Chapter #

APPLYING OWL REASONING TO GENOMIC DATA

AUTHORS: Katy Wolstencroft[1], Robert Stevens[1] and Volker Haarslev[2]
Affiliation 1. School of Computer science, University of Manchester, Oxford Road, Manchester, United Kingdom, M13 9PL
2. Department of Computer Science and Software Engineering, Concordia University, Montreal, Quebec, Canada

Abstract: The core part of the Web Ontology Language (OWL) is based on Description Logic (DL) theory, which has been investigated for more than 25 years. OWL reasoning systems offer various DL-based inference services such as (i) checking class descriptions for consistency and automatically organizing them into classification hierarchies, (ii) checking descriptions about individuals for consistency and recognizing individuals as instances of class descriptions. These services can therefore be utilized in a variety of application domains concerned with representation of and reasoning about knowledge, for example, in biological sciences. Classification is an integral part of all biological sciences, including the new discipline of genomics. Biologists not only wish to build complex descriptions of the categories of biological molecules, but also to classify instances of new molecules against these class level descriptions. In this chapter we introduce to the non-expert reader the basics of OWL DL and its related reasoning patterns such as classification. We use a case study of building an ontology of a protein family and then classifying all members of that family from a genome using DL technology. We show how a technically straight-forward use of these technologies can have far-reaching effects in genomic science.

Key words: Protein Classification, OWL DL, Reasoning, Reasoning Patterns, Protein Phosphatases.

1. INTRODUCTION

In this Chapter, we look at an example where the strict semantics of OWL-DL, when used to define the classes of a protein family, can be used to great effect in biological data analysis. Conceptually, this is a straight-forward example of knowledge of a domain being used in computational form. We first give the biological context, problem and motivation for this work. We then look at the analysis technique and in the second half of the chapter move from the biological aspects to the description logic aspects of this work. One simple message is that OWL-DL has been used to make biological discoveries. We also show that a great deal can be done with only using a subset of OWL-DL's expressivity.

1.1 Background

Bioinformatics encompasses computational and mathematical techniques for analysing, managing and storing biological data. It is a relatively new discipline in science which has grown as a direct result of advances in technologies and techniques in biochemistry, molecular biology and genetics [1]. The development of new techniques in DNA and protein sequencing, for example, has lead to an exponential growth in the production of biological sequence data. In order to make use of this data, however, it needed to be analysed, categorised and recorded in a systematic way.

The majority of bioinformatics data was, and continues to be, published in public repositories, which are distributed throughout the world. These resources provide a rich source of research material for the bioinformatician. Algorithms for searching, predicting, or classifying data in these repositories have been developed to help with the task of extracting and integrating the biological information between them. The data repositories and analysis tools together provide a 'toolkit' for the bioinformatician.

Producing algorithms to analyze sequence data is only a fraction of the problem faced by bioinformaticians. Managing data and annotating it with the knowledge previously derived from experiments in laboratories or *in silico* are also important considerations [2]. For example, PubMed [3], the digital archive of life sciences journal literature, contains in excess of 15 million citations. Each citation represents the collection of one or more fragments of biological knowledge. Associating knowledge from this resource with the genes and proteins relating to it in biological sequence resources is an enormous task [4], [5]. The scale of the problem, the complexity of the data, and the inevitable and constant revision of knowledge over time makes this a grand challenge in bioinformatics.

Molecular biology aims to help better understand the functions and processes that occur in living systems by starting from the basic building blocks of life. DNA encodes the genetic information of life, which means DNA contains all the information, in the form of genes, a cell needs to replicate and function. Genes are described as the basic unit of heredity and almost always produce a functional product, a protein. Proteins are complex molecules that carry out the majority of biological functions within a cell. Understanding what genes and proteins are present helps scientists understand how living organisms work.

In bioinformatics, genes and proteins are generally represented as sequences. DNA is made up of a series of nucleic acid molecules, adenine (a), guanine (g), cytosine (c) and thymine (t). The order of these four molecules encodes the sequence of the resulting protein products. Proteins are made up of amino acid molecules. There are twenty different amino acids used within cells.

