a lattice that provides insight about which type of entities are related and how [12]. Callahan *et al.* proposed to map LOD from various datasets to an upper-level ontology named SIO. This ontology serves consequently as a global schema and its terms are used to write federated queries over LOD datasets [13]. SADI is a general framework to facilitate the discovery and use of web services [14]. Because it has been developed with semantic web technologies, SADI is well adapted to define pipelines that can query SPARQL endpoints and integrate their results. The COEUS platform follows a similar rationale but includes a federation layer that facilitates data integration [15].

Any of these solutions are well adapted when either entities have a unique URI over distinct datasets, or when links have been defined between datasets. Unfortunately, these two prerequisites are not guaranteed in LOD. In this work, we want to be able to use any dataset of the LOD, even if this requires to define novel mappings between datasets, using various relationship types. For this reason we propose a simple but generic way for selecting and integrating LOD to be mined.

2.2 Mining LOD

The emergence of several workshops about the mining of LOD illustrates the gain of interest for this topic, both in the semantic web and data mining communities [16, 17].

A first type of contribution in this domain aims at completing or correcting the LOD. In that vein, Gangemi *et al.* proposed an approach to type automatically DBpedia entities using graph patterns and disambiguation techniques [18]. Other authors studied how to propose automatically missing links, particularly between unrelated datasets [19, 20]. For example, Brenninkmeijer *et al.* developed a tool for proposing `owl:sameAs` links between unrelated drugs of LOD datasets [21].

A second group of works explores how some particularities of LOD can help data mining. For example, Percha *et al.* used paths between distinct drugs in linked data to predict novel drug-drug interactions [22]. Here, the fact that relationships and entities are typed in LOD enabled to define features that characterise possible paths between drugs and consequently to train a random forest classifier. Pathak *et al.* proposed a study on how federated queries over Electronic Health Records and drug related LOD could enable the discovery of novel drug-drug interactions [23].

To our knowledge, only few seminal works have explored how LOD mining can take advantages of knowledge representation [24, 25]. In this work we propose to explore this direction using ILP.

2.3 Inductive Logic Programming

ILP Principles. Inductive Logic Programming (ILP) allows to learn a concept definition from observations, *i.e.,* a set of positive examples ($E+$) and a set of negative examples ($E-$), and background knowledge (B) [26]. Given $E+$, $E-$, and B the goal of concept learning by ILP is to induce a set of rules or a theory T that is consistent ($T \cup B$ covers or explains each positive example in $E+$), and complete ($T \cup B$ does not cover or contradicts any negative example in $E-$).

In most ILP systems both B and T are represented as definite clauses (or prolog programs) in First-Order Logic (FOL), *i.e.,* a disjunction of literals with one positive literal. A rule has the form "head `:-` body" and is interpreted as: if the conditions in the body are true then the head is true as a logical consequence. The background knowledge B includes *(i)* the relational description of the examples using a set of relevant n-ary predicates such as

`protein_pathway`('insulin', 'insulin signaling pathway')

and *(ii)* a priori domain knowledge, *i.e.,* a set of rules and facts which don't refer to any example but express what is known about the elements which describe the examples, for instance

`subClass`('insulin receptor binding', 'receptor binding').

The theory T is a set of rules which cover as many of the positive examples as possible and the fewest negative examples. The head of each rule is the concept to learn whereas the body contains the induced description of the concept (based on a generalization of examples). An example of rule when studying gene responsible for a disease will have the form

`is_responsible(X) :- gene_protein(X,Y), protein_mf(Y,`'receptor binding'`).`

The rule search is performed in a clause space where the clause subsumption allows building generalizations or specializations of the clauses [27]. As the clause space is too large to be exhaustively explored, heuristic mechanisms exist to reduce its size. These mechanisms called learning biases allow the user to define which kind of rules (s)he wants to get by setting some parameters that influence the rule search strategy.

