Wageningen UR

LinkInRDF : How to link a gene to a phenotype using semantic approaches.

Ferrières Jean-Charles

<u>Master Thesis Report</u>

Plant Breeding Research
August 2010 – January 2011

<u>Master :</u> MSc BioInformatics

<u>Student :</u> Ferrières Jean-Charles

<u>Student Number :</u> 880330239100

<u>Thesis Code :</u> PBR-80433 Thesis Plant Breeding

Approved by the examination board

<u>Supervisors:</u> Pierre-Yves Chibon and Dr. Richard Finkers

Contents

Table of Figures and Tables

I. Introduction

Nowadays information generated increase exponentially and in the scientific community faces the problem of storage and linkage between information. We arrive at a point where the numbers of links (relationship between data) that refer to one data is constantly growing. Biologically speaking this means for instance, that for one gene, exist several different links to a protein, a literature or to the gene itself. This issue can also be found in QTL mapping (Phillip McClean, 1998) (Lynch, 1998). Indeed if for a QTL (interval between two markers) we have 500 genes, how can we deal with the amount of information to retrieve and filter only what we are interesting in? How can we know if all genes are playing a role in the phenotype? Which of those 500 genes is/are actually involved, related to the QTL of interest? It would therefore be handy to find easily candidates genes for a QTL trait. However to go through all the information would surely be time consuming if not messing up with human minds. Nevertheless because of storage issue and to increase efficient display of data, information is already present on the web which provide linked data in the form of web pages which communicate to each other through hyperlink. HTML ((HyperText Markup Language) is the predominant language for web pages and provides means to create structured documents, interactive forms and allows images and objects to be added. Nevertheless it has some drawbacks. For instance with HTML the user produces the content that the machine cannot understand. In other words the computer is reading a syntax but doesn’t understand the meaning of the words. Therefore the machine cannot make any reasoning. HTML is only linking pages, documents and cannot link data to each other (Linkeddatatools.com, 2009). Thus a way of linking data between systems or entities, of creating relationship and properties between things has been thought and is known as the Semantic web. But what is the semantic web?

The Semantic Web: a 3.0 web technology

The Semantic Web is a technology that link up data in a way that it is easily understood by machines. We can think of an omnipresent and intelligent web that understands the meaning of the user's requests rather than the syntax. It becomes an active help contrary to the passively help of the web 2.0 (Tim Berners-Lee, 2008). It no longer looks up keywords or pages but only data meaning. It is a way of accessing and filtering data with efficiency and making information search and data integration more efficient. The aim of the semantic web is to relate data and distribute the content to a specific user. Therefore it is crucial to find a solution to give a sense to the content that computers can understand and deal with efficiently.

The last few years efforts have been spent on improving data representation according to open standards. One of the most open standards (Croset, 2010) is the representation of data in RDF (Resource Description Framework) and its triples store data (Samwald 2010; Cruz Toledo 2010; McCusker 2009). RDF has an abstract syntax that reflects a simple graph-based data model. The aim of RDF is based on several criteria described by the W3C (W3C 2004). In other words RDF was developed to represent information in a flexible and non-constraining way. Information is collected in models (aggregation of information in RDF format) which can be displayed into graphs. Thus RDF is a simple language to refer to object and their relationship via triples. A triple takes knowledge and break it into contents (Figure 1&2). It must consist of:

- Subject: Thing that is described
- Predicate: Attribute of the thing that is described (also called property)
- Object: Thing that is referred to with the predicate

Example: (Subject) aGene -> (Predicate) hasProduct -> (Object) aProtein

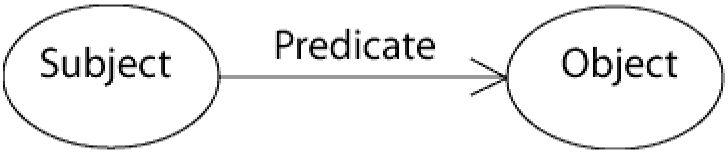
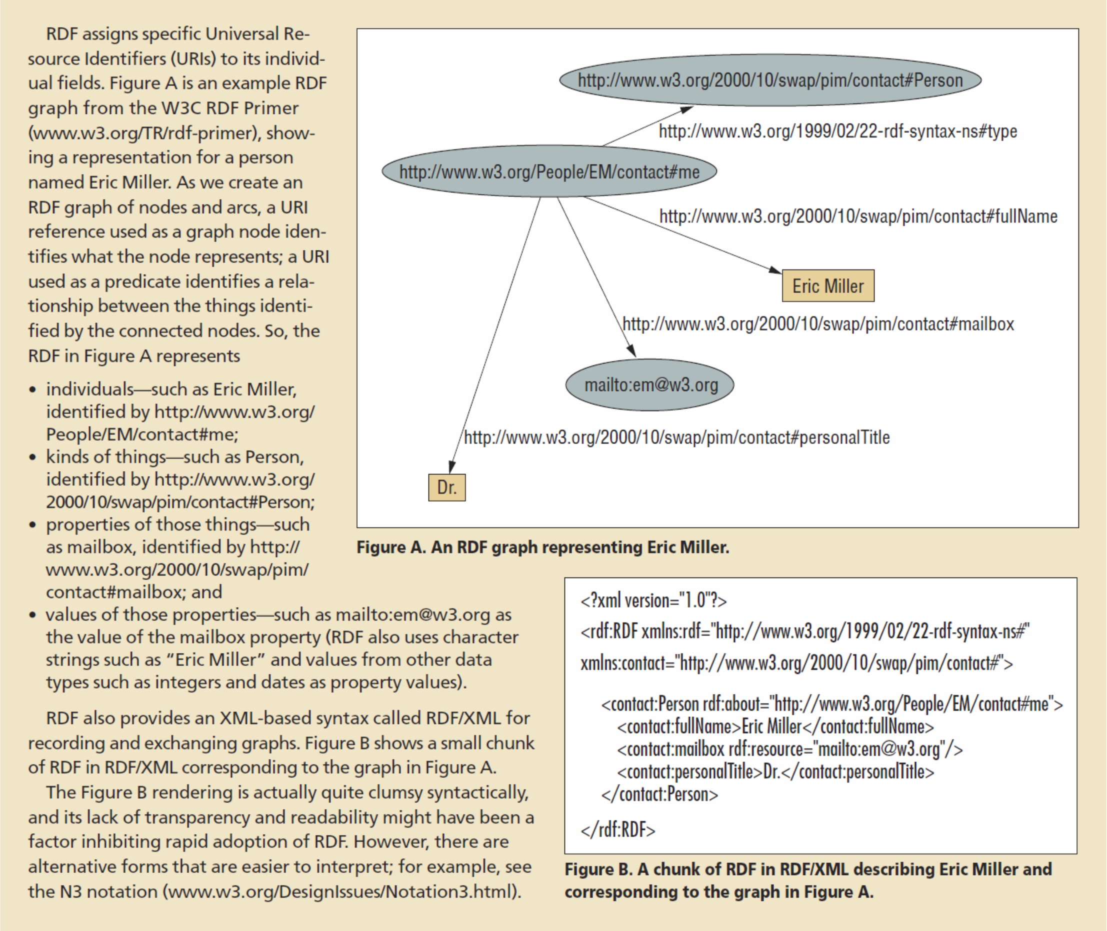


Figure 1 : RDF Graph. The nodes always stand for Subject (aGene) and Object (aProtein). The predicate (hasProduct) refers to the relationship via an arc always pointed toward the object.