A collection of three nucleic acids, encodes an amino acid. Some amino acids have more than one nucleic acid code (known as a codon), some have only one. Figure 1 shows the relationships between nucleic acids and amino acids.

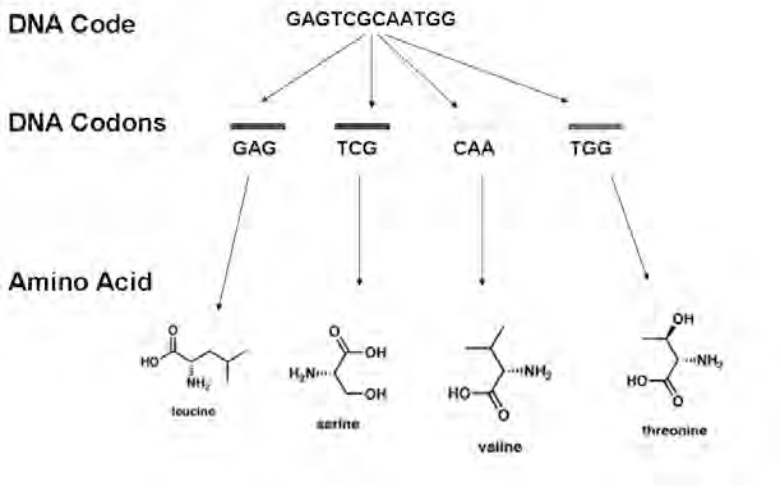

Figure #-1. The relationship between DNA and protein sequences. Each three letter DNA codon encodes an amino acid. Sequences of amino acids form proteins.

As can be seen in figure 1, amino acids are complex molecules. Each has a different shape and set of physical properties. For example, some have a positive or negative charge and some are hydrophobic (e.g leucine). The sequence of amino acids in a protein therefore helps determine its final three-dimensional structure. This structure in turn helps determine the chemical and physical interactions of this protein within the cell. These facts mean that analysing the sequences of proteins and genes can tell the scientist a lot about the functions of the gene products *in vivo*. If the function of a protein is conserved through evolution, this means that sequence features can also be conserved. Consequently, comparing protein and gene sequences across different species allows inferences to be made about the functions of unknown or uncharacterised proteins and genes by similarity measures to better characterised and experimentally verified protein and gene functions. This is true at the level of individual sequences and also at the level of the whole genome, the entire collection of genes. By organising and classifying genes and proteins into functional groups (families), one can compare typical functional properties across different species.

This process of classification is important, but knowledge-intensive. There are many tools and resources available to help scientists assess the similarity between biological sequences, but the tools themselves do not perform the classification step. The results obtained from similarity search tools must be analysed by scientists, and this is the rate-limiting step. The pace at which data is produced far outstrips the pace at which it is analysed and classified.

In this chapter we discuss a method for automated classification that could reduce this bottleneck. We use an ontology to capture the knowledge that a human uses to recognize types of proteins from a particular protein family. By combining this knowledge with existing tools for detecting sequence features we are able to perform a thorough, systematic analysis of a protein family and how it differs between organisms, illustrating the utility of such a method in comparative genomics. This methodology does not develop or test new bioinformatics algorithms for detecting sequence features. Instead, it provides a novel method for interpreting the results of these techniques and algorithms to perform automatic protein classification.

1.2 Analysing protein domains

Approaches to analysing the large data sets produced in genome sequencing projects have ranged from human expert analysis, which is

considered to be the 'gold-standard,' to the simple automation of tools such as BLAST [6] and Interpro [7].

Analysis of proteins by experts enables classification to be driven by expert knowledge, which draws on the collective knowledge in the community. Experts can interpret the information from the biological literature and apply it to the observed results. This is, however, a time-consuming process and many academic institutions cannot support large teams of bioinformaticians required for such activities. The alternative choice is automated classification. This tends to be quicker, but the level of detail is often reduced, which means proteins are often only classified into broad functional classes.