The Aleph Program. The experiments reported in this paper were conducted with the Aleph program whose basic algorithm is described in four steps [28]:

- Select a seed example to be generalized. If none exists, stop.
- Construct the most specific clause that entails the example selected, and is compliant with the language restrictions provided. This is usually a definite clause with many literals. It is called the"bottom clause".
- Find a clause more general than the bottom clause. This is done by searching for some subset of the literals in the bottom clause that has the "best" evaluation score.
- The clause with the best score is added as a rule to the current theory, and all examples made redundant are removed. Return to Step 1.

Many parameters can be set for tuning some aspect of the theory construction with Aleph. For instance, the rule evaluation function can be chosen and the default one is based on the difference between the number of covered positive examples and the number of covered negative examples. The noise parameter is the maximum negative examples that an acceptable rule may cover (default value is 0). This parameter can be set to higher values in case of noisy data (in our study, one is never sure that a gene is not responsible for a disease). The min-pos parameter is the minimal number of positive examples that a rule must cover (default value is 1). Aleph also requires other learning biases to be defined as *(i)* a set of determinations defining the predicate to learn and the predicates which can appear in the rules; *(ii)* a set of modes defining the types of predicate arguments and the way they can be chained in a rule.

As the algorithm suggests, Aleph iterates on the positive examples of the learning set for building the most specific clause of a chosen seed example which is compliant with the defined bias and covers the maximum number of positive examples and the minimum number of negative ones. When finding the best rule, the examples covered by the best rule are removed from the seed set and from the learning set (used for rule evaluation).

3 LOD Selection and Integration

3.1 Conceptualization

In our approach, the first step is to build an entity-relationship (ER) model decribing the entities to consider for a given study. The goal of the ER model is to provide an abstract model of data that are relevant to mine. This step is realized with an expert of the domain, and does not require any knowledge of what data is available in LOD and how it is structured. An ER model consists in a conceptualization usually made of entities, relationships and attributes. We use only a subset of those: entities and binary relationships without attributes (similarly to RDF properties). In our case, n-ary relationships and relationships with attributes are represented with a composition of binary relationships using the reification mechanism. Figure 2 presents the ER model defined for our study of genes responsible for Intellectual Disability (ID).

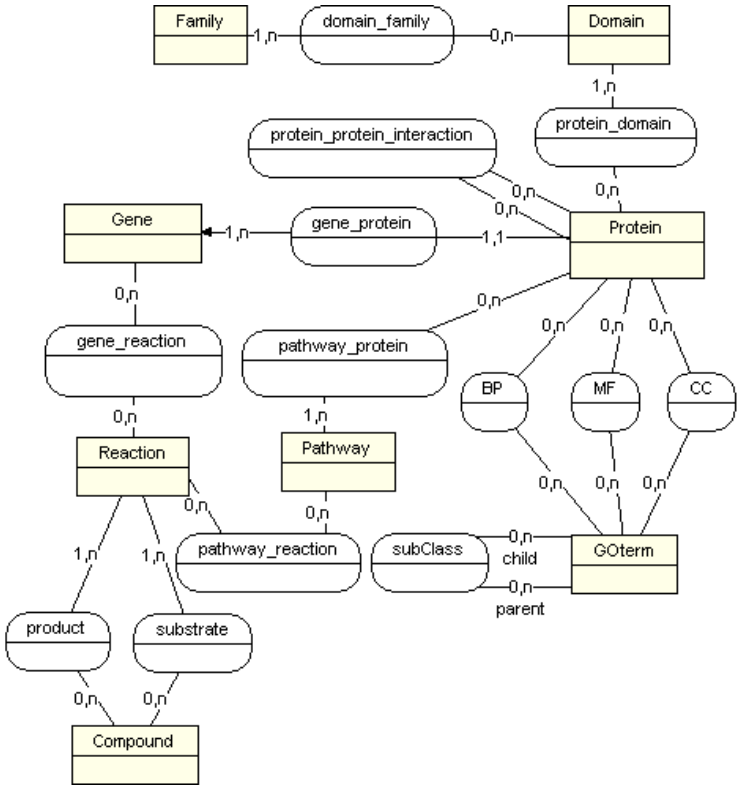

Fig. 2. ER model of data on genes responsible for Intellectual Disability (ID). For the sake of clarity, we have not represented gene location, which is composed of chromosome, arm, region, band, sub-band and sub-sub-band.