RDF assigns specific Universal Resource Identifiers (URIs) to its individual fields. Figure A is an example RDF graph from the W3C RDF Primer (www.w3.org/TR/rdf-primer), showing a representation for a person named Eric Miller. As we create an RDF graph of nodes and arcs, a URI reference used as a graph node identifies what the node represents; a URI used as a predicate identifies a relationship between the things identified by the connected nodes. So, the RDF in Figure A represents

- individuals—such as Eric Miller, identified by http://www.w3.org/People/EM/contact#me;
- kinds of things—such as Person, identified by http://www.w3.org/2000/10/swap/pim/contact#Person;
- properties of those things—such as mailbox, identified by http://www.w3.org/2000/10/swap/pim/contact#mailbox; and
- values of those properties—such as mailto:em@w3.org as the value of the mailbox property (RDF also uses character strings such as "Eric Miller" and values from other data types such as integers and dates as property values).

RDF also provides an XML-based syntax called RDF/XML for recording and exchanging graphs. Figure B shows a small chunk of RDF in RDF/XML corresponding to the graph in Figure A.

The Figure B rendering is actually quite clumsy syntactically, and its lack of transparency and readability might have been a factor inhibiting rapid adoption of RDF. However, there are alternative forms that are easier to interpret; for example, see the N3 notation (www.w3.org/DesignIssues/Notation3.html).

http://www.w3.org/2000/10/swap/pim/contact#Person
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/People/EM/contact#me
http://www.w3.org/2000/10/swap/pim/contact#fullName
Eric Miller
http://www.w3.org/2000/10/swap/pim/contact#mailbox
mailto:em@w3.org
http://www.w3.org/2000/10/swap/pim/contact#personalTitle
Dr.

Figure A. An RDF graph representing Eric Miller.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>