For example, taking the top BLAST hit as a basis for classification of an unknown protein can infer relationships between the unknown protein and previously characterized proteins, allowing the new sequences to be annotated as 'similar to' a characterized protein. This has value, but it also has intrinsic problems. One of the largest problems is that the databases of characterized sequences contain sequences with differing degrees of annotation. Some sequences were experimentally characterized in laboratory experiments and annotated by human experts, whilst others were already classified using similar automated methods, and so are annotated as 'similar to' another protein already [8]. Annotating new sequences against these proteins has great potential for propagating errors if the original assignment is incorrect. Also, the annotations do not provide information regarding the experimental details of the similarity assignment, i.e. which version of BLAST was used, with what parameters, and what was the resulting similarity score. Without this data provenance, the annotation should not be re-used for further comparisons.

Another problem with similarity methods is that both full length and truncated sequences can be contained within the same BLAST indexed database. If the unknown sequence shows high similarity to a characterized, truncated sequence, there is no method for determining if the unknown sequence is also truncated, or if the unknown sequence simply shows high similarity with the known sequence for part of its length.

Like similarity measures, using automated classification methods on protein motif and domain matching techniques (discussed further in section 1.3) can also be a valuable 'first pass' for large scale annotation, but it too can be limited at a detailed level. These methods report the presence of functional domains, but it is the unique combinations of these domains that determine the protein function. Human experts are still required to interpret these combinations of functional domains in order to provide functional annotation.

In both automated similarity assignment and protein motif detection, there is a danger of under or over annotation. Proteins can either be classified at a level that is too general to provide useful inferences from related proteins, or proteins can be classified beyond the evidence that can be derived from sequence data, inferring properties and relationships that are incorrect. Both cases propagate errors, demonstrating the limitations of current automated methods.

1.3 Classifying proteins into families

Many proteins are assemblies of sequence motifs and domains. Each domain or motif might have a separate function within the protein, such as catalysis or regulation, but it is the overall composition that gives each protein its specific function. Recognition of domain and motif composition is a powerful bioinformatics technique which can be employed in the classification of proteins.

There are many tools dedicated to discovering protein features and functional domains and motifs (hereafter referred to as p-domains). Examples include, PROSITE [9] and Pfam [10]. These tools each employ different methods of analysis to detect sequence features and p-domains, for example, PROSITE uses simple pattern-matching to single motifs, whereas Pfam uses hidden markov models (HMMs). Researchers routinely use many different p-domain detection tools together to build up a consensus of results. To facilitate this process, InterPro encapsulates many of these tools, and allows scientists to perform analyses over all of them with one query submission to the tool InterproScan.

Interpro currently enables the querying of sixteen different algorithms and tools and in this work, we define p-domains as any sequence features identified by tools within the Interpro Collective.

InterproScan provides a mechanism for the automation of p-domain analysis, but not for the interpretation of that analysis. It reports the presence of p-domains, but not the consequences for family or subfamily membership. In certain cases, the presence of a p-domain is diagnostic for membership of a particular protein family; for example, the G-protein coupled receptor like domain in G-protein receptors. However, further classification into subfamilies is not usually possible without further interpretation over the results of p-domain analyses. Previously, this has not been attempted. In this method we have replaced this human intervention step with further automation which uses knowledge captured in an ontology.

Ontologies provide a technology for capturing and using human understanding of a domain within computer applications [11]. The use of

ontologies to capture human knowledge in biology and annotate data accordingly is becoming well established. For example, the Gene Ontology describes all gene products common to eukaryotic genomes. Individual proteins are annotated with terms from this ontology to promote a common understanding across the community about their function(s) [12].

Other uses of ontologies, however, are more unusual in biology. For example, the use of reasoning over formal ontologies and their instances, enabling data *interpretation* has not been explored. In this study, we present a new method which uses ontological reasoning for data interpretation and illustrates the advantages of such an approach. This method allows the combination of advantages gained from human expert analysis with the benefits of the increased speed in automated annotation methods. We use a protein family-specific ontology, defined in the OWL language [13], to capture the human understanding of a protein family together with p-domain analyses, using InterproScan, to automate the analysis of each protein in that family.