Table 1. Sources and count of distinct collected individuals that instantiate each entity of our ER model

Entity	SPARQL endpoint	#Individuals
`Gene`	`cu.gene.bio2rdf.org/sparql` `cu.kegg.bio2rdf.org/sparql`	549
`Protein`	`beta.sparql.uniprot.org/sparql`	1257
`Pathway`	`cu.kegg.bio2rdf.org/sparql` `www.ebi.ac.uk/rdf/services/reactome/sparql`	580
`Reaction`	`cu.kegg.bio2rdf.org/sparql`	433
`Compound`	`cu.kegg.bio2rdf.org/sparql`	628
`GOterm`	`cu.goa.bio2rdf.org/sparql` `sparql.bioontology.org/sparql`	7770
`Domain`	`cu.interpro.bio2rdf.org/sparql`	262
`Family`	`cu.interpro.bio2rdf.org/sparql`	781
	Total	12260

3.2 Mapping the ER Model onto LOD and Individual Identification

Mapping Definition. LOD integration consists primarily in mapping our expert-defined ER model onto LOD types of entities and of relationships. This mapping is materialized by defining correspondances from each entity of the model to one or many RDF entity types of LOD; and from each relationship to RDF properties. Indeed, distinct LOD datasets may use distinct entity types to refer to a single entity of our model. For instance, the entity `Gene` of the model is mapped to two entity types: `<http://bio2rdf.org/geneid:vocabulary:Gene>` and `<http://bio2rdf.org/kegg_vocabulary:Gene>` respectively used in two datasets of Bio2RDF: NCBI Gene and KEGG. Each entity is mapped to a *concept definition* that can either be a RDF entity type, the domain or range of a property, or an union, intersection or negation of another concept definition. Similarly, the relationships of the ER model can be mapped to one property or a composition of properties (or inverse properties), or to an artificial property subsuming them. For instance, the relationship `gene_reaction` between a gene and a reaction (which represents the fact that the gene produces an enzyme that catalyzes the reaction) can be mapped to $\texttt{kegg:xGene}^{-} \circ \texttt{kegg:xEnzyme}^{-}$ [1]. Table 1 and Table 2 list entities and relationships of our ER model and the datasets they are mapped to.

Individual Identification. Because the mapping can associate one entity with two datasets, it can cause redundancy. To guarantee the consistency of data related by our mapping, we need additional information on individual identity.

[1] The property `kegg:xGene` relates genes to enzymes, and `kegg:xEnzyme` relates enzymes to reactions. Since `gene_reaction` relates reactions to genes, we need to use their inverse properties denoted by $^{-}$. The symbol $\circ$ denotes the composition of properties.

Table 2. Sources and count of distinct collected instances for each relationship of our ER model

Relationship	SPARQL endpoint	#Individuals
gene_protein	beta.sparql.uniprot.org/sparql	819
gene_reaction	cu.kegg.bio2rdf.org/sparql	500
pp_interaction	cu.irefindex.bio2rdf.org/sparql	742
pathway_protein	www.ebi.ac.uk/rdf/services/reactome/sparql	767
protein_domain	cu.interpro.bio2rdf.org/sparql	262
pathway_reaction	cu.interpro.bio2rdf.org/sparql	706
substrate	cu.kegg.bio2rdf.org/sparql	938
product	cu.kegg.bio2rdf.org/sparql	960
protein_bp	cu.goa.bio2rdf.org/sparql	10242
protein_cc	cu.goa.bio2rdf.org/sparql	4358
protein_mf	cu.goa.bio2rdf.org/sparql	4063
subClass	sparql.bioontology.org/sparql	12779
domain_family	cu.interpro.bio2rdf.org/sparql	1238
gene_chromosome	cu.gene.bio2rdf.org/sparql	538
gene_chromosome_arm	cu.gene.bio2rdf.org/sparql	538
gene_chromosome_region	cu.gene.bio2rdf.org/sparql	538
gene_chromosome_band	cu.gene.bio2rdf.org/sparql	538
gene_chromosome_subband	cu.gene.bio2rdf.org/sparql	311
gene_chromosome_subsubband	cu.gene.bio2rdf.org/sparql	63
	Total	40900