</rdf:RDF>
```

Figure B. A chunk of RDF in RDF/XML describing Eric Miller and corresponding to the graph in Figure A.

Figure 2 : Resource Description Framework. (Nigel Shadbolt, 2006)

However RDF is a simple language to describe properties or relationship between entities and has a limited vocabulary. Therefore comes RDF schema (RDFS) which is an extensible knowledge representation language also known as RDF vocabulary description language. RDFS is used to provide basic elements for the description of ontologies using classes and properties to define entities (W3C, 2004).

Moreover different semantic languages exist as OWL (Web Ontology Language) (McGuinness, 2004). OWL is intended to be compatible with RDF Schema (RDFS), and to be capable of augmenting the meanings of existing Resource Description Framework (RDF) vocabulary (Hayes, 2004). So, the meaning of OWL ontologies is defined by extension of the RDFS meaning, and OWL appears then to be a semantic extension of RDF that provides a more detailed ontology vocabulary and is therefore use to define the types of object and the way they are related to each other (E. Neumann, 2008)(Patel-Schneider, 2004).

In this study we looked how to link a gene of interest to its phenotype using semantic approaches and that's how LinkInRDF was created. Knowing that the main source of information is bio2rdf, which focuses essentially on human and mouse genomes (Belleau, 2008), we could not investigate plant genes or at least not retrieve enough information. Therefore a set of different genes has been selected. At first a human gene, the pyruvate dehydrogenase E1 component subunit beta (*pdhB*), coding for a protein which takes place in a protein complex was chosen. PdhB if mutated can lead to secondary effect as lactic acidosis or development delay (Okajima, 2008). We expected this gene to bring a lot of information in RDF format via bio2RDF. For similar reasons a house mouse gene coding for the hexokinase 1 protein (*Hk1*) mainly involved in sugar metabolism was chosen. An overexpression of this gene presents unwilling glucose phosphorylation (Arora, 1990). Also two genes from yeast have been tested. The gene Ppq1p coding for a protein involved in amino acid metabolism (serine/threonine phosphatase) and the nuclear cap binding protein *CBC2* which lacks information about a possible involvement in any pathway but is crucial for the development (Lall, 2005). Finally a plant gene has been tested with little expectation regarding the amount of information in semantic format. This last gene selected comes from potato and is coding for the beta-carotene hydroxylase (*bch*) protein known to be responsible of tuber flesh color in potato (Kloosterman, 2010). Those genes were used as a starting point to retrieve all possible information available in RDF format or commonly said, using a semantic approach.

II. Materials and Methods

In order to create a program that would answer the biological question we used a virtual machine based Linux (Fedora 32bits). The hardware possesses 1 Go memory and run on a single core processor. The software created doesn't required a specific environment. However more investigation needs to be applied to evaluate the behaviour of the software under Windows or Macintosh. Also the program is java dependant.

First of all we used for programmatic oriented object language Java and JDK (Java development Kit) version 6u23 for Linux to produce the program. In order to work with semantic we focused on RDF which is a World Wide Web Consortium (W3C) specification designed as a metadata model for Web resources. It is now used as a general method of modelling knowledge. In this study RDF is the format used to retrieve semantic data and data written in a file are display in RDF format.

In a second time, since semantic information are wanted, the main source of information chosen that provides data in RDF format is bio2RDF. Indeed it is the only database available nowadays that contains biological information in RDF format. Although Bio2RDF aim to standardize data into semantic formats (RDF) (Nolin, 2008), it focuses essentially on human and mouse genomes. It combines RDF graphs of 65 million triples, 8million topics (Belleau, 2008) and collects information over 42 different public databases. Those numerous triples are the reason why the study retrieves mainly data via bio2RDF. Bio2RDF can be consulted at *www.bio2rdf.org* and the 42 databases can be query through bio2RDF using the recent query language SPARQL. SPARQL is indeed a query language and a data access protocol that has been thought for the Semantic Web. It consists of a query language, a means of conveying a query to a query processor service, and the XML format in which query results will be returned. However to use SPARQL an engine is necessary. Therefore Jena 2.6.4 (*http://sourceforge.net/projects/jena/*), which is a Java framework for building Semantic Web applications, was used. It provides a programmatic environment for RDF, RDFS OWL and SPARQL and includes a rule-based inference engine ARQ 2.8.7. ARQ is indeed a query engine that supports SPAQRL RDF query language (Cruz, 2006). Data retrieved from bio2RDF are geneID information using *http://geneid.bio2rdf.org/sparql endpoint* and information from pubmed (authors, chemicals, titles, pubmedID) using *http://pubmed.bio2rdf.org/sparql.*An overview of the process followed by LinkInRDF is shown in Appendix as well as the class diagram (Figure 7 & 8).

In a third time Uniprot, a central database of protein sequence and function created by joining the information contained in Swiss-Prot, was also used. Indeed it is the main source for protein information and is available in RDF. Each protein can be consulted online in RDF format at www.uniprot.org. It can however overlap with bio2RDF. Since the latest works mainly on human and mouse genomes, Uniprot provides the complementary information or information for other proteins that bio2RDF does not support. Another reason why we choose Uniprot is because it collects data from SwissProt that is a protein database for protein reviewed only. Therefore in Uniprot can be retrieved both reviewed and not reviewed protein. That is also the reason why the program handle a reviewed Boolean that allow the user to choose between verified information or all information (see Results figure 4).