In this chapter, we use the protein phosphatase family as a case study. The method we have developed enables the analysis of all protein phosphatases in a genome. We find that in classifying proteins, our system can perform at least as well as a human expert. In this context, the biology of protein phosphatases is not important. They provide a useful case study for the use of ontology technology to provide automated recognition over identified protein sequence features. The provision of this extra step and the consequent biological findings are important; the fact we used protein phosphatases is not so important.

1.4 The Protein Phosphatase family

Phosphorylation and dephosphorylation reactions form important mechanisms of control and communication in almost all cellular processes including, metabolism, homeostasis, cell signaling, transport, muscle contraction and cell growth. These reactions allow the cell to respond to external stimuli, such as hormones and growth factors [14], as well as responding to cellular stress and cytokines [15].

The enzymes primarily involved in catalyzing phosphorylation events can be divided into two families, protein kinases and protein phosphatases. Kinases are involved in the phosphorylation of the amino acids serine, threonine and tyrosine [16] and phosphatases are involved in the removal of phosphates from these residues. It is the careful balance between these two opposing reactions that controls the phosphorylation state of a multitude of biological molecules and ultimately controls almost all biological processes [17].

Protein phosphatases all perform the same chemical reaction in the cell, the removal of a phosphate group, but the phosphatases are diverse in biological function and catalytic activity. They can be broadly divided into two subfamilies, the serine/threonine phosphatases and the tyrosine phosphatases. Recent reviews on the protein phosphatase family ([18], [19] and [20]) focus on either one or the other. There have been extensive studies into the characterisation of each in the human genome. Whilst the distinction between the broad classes of serine/threonine and tyrosine subfamilies is often easy to determine, some closely related proteins have little difference between them. The difficulty of fine-grained classification is therefore increased with the subtlety of the differences between closely related proteins, which can perform different biological functions. In figure 2 we show the differences in domain architecture of one subfamily of phosphatases, the receptor tyrosine phosphatases.

Protein phosphatases are popular targets for medical and pharmaceutical research as they have been associated with a number of serious human diseases, such as cancers, neurodegenerative conditions and, most recently, diabetes [21], [22], [23] and [24].

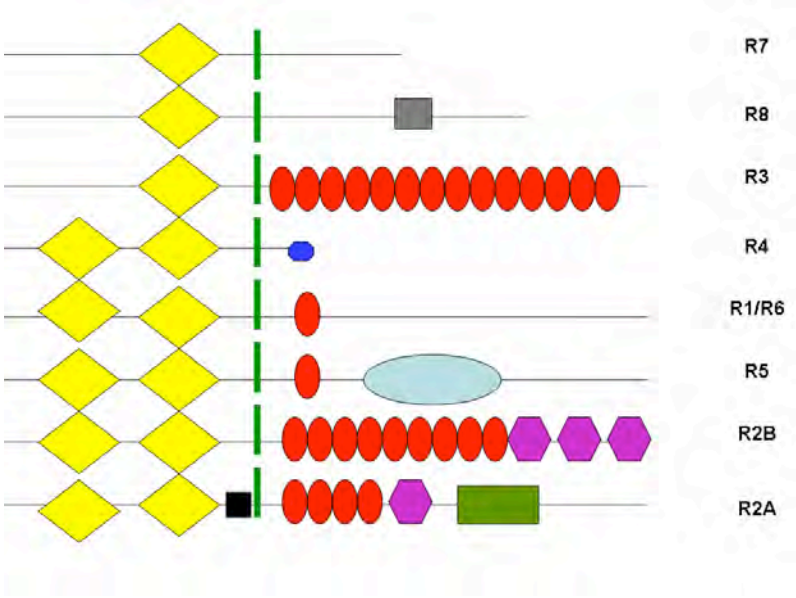

Figure #-2. The differences in domain architecture of the receptor tyrosine phosphatase subfamily. Yellow = phosphatase catalytic domain. Green bar = transmemebrane region,

purple = immunoglobulin domain, red circle = fibronectin domain, green square = MAM domain, light blue oval = carbonic anhydrase domain, grey square = adhesion recognition site, dark blue = glycosylation and black = cadherin-like domain