Individuals are identified in LOD by their URI*s*. The main issue in mining LOD from several datasets is that two distinct URI*s* from different LOD datasets may refer to the same real world object. Individuals' URI*s* links from one LOD dataset to another may be available, ideally using the property `owl:sameAs`, although sometimes a less precise link, such as `rdfs:seeAlso` or a dataset dependent predicate is used. For entity types that map to concepts from several LOD datasets, an automatic way of resolving identity of individuals needs to be established. This can be achieved through several means, such as:

- Using when available in LOD, links that express equivalence between alternative URI*s* of an individual.
- Using data from LOD associated with individuals to assess the identity:
  - URI*s* itselves sometimes embed enough data to assess that two individuals are identical. For example, in some datasets, gene URI*s* contain the NCBI Gene ID: the human gene with Gene ID 5091 is represented by the URI `<http://bio2rdf.org/geneid:5091>` in Bio2RDF NCBI Gene, and `<http://bio2rdf.org/kegg_vocabulary:hsa:5091>` in Bio2RDF KEGG. An obvious link between the two URI*s* can be made on the basis of the Gene ID.
  - Individuals can be associated with literals that identify them across datasets, such as the HGNC gene symbol for genes.

Using these methods, given a URI of a given dataset, we can find the corresponding URI in another dataset. Links we generated this way are available online at `http://www.loria.fr/~coulet/dils14/individual_identities.html`.

3.3 Triple Retrieval and Storage

For the purpose of ILP mining, a set of positive examples and a set of negative examples must be provided. In our study, positives examples are genes responsible for ID, while negative examples are genes that are not responsible for it. Positive examples are genes from a state of the art study about genes responsible for ID by Inlow and Restifo [29]. We selected negative examples among genes responsible for diseases other than ID. To this aim, we first selected phenotypes in OMIM which do not contain ID as a symptom. From this large set of phenotypes, biomedical experts advised the selection of a subset of phenotypes clearly distinct from ID. Genes responsible for these phenotypes, were then retrieved from OMIM. The final set of negative examples is selected from stratified sampling with respect to the overall number of genes associated with each phenotype.

Given the ER model and its mapping to LOD entity types and roles, SPARQL queries can be built in a systematic way to retrieve the data from LOD. As an illustration, for building a SPARQL query to retrieve the families of protein domains (`domain_family` relationship in Table 2) the following mapping was used: the `Domain` entity is mapped to the entity type `<http://bio2rdf.org/interpro_vocabulary:Domain>`; `Family` is mapped to `<http://bio2rdf.org/interpro_vocabulary:Family>`; and the `domain_family` relationship is mapped to the property `<http://bio2rdf.org/interpro_vocabulary:contains<sup>-</sup>>`. On this basis, the following query is built:

```
SELECT ?x ?y
WHERE {
   ?x a <http://bio2rdf.org/interpro_vocabulary:Domain>.
   ?y a <http://bio2rdf.org/interpro_vocabulary:Family>.
   ?y <http://bio2rdf.org/interpro_vocabulary:contains> ?x.
   FILTER(?x = ...)
}
```

The `FILTER` statement of the query is used to retrieve only triples associated with genes reponsible/not responsible for ID.

SPARQL queries are generated and executed, then retrieved data is automatically stored in a triple store. Our triple store relies on a simple relational databse built upon the ER model. To each entity corresponds a table whose columns are a local identifier and URI*s* from each dataset mapped to that entity. To each relationship corresponds a table whose columns are the local identifiers of its subject and its object. The number of individuals collected with this method starting from a list of 549 genes (282 positive and 267 negative examples) are indicated in Table 1 and Table 2 (last column).

4 ILP Mining of LOD

4.1 ILP Experiments and Results

The aim of the mining step is to learn by ILP the concept of genes responsible for Intellectual Disability (ID) from the set of integrated triples relative to positive and negative examples of genes. The experiments were conducted with the Aleph program by setting the parameters *rule size, minpos, noise, minacc* respectively to 6, 5, 3 and 85%. As the noise value is set to 3, the *minacc* parameter allows us to ensure that the ratio between numbers of positive examples and negative examples covered by a rule is not below 85%.