In addition, considering that a same gene can have different accession number according to different databases we also used BioJava 3.0, a free Java framework available at *http://biojava.org/wiki/BioJava:Download*. BioJava is an open-source project dedicated to providing a Java framework for processing biological data. It provides analytical and statistical routines, parsers for common file formats and allows the manipulation of sequences and 3D structures (Holland, 2008). Therefore BioJava was used to parse file from genBank or Embl that are not in RDF format and not found in bio2RDF.

Also additional information were retrieved from two enzymatic and pathways databases. The first one is Kegg (Kyoto Encyclopedia of Genes and Genomes) which is an online database dealing with genomes, enzymatic pathways, and biological chemicals (Kanehisa, 1997, 2004 and 2006). It consists of 16 main databases and can be utilized for modeling and simulation, browsing and retrieval of data. The second one is BRENDA which is a collection of enzyme functional data available to the scientific community and that comprises molecular and biochemical information on enzymes (Bathelmes, 2006)(Chang, 2008).

Table 1: Source of information for studied genes, proteins and pathways.

Source	Gene	Protein	Pathway
	NCBI geneID:15275	Uniprot P11770	BRENDA EC 2.7.1.1
	NCBI geneID:855923	Uniprot P32945	BRENDA EC 3.1.3.16
	NCBI geneID:855925	Uniprot Q08920	NC
	NCBI geneID:5162	Uniprot P11177	ko00650 Kegg
			ko00290 Kegg
			ko00620 Kegg
			ko00010 Kegg
	NCBI gu233534	Uniprot D2JN88	NC

Finally we use some genetic material. Five genes were selected. A gene (Hk1 [geneID|15275]) from *Mus musculus* (Mouse) coding for the hexokinase 1 protein [Uniprot|P17710] involved in sugar metabolisms, Amino sugar and nucleotide sugar metabolism among others (see table 1).

A gene (Ppq1p [geneID|855923]) from *Saccharomyces cerevisiae*(Yeast) which code for a protein [Uniprot|P32945] involved in amino acid metabolism and more specifically known as a serine/threonine phosphatase protein. Besides a second gene from *Saccharomyces cerevisiae* was chosen (*CBC2* [geneID|855925]). It codes for a nuclear cap binding protein [Uniprot|Q08920] and no information is available regarding a possible involvement in a pathway or a phenotype.

Moreover the Pyruvate dehydrogenase E1 component subunit beta (*pdhB* [geneID|5162*)* from *homo sapiens* was tested. This gene is well known and well studied. It encodes the protein of the same name [Uniprot|P11177] that is involved in a protein complex and take part in pathways that use pyruvate (Table 1).

At last a potato gene, the beta-carotene hydroxylase (*bch* [gb| gu233534]) from *Solanum phureja x Solanum tuberosum* USW42 hybrid clone c was used. This gene codes for the protein of the same name [Uniprot|D3JN88] and is associated to the yellow flesh color in potato.

III. Results

To test the genes describe in Materials and methods their respective IDs were used. Thus for a given gene ID information have been retrieved. First of all from bio2RDF information regarding the gene of interest are selected in a model that we named gene model (Figure 3). The geneID SPARQL endpoint is therefore used and each information related to the gene of interest is qualified as an attribute.

From the gene model the attribute protein ID is selected and therefore all information in RDF format related to the specified protein are as well put into a new model, the protein model. For each protein titles from related publications are retrieved and allow us to make a query onto the PubMed SPARQL endpoint to get all the information available about publications (authors, PubMed ID, chemicals, titles). Every information are then collapsed into a single model which create a RDF file (Figure 4).

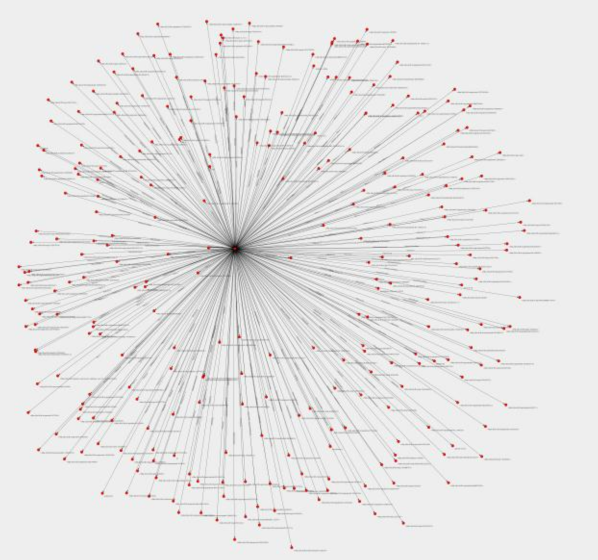
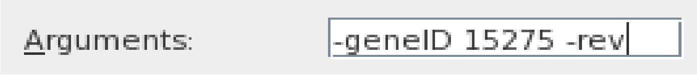


Figure 3 : Graph from the hexokinase 1 [geneID|15275]. The central point (node) represent the gene linked to its attributes.

Regarding the command line, the user gives in arguments a gene ID, precede by the database it refers to, and a Boolean which indicate whether or not the user wants information about all the proteins found or only those which have been reviewed by Uniprot (Figure 5). In the latest case the presence of the Boolean means the user wants reviewed protein only. On the contrary the absence of the “-rev” argument means the user agrees to retrieve information about every protein found in the process.



Arguments: -geneID 15275 -rev

Figure 4 : Command line example for running LinkInRDF. First argument with the prefix "-" must be the database where the ID comes from (geneID, genBank or EMBL, Uniprot/swissprot). Second argument is optional. In this example the gene comes from geneID and only reviewed protein are collected.

All along the run of the program, pictures of model are generated as graph with for nomination the name of the gene, protein or publication printed. Those pictures give a overview of the complex relationship between all entities and the size can vary according to the entity. However a picture of the complete model is not generated because it would condensate information in a non-readable way.

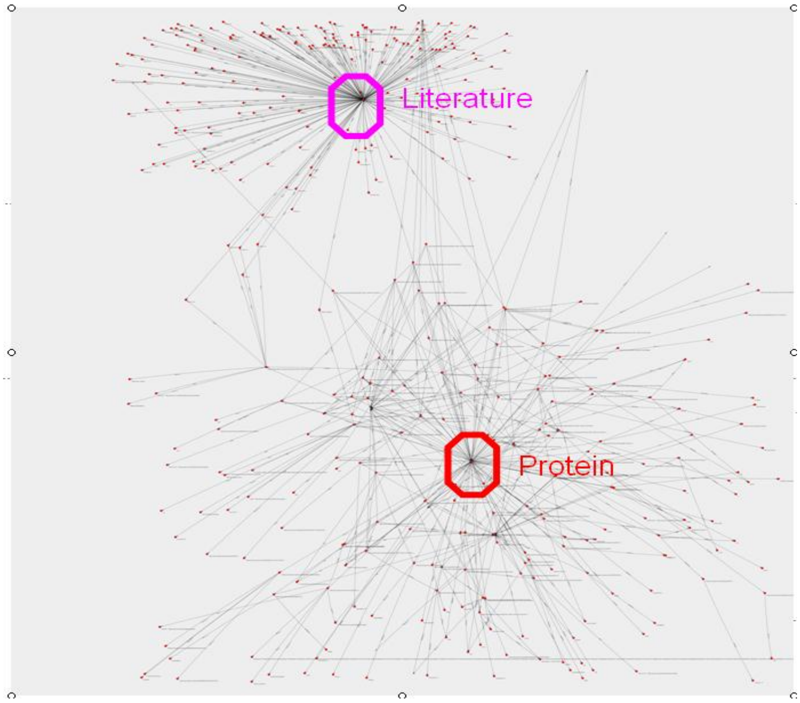
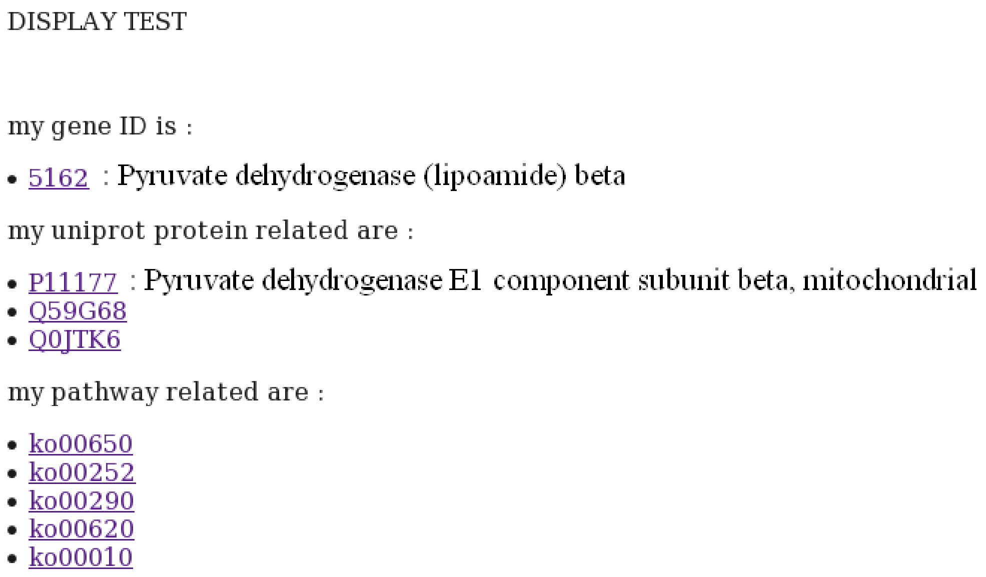


Figure 5 : Graph representation of the RDF model of the hexokinase 1 protein. The red endpoint represent the protein linked to one citation in this example (magenta endpoint). The citation itself refers to many subject that can be authors, chemicals…

Besides we manage to make a relation between pathways where the protein is interacting and the gene of interest. Indeed we found links to pathways in which ones the gene of interest is involved. For instance we knew that the pyruvate dehydrogenase E1 component subunit beta from *homo sapiens* was involved in particular in pyruvate metabolism. Using LinkInRDF we got for output links that refers to Kegg references in bio2RDF (Figure 6). The tool was able to retrieve five pathways of the related pathways(see chapter Material and Mehthods) : koOO650 that refers to the butanoate metabolism , koOO290 to the Valine, Leucine and Isoleucine biosynthesis, ko00620 to the pyruvate metabolism and ko00010 to the glyclolysis/gluconeogenesis. Also from the protein model in most cases a reference to Kegg genome database provides a link to known pathways where the pyruvate dehydrogenase takes part of as well as a link to the OMIM database referring to the pyruvate dehydrogenase complex defiency. In this example linkInRDF manages to link a gene of interest to its phenotype using semantic (RDF).