The outcome of the mining experiment is used both for predictive and descriptive purposes. The predictive power of the first-order logic (FOL) rules is evaluated by cross validation whereas their descriptive power is analysed qualitatively.

Our first experiment applies to the genes and their background knowledge (*i.e.,* proteins, pathways, etc.) including their GO annotations plus their direct parents using the *is-a* relationship (denoted by $\texttt{subClass}_1$) between GO-terms. Then we wanted to assess the contribution of domain knowledge by allowing 2 to 4 generalisation inferences on the *is-a* GO structure, which is a rooted directed acyclic graph. For n generalization steps, we add $2 \times n$ inference rules in the .b file (one of the three inputs of the Aleph program) as follows:

One inference rule for each i in $2 \dots n$:

$\texttt{subClass}_i(\texttt{X},\texttt{Z})$:- $\texttt{subClass}_{i-1}(\texttt{X},\texttt{Y})$, $\texttt{subClass}_1(\texttt{Y},\texttt{Z})$.

One inference rule for each i in $1 \dots n$:

$\texttt{subClass}(\texttt{X},\texttt{Y})$:- $\texttt{subClass}_i(\texttt{X},\texttt{Y})$.

One rule expressing the reflexivity of the `subClass` relationship :

`subClass(X,X) :- goterm(X).`

In this study the mining experiment was executed with n varying from 1 to 4, leading to a maximum of 1 ($G1$ experiment), 2 ($G2$ experiment), 3 ($G3$ experiment), and 4 ($G4$ experiment) generalization steps respectively. Examination of the resulting 4 theories revealed that the produced rules mostly contain predicates related to GO-terms. Other predicates representing pathways or interactions between proteins occur very rarely. This can be explained by the fact that GO annotations are plethoric compared to data on pathways, protein domains or protein-protein interactions. This motivated us to run a fifth experiment (named $no-GO$) for analyzing all predicates excepting the GO-term facts. Complete theories produced in the five experiments are accessible online at `http://www.loria.fr/~coulet/dils14/theories.pdf`. Table 3 shows several metrics calculated for monitoring the effect of adding GO-term facts and increasing the number of generalization steps. The number of rules in the theory doubles when adding GO-term facts (from $no-GO$ to $G1$) and the average number of covered examples increases from 8.4 to 14, with the maximum increasing

Table 3. Statistics on the theories produced by our five experiments. avg/max/min pos covered: Average/maximum/minimum number of positive examples covered by the rules of each theory.

Experiment	#rules	avg pos covered	max pos covered	min pos covered
$no-GO$	11	8.4	15	5
$G1$	22	14	35	6
$G2$	19	15.5	38	6
$G3$	18	15.1	39	6
$G4$	16	16.2	42	5

from 15 to 35. This indicates that GO-term facts play a very positive role in the ILP process during learning. As the number of generalization steps increases from 1 to 4 the number of rules decreases (from 22 to 16) whereas the average number of covered examples slightly increases from 14 to 16.2, with a increase of the maximum (from 35 to 42). These results confirm the intuition that with more generalization steps, theories tend to become more compact with less rules, each of them covering more examples. However it is important at that stage to compare the predictive power of each theory.

4.2 Evaluation of the Results

We first evaluate the outcome of the mining step from a predictive point of view using cross-validation. Dedicated Knime workflows were used for that purpose [30, 31]. During cross-validation, a gene is predicted as responsible for intellectual disability if it is covered by at least one rule of the theory. Otherwise, it is predicted as not responsible for intellectual disability.

Table 4 reports the results of the 100-fold cross validation of ILP learning for the experiments $no-GO$ and $G1$ to $G4$. The results show that without GO-term facts ($no-GO$), the prediction accuracy is rather low (57.6%) with a high specificity but a very low sensitivity. When using GO-terms the prediction indicators are better. They improve up to an accuracy of 72.6% as we allow Aleph to use more domain knowledge (by performing more generalization).

Table 4. Results of the 100-fold cross validation the theories produced by the 5 experiments. TP/FP: True/False Positives, TN/FN: True/False Negatives, Sens.: Sensitivity, Spec.: Specificity, Acc: Accuracy.

Experiment	TP	FP	TN	FN	Sens.(%)	Spec.(%)	Acc.(%)
$no-GO$	67	**19**	**248**	213	23.9	**92.9**	57.6
$G1$	154	42	225	126	55	84.3	69.3
$G2$	162	54	213	118	57.9	78.9	68.6
$G3$	156	44	223	124	55.79	83.5	69.3
$G4$	**166**	36	231	**114**	**59.3**	86.5	**72.6**

4.3 Qualitative Analysis and Discussion

We analyze here the obtained theories from the concept-to-learn point of view, *i.e.,* how well do the rules characterize genes responsible for ID? In the absence of GO-term facts ($no - GO$ experiment), we observe several rules containing predicates related to chromosomal localization such as rule 4 and 5 pointing to chromosomes 1 and X as possible reservoirs of genes for ID. In addition, rule 8:

$$\texttt{is\_responsible(A) :- gene\_ch\_band(A, '22q13').}$$

points to a more constraint location on chromosome 22. Other rules contain pathway predicates (rules 6, 7, 9, 10, 11) in which one can mostly recognize pathways involved in the metabolism of the cell. Indeed inherited metabolic disorders are considered as an important etiology for intellectual disability [32].

In the presence of GO-term facts (experiments $G1$ to $G4$), the repertoire of GO-terms appearing in the predicates either as direct protein annotation or as common ancestor after generalization varies with the experiment and the generalization degree. In total we counted 47, 7 and 14 distinct GO-terms pertainining from the Biological Process (BP), Molecular Function (MF) and Cellular Component (CC) aspects of the GO ontology respectively. Among the BP-terms of GO we could again recognize terms describing metabolic processes of the cell, but also terms related to gene expression mechanisms and to nervous system development which make sense when dealing with ID. Interesting rules are combining `protein_bp` and `protein_mf` predicates such as rule 16 in the $G4$ theory:

$$\begin{aligned}\texttt{is\_responsible(A) :- gene\_protein(A, B),}\ &\texttt{protein\_mf(B, C),}\\ &\texttt{subClass(C, 'ion binding'),}\\ &\texttt{protein\_bp(B, 'carbohydrate metabolic process').}\end{aligned}$$

Such rules suggest that the descriptive power of the theories increases when domain knowledge is taken into account. The value of adding generalization can be illustrated on rules sharing `subClass` predicates concerning organonitrogen compound metabolism. Rules 3 from $G1$ and 7 from $G2$ theories both contain the `subClass(C, 'organonitrogen compound` **metabolic** `process')` predicate and each of them covers 23 positive examples. Rules 4 from $G3$ and 4 from $G4$ theories both contain the `subClass(C, 'organonitrogen compound` **catabolic** `process')` predicate which refers to a more specific GO-term than in $G1$ and $G2$ (catabolism is one aspect of metabolism) but these rules cover 39 positive samples in the $G3$ and 42 positive samples in the $G4$ theories. Thus allowing for more generalization steps has helped to increase the coverage of the rule but also to better specify the feature shared by the positive samples. Several other examples similar to this one are found across the $G1$ to $G4$ theories. In our hands, the limitation for generalization is yet the execution time. It becomes very long when the generalisation level increases, because the number of inferred facts to consider increases. Parallel computing becomes necessary especially for cross-validation.

5 Conclusion and Perspectives

One current limit of using LOD is that some datasets are outdated, as most of them are built punctually from non-LOD resources. Thus, LOD may be incomplete or may contain data no longer endorsed by its original publisher. For instance, the KEGG pathway `<http://bio2rdf.org/path:hsa00252>` found in Bio2RDF no longer exists in the KEGG Pathway database (`hsa00252` is not associated to a pathway anymore). This may cause conflitcs when one wants to integrate LOD data.