DISPLAY TEST

my gene ID is :

- 5162 : Pyruvate dehydrogenase (lipoamide) beta

my uniprot protein related are :

- P11177 : Pyruvate dehydrogenase E1 component subunit beta, mitochondrial
- Q59G68
- Q0JTK6

my pathway related are :

- ko00650
- ko00252
- ko00290
- ko00620
- ko00010

Figure 6 : Result of the research for the pyruvate dehydrogenase E1 component subunit beta from *homo sapiens* (geneID|5162). The three proteins are actually the same but only the protein P11177 is reviewed in Uniprot database.

Similarly the gene of hexokinase 1 (geneID|15275) from *mus musculus* was found to be involved in EC 2.7.1.1 according to BRENDA. There we can retrieve a list of all the pathway from Kegg database known to be effective with the product. Therefore the EC number is actually the reaction in which the protein of interest is involved in different pathway (mentioned in chapter Material and Methods). Also many proteins are found to be related to this gene but only one of them has been reviewed by uniprot: P17710.

Following the same logic the serine/threonine phosphatase gene (Ppq1p: geneID|855923) from *Saccharomyces cerevisiae* is found to be involved in EC 3.1.3.16 according to BRENDA. Although not direct link to Kegg database were available, the EC number refers to the reaction involving different pathways. We therefore have access to pathways in Kegg that correlate with the mentioned pathway about the protein of interest. Thus once again LinkInRDF has successfully linked a gene to a pathway using semantic approach.

Another example used was the gene CBC2I(geneID|855925) from *Saccharomyces cerevisiae* as well. It has for product a nuclear binding cap protein subunit 2 (Uniprot|Q08920)**.** This gene has no information regarding its possible implication in any pathway or key role in a phenotype. This lack of information is confirmed using a semantic approach since no pathways were found and no direct proven key role in a phenotype. However information about a key role in development that we had collected before the study is confirmed by the literature retrieved through LinkInRDF (Fortes, 1999).

Finally for the beta carotene hydroxylase gene (bch) from potato as expected little information was available in RDF format from bio2RDF. Although the protein is available in Uniprot, LinkInRDF could only retrieve information about the literature that prove that *bch* is responsible for the yellow flesh color in potato. Despite we retrieved the same information through our tool than without the tool, it shows that using semantic we can effectively find the relation between a gene of interest (bch) and a pathway or the phenotype related (yellow flesh color).

The few examples used have shown that within the limited amount of information available in semantic format (RDF mainly) we can draw a relation from a gene to its phenotype or to the pathway its product is involved in.

IV. Discussion

At first, the results showed that we can obtain the same information about a gene than what we can get manually going through the web. Also the efficiency is better since a simple click brought us up to the pathway or phenotype related to the selected gene. As well, every entity is directly link to another one by one or several attributes. Unfortunately the program is at the moment limited in two directions. One is that we aggregate a maximum of information as possible which could lead to storage or memory issues in the case of simultaneous and multiple queries or multi-gene complex research. Indeed it collects all the data available about a gene. This first limitation can be easily fixed by making it search specific data dependant.

The other issue is that all information are not all available in a semantic format and could lead to some limitation to the semantic research and the display of information to the user. As mentioned above it would be possible to make a restricted selection of information and hopefully it could be extending to a friendly user interface so that the biologist can decide himself which data are relevant. Although RDF data available nowadays are not so numerous we can expect that many more will be available in a close future. Nevertheless we could consider converting all the data to set more data in a semantic format.

In the second time the efficiency needs to be pointed out. Despite the semantic approach brings efficiency and speed to research and display of information, it is web dependant. Indeed the data retrieved are from different web services which slow down the efficiency. Whether the data are not available in semantic format or the servers are down or updating or the query demand is too high to treat all requests. Let us not forget the time that increase with the number of queries. Besides links to data are often duplicates if we consider that Uniprot have its own data in RDF format and bio2RDF duplicate the URIs to its own URIs. In other words when Uniprot proposes an URI for my gene of interest such as `http://purl.unipprot.org/geneid/15275`, while Bio2RDF creates its own

URI http://bio2rdf.org/geneid:15275. It is therefore important to consider an alternative solution. First of all we can think of an universal system to deal with URI that refers to a specific data. It consists on officialise URIs for data implying no duplicate and no different way of naming the same object. A second solution is to combine the data in RDF format from the official different databases that provide information (Kegg, Uniprot, GoTerms…) and to link them together using their URIs only. A third possibility is to use OWL format that allow us to refer a same object with key word “same as”. Therefore we could use the former example as followed:

```
<rdf:Description rdf:about=”geneid:15275’>
<owl:sameAs
rdf:resource=http://purl.unipprot.org/geneid/15275”/>
<owl:sameAs rdf:resource=http://bio2rdf.org/geneid:15275 “/>
</rdf:Description>
```

OWL could be used whether entirely, which would mean converting data from RDF to OWL, or we could consider a combination of RDF and OWL which would refer to the same object from different database whether the data are available in RDF or in OWL. The possibility of combination needs more investigation to find out if it would be possible, handy and efficient to do so.

The last example would be more powerful since a query on geneID:15275 would return all the objects that refer to it with the subject owl:sameAS. However data are not yet so much develop into OWL and it would take time to convert all the data from RDF to OWL. But it is definitively the direction to follow for more efficiency in retrieving understandable and reliable data.

In addition to increase efficiency in transfer of information through the web it is necessary to consider an alternative solution to store data and where to retrieve them. At first we could store a mirror image of bio2RDF locally. This mirror would speed up the request answering time to the database and save us time when to deal with hundred of genes at the same time. A second option is to store our own data in local or on the web that would definitively once again increase the speed of the queries.

Besides it would increase efficiency in querying time and retrieving/returning information.

Moreover we could question the relevancy of the data. Are all data verified and confirmed or just hypothetic assumption and aggregation of data that are waiting to be controlled? This question is actually an issue to sort data wanted by the user and to judge whether or not those data are relevant. To create a filter to sort data by reliability would as well be a benefice for retrieving seeks data and no decoy or irrelevant information that would only make the work of a biologist even more complicated.

In a third part we should not forget to mention the future work that needs to be done to improve efficiency, reliability of the program. All previous suggestions should enter into consideration once the program is properly working according to user's expectation. Indeed actually the research is done according to a gene ID a GenBank or EMBL accession number or yet a uniprot ID. In the future we could increase the input to allow the user to give several genes ID at the same time. Implementation for GenBank accession number and EMBL accession number are ready but not yet optimal. Indeed the conversion from GenBank or EMBL file to RDF format is limited. It could be an option to convert more data so that the research can be extended. On the same way the idea would be to be able to start from a protein ID to retrieve all information upstream and downstream the process. At the moment the user can provide a protein ID but only the downstream process is complete. Also a user friendly interface could be done to allow a better display of information rather than to create files and pictures in a separate folder. For instance the idea is that the user can make a request by selection of specific fields and choose what for output is preferred (HTML, RDF, OWL, text, PNG or others format). The generation of pictures is not optional and that could be an option to provide. Additionally the user can load a set of gene's ID whether they are from geneID, GenBank or EMBL.

Finally an application of the program could be investigated. Two options are investigated. The first option is to develop an application as software that every user can download and install on computers. However libraries and update would need to be checked online regularly. This option is possible but does not appear to be the optimal solution. According to the request done to several different databases it would be a better choice to develop an online application accessible by everyone at anytime. It would save memory on user's computer and increase efficiency as well as an easier maintenance and storage of data. Besides the efficiency could be increase if a own data storage is develop so that the program is no longer web services dependant and require for instance only monthly updates of data. At last an online application with own storage database would be the best option since the endpoint would be located in Europe and not in Canada (bio2RDF). Therefore countries located in Europe, Africa, western countries and middle-east states would earn benefits in time.

V. Conclusion

We showed that, although the program is still to an early stage development the aim to find a link between a gene and its correlated pathways or phenotypes using semantic approaches was reached. Further work on the program would be necessary to optimize treatment of data and make it accessible to biologists. We are confident that this program will be a useful tool in the future considering a first application as the prediction candidate genes for QTL markers. Besides more commitment from the community into semantics approaches would help to improve development of tools based semantic. According to Tim Berners-Lee, proponent of the semantic web, this new technology is the future of understanding and dealing with huge amount of data but must first be demonstrated by life sciences. We definitively agree and we believe that semantic will lead to more efficiency and save a lot of time in research. We also believe that our program would be one of those life science semantic applications to convince and convert the scientific community to switch rapidly to semantic.

Acknowledgment

I would like to thank both my supervisors Dr. Richard Finkers and Pierre-Yves Chibon for guiding me and support me all along this thesis. Their supervision help me progress smoothly in the project.

I would also like to acknowledge Helena De Weerd for her assistance and help. The whole Statistics & Bioinformatics group for making me realize the value of working together as a group

Not forget Plant Breeding Research for the disposal of room and material for this thesis.

Last but not least I would like to thank my family, friends and colleagues for their support.

Literature

Arora, K.K., Fanciulli, M., Pedersen, P.L. (1990) Glucose phosphorylation in tumor cells. Cloning, sequencing, and overexpression in active form of a full-length cDNA encoding a mitochondrial bindable form of hexokinase. J. Biol. Chem.

Barthelmes J, Ebeling C, Chang A, Schomburg I, Schomburg D (2006). "BRENDA, AMENDA and FRENDA: the enzyme information system in 2007". Nucleic Acids Res 35 (Database issue): D511-D514.