The imbalance between GO-term facts and other facts in the learning dataset leads to a majority of predicates referring to GO-terms in the theories. One way to avoid the overwhelming effect of the GO-term is be to limit their number on the basis of the evidence code associated with GO annotations. These codes specify the way an annotation has been assigned to a protein. Filtering out annotations with *IEA* (Inferred from Electronic Annotation) code would decrease the volume of GO annotations and restrain the study to well established ones. Another solution is to run two separate experiments on two complementary datasets composed on the one hand of GO-term facts and on the other hand, and of other predicates. This would lead to two separate theories that would then be combined by designing and evaluating a global prediction model as proposed in [33, 34]. This will require a selection of the best rules from each theory. Indeed, the theory rules could be evaluated with respect to their statistical significance, for instance by evaluating how specific they are to the genes responsible for intellectual disability when compared to all other known genes.

Experiments reported in this paper demonstrate the interest of ILP methods for mining an integrated dataset derived from LOD. Other mining methods can be tested such as graph-based ones, which are also adapted to graph representations. However ILP methods allow to take into account valuable knowledge that is available within biomedical ontologies.

References

[1] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[2] Erick Antezana, Martin Kuiper, and Vladimir Mironov. Biological knowledge management: the emerging role of the semantic web technologies. *Briefings in Bioinformatics*, 10(4):392–407, 2009.

[3] Francois Belleau, Marc-Alexandre Nolin, Nicole Tourigny, Philippe Rigault, and Jean Morissette. Bio2rdf: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics*, 41(5):706 – 716, 2008. Semantic Mashup of Biomedical Data.

[4] Matthias Samwald, Anja Jentzsch, Christopher Bouton, Claus Kallesøe, Egon L. Willighagen, Janos Hajagos, M. Scott Marshall, Eric Prud'hommeaux, Oktie Hassanzadeh, Elgar Pichler, and Susie Stephens. Linked open drug data for pharmaceutical research and development. *J. Cheminformatics*, 3:19, 2011.

[5] Akira R. Kinjo, Hirofumi Suzuki, Reiko Yamashita, Yasuyo Ikegawa, Takahiro Kudou, Reiko Igarashi, Yumiko Kengaku, Hasumi Cho, Daron M. Standley, Atsushi Nakagawa, and Haruki Nakamura. Protein data bank japan (pdbj): maintaining a structural data archive and resource description framework format. *Nucleic Acids Research*, 40(Database-Issue):453–460, 2012.

[6] The EBI RDF Plateform:. http://www.ebi.ac.uk/rdf/.

[7] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[8] Adrien Coulet, Malika Smaïl-Tabbone, Pascale Benlian, Amedeo Napoli, and Marie-Dominique Devignes. Ontology-guided data preparation for discovering genotype-phenotype relationships. *BMC bioinformatics*, 9(Suppl 4):S3, 2008.

[9] Adrien Coulet, Malika Smaïl-Tabbone, Amedeo Napoli, and Marie-Dominique Devignes. Ontology-based knowledge discovery in pharmacogenomics. In *Software Tools and Algorithms for Biological Systems*, pages 357–366. Springer, 2011.

[10] Benjamin M. Good and Mark D. Wilkinson. The life sciences semantic web is full of creeps! *Briefings in Bioinformatics*, 7(3):275–286, 2006.

[11] M. Scott Marshall, Richard D. Boyce, Helena F. Deus, Jun Zhao, Egon L. Willighagen, Matthias Samwald, Elgar Pichler, Janos Hajagos, Eric Prud'hommeaux, and Susie Stephens. Emerging practices for mapping and linking life sciences data using rdf - a case series. *J. Web Sem.*, 14:2–13, 2012.

[12] Mehwish Alam, Melisachew Wudage Chekol, Adrien Coulet, Amedeo Napoli, and Malika Smaïl-Tabbone. Lattice based data access (lbda): An approach for organizing and accessing linked open data in biology. In *Proceedings of the International Workshop on Data Mining on Linked Data (DMoLD 2013)*, 2013.

[13] Alison Callahan, José Cruz-Toledo, and Michel Dumontier. Querying bio2rdf linked open data with a global schema. In *Proceedings of Bio-ontologies SIG*, 2012.

[14] Mark D. Wilkinson, Benjamin P. Vandervalk, and E. Luke McCarthy. The semantic automated discovery and integration (sadi) web service design-pattern, api and reference implementation. *J. Biomedical Semantics*, 2:8, 2011.

[15] Pedro Lopes and José Luís Oliveira. Coeus: "semantic web in a box" for biomedical applications. *J. Biomedical Semantics*, 3:11, 2012.

[16] Johanna Völker, Heiko Paulheim, Jens Lehmann, Mathias Niepert, and Harald Sack, editors. *Proceedings of the Second International Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data*, volume 992 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.

[17] Claudia d'Amato, Petr Berka, Vojtech Svátek, and Krzysztof Wecel, editors. *Proceedings of the International Workshop on Data Mining on Linked Data*, volume 1082 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.

[18] Aldo Gangemi, Andrea Giovanni Nuzzolese, Valentina Presutti, Francesco Draicchio, Alberto Musetti, and Paolo Ciancarini. Automatic typing of dbpedia entities. In *The Semantic Web–ISWC 2012*, pages 65–81. Springer, 2012.

[19] Axel-Cyrille Ngonga Ngomo. Link discovery with guaranteed reduction ratio in affine spaces with minkowski measures. In *The Semantic Web–ISWC 2012*, pages 378–393. Springer, 2012.

[20] Mengling Xu, Zhichun Wang, Rongfang Bie, Juanzi Li, Chen Zheng, Wantian Ke, and Mingquan Zhou. Discovering missing semantic relations between entities in wikipedia. In *The Semantic Web–ISWC 2013*, pages 673–686. Springer, 2013.

[21] Christian Y. A. Brenninkmeijer, Ian Dunlop, Carole A. Goble, Alasdair J. G. Gray, Steve Pettifer, and Robert Stevens. Computing identity co-reference across drug discovery datasets. In *Proceedings of the 6th International Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS 2013)*, 2013.

[22] Bethany Percha, Yael Garten, and RUSS B Altman. Discovery and explanation of drug-drug interactions via text mining. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 410–421. World Scientific, 2012.

[23] Jyotishman Pathak, Richard C Kiefer, and Christopher G Chute. Mining anticoagulant drug-drug interactions from electronic health records using linked data. In *Data Integration in the Life Sciences*, pages 128–140. Springer, 2013.

[24] Mathieu d'Aquin, Gabriel Kronberger, and Mari Carmen Suárez-Figueroa. Combining data mining and ontology engineering to enrich ontologies and linked data. In *Workshop: Knowledge Discovery and Data Mining Meets Linked Open Data-Know@ LOD at Extended Semantic Web Conference (ESWC)*, volume 2012, 2012.

[25] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pages 413–422. International World Wide Web Conferences Steering Committee, 2013.

[26] Stephen Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

[27] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19(20):629–679, 1994.

[28] Ashwin Srinivasan. The aleph manual. available at http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/aleph/, 2007.

[29] Jennifer K Inlow and Linda L Restifo. Molecular and comparative genetics of mental retardation. *Genetics*, 166(2):835–881, 2004.

[30] Michael R Berthold, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. Knime-the konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1):26–31, 2009.

[31] Renaud Grisoni, Emmanuel Bresso, Marie-Dominique Devignes, and Malika Smaïl-Tabbone. Méthodologie et outils pour l'extraction de connaissances par Programmation Logique Inductive (PLI) (Poster). In *13ème Conférence Francophone sur l'Extraction et la Gestion des Connaissances- EGC 2013*, Toulouse, France, 2013.

[32] Clara DM van Karnebeek and Sylvia Stockler. Treatable inborn errors of metabolism causing intellectual disability: a systematic literature review. *Molecular genetics and metabolism*, 105(3):368–381, 2012.

[33] Michael R. Berthold, Katharina Morik, and Arno Siebes, editors. *Parallel Universes and Local Patterns*, volume 07181 of *Dagstuhl Seminar Proceedings*, 2007.

[34] Arno Knobbe, Bruno Crémilleux, Johannes Fürnkranz, and Martin Scholz. From local patterns to global models: The lego approach to data mining. In *International Workshop From Local Patterns to Global Models co-located with ECML/PKDD'08*, pages 1–16, Antwerp, Belgium, September 2008.