Tim Berners-Lee The Semantic Web of Data, 2008

François Belleau, Nicole Tourigny, Benjamin Good, and Jean Morissette (2008). "Bio2RDF : A Semantic Web Atlas of Post Genomic Knowledge about Human and Mouse". Springer Berlin / Heidelberg. p. 153-160.

Chang A, Scheer M, Grote A, Schomburg I, Schomburg D (2008). "BRENDA, AMENDA and FRENDA the enzyme information system: new content and tools in 2009". Nucleic Acids Res (Database issue): D588-D592.

Cruz-Toledo,J., Dumontier,M., Parisien,M., Major,F. (2010) "RKB: a Semantic Web knowledge base for RNA". J Biomed Semantics. 1 Suppl 1:S2.

Cruz, I.; Decker, S.; Allemang, D.; Preist, C.; Schwabe, D.; Mika, P.; Uschold, M.; Aroyo, L. (Eds.)(2006) "The Semantic Web". 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings Series: Lecture Notes in Computer Science, Vol. 4273 Subseries: Information Systems and Applications.

Fortes P, Kufel J, Fornerod M, Polycarpou-Schwarz M, Lafontaine D, Tollervey D, Mattaj IW. (1999)**. "**Genetic and physical interactions involving the yeast nuclear cap-binding complex." Mol Cell Biol. 1999 Oct;19(10):6543-53.

Patrick Hayes (10 February 2004). "RDF Semantics". Resource Description Framework. World Wide Web Consortium. http://www.w3.org/TR/2004/REC-rdf-mt-20040210/. Retrieved 5 January 2011.

R. C. G. Holland, T. A. Down, M. Pocock, A. Prlić, D. Huen, K. James, S. Foisy, A. Dräger, A. Yates, M. Heuer and M. J. Schreiber. (2008) BioJava: an open-source framework for bioinformatics. Bioinformatics 24 (18): 2096-2097. doi: 10.1093/bioinformatics/btn397

Kanehisa M, Goto S, Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, Katayama T, Araki M, Hirakawa M.(2006) "From genomics to chemical genomics: new developments in KEGG". Nucleic Acids Res. 2006 Jan 1;34(Database issue):D354-7.

Kanehisa M, Goto S, Kawashima S, Okuno Y, Hattori M. (2004) "The KEGG resource for deciphering the genome". Nucleic Acids Res. 2004 Jan 1;32 (Database issue):D277-80.

Kanehisa, M., (1997) "A database for post-genome analysis.*"* Trends in Genetics, 1997. **13**(9): p. 375-376.

Kloosterman,B., Oortwijn,M., Celis-Gamboa,C., uitdeWilligen,J., America,T., de Vos,R., Visser,R.G.F. and Bachem,C.W.B. (2010) From QTL to candidate gene: Genetical genomics of simple and complex traits in potato using a pooling strategy. BioMed Central doi: 10.1186/1471-2164-11-158.

Lall, S., Piano, F., Davis, R.E. (2005) Caenorhabditis elegans decapping proteins: localization and functional analysis of Dcp1, Dcp2, and DcpS during embryogenesisMol. Biol. Cell

Lynch M, Walsh B (1998). "Genetics and analysis of quantitative traits. Sinauer Associatiates, Sunderland MA, chapter 15.

Phillip McClean (1998). "Mapping QTL with Molecular Markers".

McCusker,J.P., Phillips,J.A., González Beltrán,A., Finkelstein,A., Krauthammer,M. (2009) “Semantic web data warehousing for caGrid. BMC Bioinformatics”. 2009 Oct 1;10 Suppl 10:S2.

Deborah McGuinness and Frank van Harmelen (10 February 2004). "OWL Web Ontology Language Overview". W3C Recommendation for OWL, the Web Ontology Language. World Wide Web Consortium. http://www.w3.org/TR/2004/REC-owl-features-20040210/. Retrieved 5 January 2011.

McKee, A.H., Kleckner, N. (1997) Mutations in Saccharomyces cerevisiae that block meiotic prophase chromosome metabolism and confer cell cycle arrest at pachytene identify two new meiosis-specific genes SAE1 and SAE3. *Genetics*

E. Neumann, A Life Science Semantic Web: Are We There Yet? Sci. STKE 2005, pe22 (2005).

Nigel Shadbolt, Tim Berners-Lee and Wendy Hall, “The Semantic Web Revisited” IEEE Intelligent Systems 21(3) pp. 96–101, May/June 2006.

Marc-Alexandre Nolin, Peter Ansell, François Belleau, Kingsley Idehen, Philippe Rigault, Nicole Tourigny, Paul Roe, James M Hogan, Michel Dumontier. Bio2RDF Network Of Linked Data. 2008. International Semantic Web Conference (ISWC2008): Semantic Web Challenge.

Peter F. Patel-Schneider, Patrick Hayes and Ian Horrocks (10 February 2004). "OWL Web Ontology Language Semantics and Abstract Syntax Section 5. RDF-Compatible Model-Theoretic Semantics". W3C Recommendation for OWL, the Web Ontology Language. World Wide Web Consortium. http://www.w3.org/TR/owl-semantics/rdfs.html. Retrieved 5 January 2011

K. Okajima, L.G. Korotchkina, C. Prasad, T. Rupar, J.A. Phillips III, C. Ficicioglu, J. Hertecant, M.S. Patel and D.S. Kerr (2008) “Mutations of the E1β subunit gene (*PDHB*) in four families with pyruvate dehydrogenase deficiency “.Molecular Genetics and Metabolism Volume 93, Issue 4, April 2008, Pages 371-380

Samuel Croset, Christoph Grabmüller1, Chen Li1, Silvestras Kavaliauskas1, Dietrich Rebholz-Schuhmann1. (2010) "The CALBC RDF Triple store:retrieval over large literature content".

Samwald,M., Stenzhorn,H. (2010) "Establishing a distributed system for the simple representation and integration of diverse scientific assertions". J Biomed Semantics. 1 Suppl 1:S5.

Samwald,M., Chen,H., Ruttenberg,A., Lim,E., Marenco,L., Miller,P., Shepherd,G., Cheung,KH. (2010) "Semantic SenseLab: Implementing the vision of the Semantic Web in neuroscience". Artif Intell Med. 2010 Jan;48(1):21-8. Epub 2009 Dec 16.

Shen, E.C., Stage-Zimmermann, T., Chui, P., Silver, P.A. (2000)The yeast mRNA-binding protein Npl3p interacts with the cap-binding complex. J. Biol. Chem.

Associated websites

Jena – A Semantic Web Framework for Java, 2009 :
http://jena.sourceforge.net/ (consulted 5.012011)

Tutorial Semantic Web 2009 :
http://www.Linkeddatatools.com/ (consulted 5.012011)

The Semantic Web Activity 2010 :
http://www.w3.org/ (consulted 5.012011)

The Gene Ontology 2006 :
http://www.geneontology.org/GO.format.obo-1_2.shtml (consulted 5.012011)

Parser BioJava 1 July 2008:

http://www.biojava.org/wiki/BioJava:CookBook:OBO:parse (consulted 5.012011)

W3C RDF Vocabulary Description Language 1.0: RDF Schema, 10 February 2004: http://www.w3.org/TR/rdf-schema/ (consulted 5.012011)

W3C OWL Web Ontology Language Overview, 10 February 2004: http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1.2 (consulted 5.012011)

W3C Resource Description Framework (RDF): Concepts and Abstract Syntax. 10 February 2004 http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#s1.2 (consulted 5.012011)

Appendix

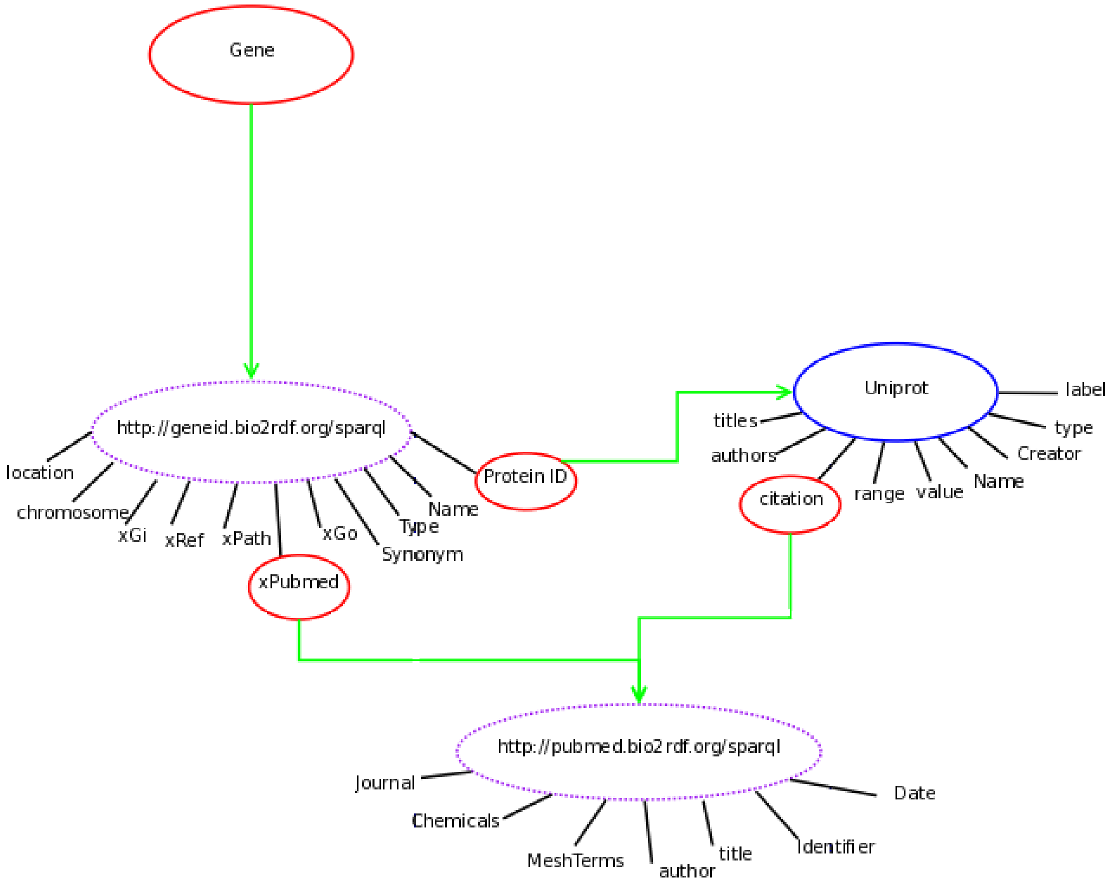


Figure 7 : FlowChart of the process from LinkInRDF. First a geneID is given. From then information about the gene are retrieve on geneID SPARQL endpoint. The protein ID is selected to retrieve information about the protein from Uniprot database. Finally Pubmed reference from the gene model and citation from the protein model are combine and use to retrieve information about the literature from pubmed SPARQL endpoint.

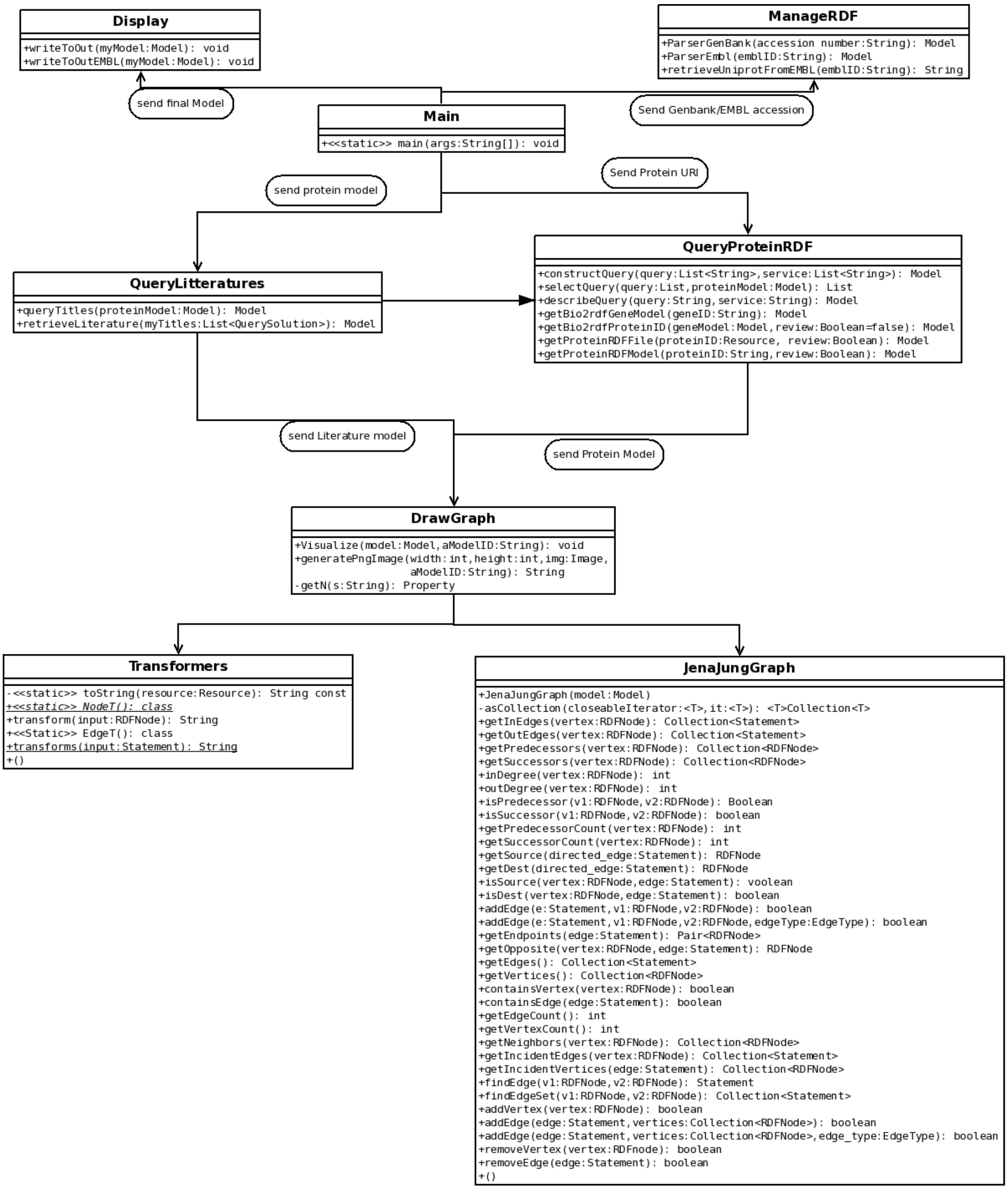

Display
+writeToOut(myModel:Model): void
+writeToOutEMBL(myModel:Model): void
ManageRDF
+ParserGenBank(accession number:String): Model
+ParserEmbl(emblID:String): Model
+retrieveUniprotFromEMBL(emblID:String): String
send final Model
Main
+<<static>> main(args:String[]): void
Send Genbank/EMBL accession
Send Protein URI
send protein model
QueryProteinRDF
+constructQuery(query:List<String>,service:List<String>): Model
+selectQuery(query:List,proteinModel:Model): List
+describeQuery(query:String,service:String): Model
+getBio2rdfGeneModel(geneID:String): Model
+getBio2rdfProteinID(geneModel:Model,review:Boolean=false): Model
+getProteinRDFFile(proteinID:Resource, review:Boolean): Model
+getProteinRDFModel(proteinID:String,review:Boolean): Model
QueryLitteratures
+queryTitles(proteinModel:Model): Model
+retrieveLiterature(myTitles:List<QuerySolution>): Model
send Literature model
send Protein Model
DrawGraph
+Visualize(model:Model,aModelID:String): void
+generatePngImage(width:int,height:int,img:Image,
aModelID:String): String
-getN(s:String): Property
Transformers
-<<static>> toString(resource:Resource): String const
+<<static>> NodeT(): class
+transform(input:RDFNode): String
+<<Static>> EdgeT(): class
+transforms(input:Statement): String
+()
JenaJungGraph
+JenaJungGraph(model:Model)
-asCollection(closeableIterator:<T>,it:<T>): <T>Collection<T>
+getInEdges(vertex:RDFNode): Collection<Statement>
+getOutEdges(vertex:RDFNode): Collection<Statement>
+getPredecessors(vertex:RDFNode): Collection<RDFNode>
+getSuccessors(vertex:RDFNode): Collection<RDFNode>
+inDegree(vertex:RDFNode): int
+outDegree(vertex:RDFNode): int
+isPredecessor(v1:RDFNode,v2:RDFNode): Boolean
+isSuccessor(v1:RDFNode,v2:RDFNode): boolean
+getPredecessorCount(vertex:RDFNode): int
+getSuccessorCount(vertex:RDFNode): int
+getSource(directed_edge:Statement): RDFNode
+getDest(directed_edge:Statement): RDFNode
+isSource(vertex:RDFNode,edge:Statement): voolean
+isDest(vertex:RDFNode,edge:Statement): boolean
+addEdge(e:Statement,v1:RDFNode,v2:RDFNode): boolean
+addEdge(e:Statement,v1:RDFNode,v2:RDFNode,edgeType:EdgeType): boolean
+getEndpoints(edge:Statement): Pair<RDFNode>
+getOpposite(vertex:RDFNode,edge:Statement): RDFNode
+getEdges(): Collection<Statement>
+getVertices(): Collection<RDFNode>
+containsVertex(vertex:RDFNode): boolean
+containsEdge(edge:Statement): boolean
+getEdgeCount(): int
+getVertexCount(): int
+getNeighbors(vertex:RDFNode): Collection<RDFNode>
+getIncidentEdges(vertex:RDFNode): Collection<Statement>
+getIncidentVertices(edge:Statement): Collection<RDFNode>
+findEdge(v1:RDFNode,v2:RDFNode): Statement
+findEdgeSet(v1:RDFNode,v2:RDFNode): Collection<Statement>
+addVertex(vertex:RDFNode): boolean
+addEdge(edge:Statement,vertices:Collection<RDFNode>): boolean
+addEdge(edge:Statement,vertices:Collection<RDFNode>,edge_type:EdgeType): boolean
+removeVertex(vertex:RDFnode): boolean
+removeEdge(edge:Statement): boolean
+()

Figure 8: Class Diagram

Structure of the program

Command line arguments (2arguments):

First argument : use prefix “-“ + the database +” ”+ the identifier. Database can be geneID, genBank, embl or uniprot.

Second argument : use prefix “-” + rev. rev refers to reviewed for protein reviewed. The absence of the second argument would imply a default of rev = false. Which means every protein found will be collected and a graph will be drawn.

- **Class Main**
    - Method : static void main (String [] args)

Get the arguments from the command line. According to the arguments send them to the reliable classes.

- **Class ManageRDF**
    - Method : Model parserGenBank (String accesionNumber)

Get a GenBank accession number, retrieve the GenBank file, parse the file and convert it to RDF. Return the RDF model.

- Method : Model parserEMBL (String accessionNumber)

Get a EMBL accession number, retrieve the file and parse it. Send accession number to previous method for conversion. Return the gene model.

- Method : String retrieveUniprotFromEMBL (String emblID)

Get a EMBL accession number, retrieve the protein ID and return it.

- **Class QueryProteinRDF**
    - Method: Model constructQuery (List<String> queries, List<String> services)

Get a list of queries and a list of services. Proceed the queries according to their respective services and return a RDF model.

- Method: List selectQuery (String query , Model model)

Get a query and a service. Proceed the query and return a list containing results from the query.

- Method: Model describeQuery (String query, String service)

Get a query and a service. Proceed the query and return a RDF model.

- Method: Model getBio2RDFGeneModel (String geneID)

Get a geneID identifier. Retrieve the gene model and draw a graph. Return the model.

- Method: List<QuerySolution> getBio2RDFProteinID (Model geneModel)

Get the gene model. Retrieve attribute xPath (URIs) from the model and return the list of xPath.

- Method: Model getProteinRDFFile (Resource proteinID, Boolean review)

Get a resource that correspond to a protein ID and a Boolean. Retrieve the protein ID. Call below method. Return a protein model.

- Method : Model getProteinRDFModel (String proteinID, Boolean review)

Get a String (protein identifier) and a Boolean. Retrive the Uniprot protein RDF model, check the boolean and review attribute. Return the protein RDF model.

- **Class QueryLiterature**
  - Method: Model queryTitles (Model proteinModel)

Get a RDF protein model. Query the literature titles available. Call below method. Return an RDF model from the literature.

- Method: Model retrieveliteratureInfo (List<QuerySolution> titles)

Get a list of literature's titles. Query Pubmed endpoint to retrieve RDF model from the literature. Return Literature RDF model.

- **Class DrawGraph**
  - Method: void visualize (Model model, String modelID)

Get a model and an identifier that correspond to the model. Create parameters for the graph. Call below method.

  - Method: String generatePngImage (int width, int height, Image img ,String modelID)

Get parameters from the previous methods. Generate the picture available in the folder from the project. Return the picture.

- **Class Display**
  - Method: void writeToOut(Model model)

Get a model. Query the model to retrieve reliable information. Display the information in HTML format.

  - Method: void writeToOutEMBL (Model model)

- *Get a model issue from a EMBL or GenBank accession number. Query the model to retrieve reliable information. Display the information in HTML format.*