2. OWL DL REASONING PRINCIPLES

The case study presented here uses OWL DL reasoning to solve a problem in analyzing protein sequence data. We can computationally find the sequence features in a given protein sequence. The problem is that we need to computationally recognize the consequences of the presence of a particular set of protein features. This is bioinformatics knowledge and exactly the kind of knowledge that can be captured in an OWL ontology. Before presenting the results of using this ontology, we provide an abstract view on the underlying algorithmic principles and computational tools employed. The goal is to enable computational biologists to transfer the applied techniques to their domain of interest and apply them to their problem solving needs. In the following we assume some familiarity with the ideas of OWL DL but we present a short review of the main notions of OWL DL in order to keep this chapter self-contained.

The core part of OWL, called OWL DL and its subset OWL Lite, is based on Description Logic (DL) theory [25] , which has been investigated for more than 25 years. Description logics can be viewed as a family of knowledge representation languages, primarily intended to specify knowledge of any kind in a formal way. This formal specification provides the basis for OWL DL reasoning tools that process OWL DL knowledge bases (KBs), or ontologies, and offer various inference services. An OWL DL reasoner can be considered as a domain-independent problem solving engine that can be utilized in arbitrary application domains provided the domain knowledge is specified (or encoded) in OWL DL. However, OWL DL reasoners are not general problem solvers in the sense of "Do What I Mean". Their inference services are grounded on the formal properties of knowledge representation languages such as OWL DL. So, how can one make a meaningful use of such reasoning services? To do so we have to map the domain-specific problem solving process to an inference service supported by an OWL DL reasoner. In the following we explain this process by discussing OWL DL and the reasoning services provided by OWL DL reasoners.

2.1 Basic Reasoning Services

First, we briefly review the language elements of OWL DL (OWL Specification 2004). They mainly consist of anonymous (unnamed) or named classes, properties, and their restrictions and individuals. Classes can be considered as descriptions of common characteristics of sets of individuals. Class descriptions can be either complete, i.e., they specify sufficient conditions for class membership, or partial i.e., they specify only necessary conditions for class membership. Properties are divided into object and data type properties. Object properties can be used to express binary relationships between sets of individuals, while data type properties can be viewed as binary mappings from individuals to data values. Individuals are members (otherwise known as instances) of classes and can be used to form enumerated classes. Using these elements one can compose class descriptions consisting of all language elements combined by set-based operators such as intersection-of, union-of, and complement-of. Properties are used in class descriptions by listing restrictions on the values of those properties such as type, specific value, and cardinality (number of values). These restrictions characterize instances of classes more precisely. Statements about domain knowledge can be formed by combining these elements and are expressed as axioms describing (i) that the set of instances in two classes are subsets of one another, equivalent, or disjoint, (ii) characteristics of properties such as transitivity or that the values of one property are a subset of another one, (iii) class membership and property values of individuals, and (iv) similarity and difference between individuals.

Given these language elements the following types of reasoning services are typically supported by OWL DL reasoners. Classes can be checked for consistency, (also sometimes called satisfiability) i.e., is a class description meaningful at all and can it have at least one instance. Another service consists of computing inferred subset or subclass relationships, also known as subsumption relationships, i.e., all individuals that are instances of a subclass must be also instances of its superclasses. It is important to note that a subsumption relationship is only induced by the corresponding sub- and superclass descriptions. Based on class subsumption all named classes of a KB can be automatically organized in a taxonomy or subsumption hierarchy. This process is also often referred to as the classification of a KB. Analogous to subsumption, equivalence or disjointness between classes can be inferred too. The class satisfiability checking and classification process usually provides important feedback to designers of KBs because they might learn about unsatisfiable class descriptions, which are usually considered as design errors, or inferred and possibly unexpected subsumption relationships, which

might match or violate principles of the application domain. Again, the latter case would correspond to a design error in the KB, where some class descriptions incorrectly or imprecisely model the application domain.

The second class of supported inference services is concerned with individual descriptions. Descriptions of individuals can be checked for consistency, i.e., whether they comply with the class and property statements declared in a KB. The case that individual descriptions are recognized as inconsistent corresponds either to an application domain modeling error or indicates a violation of the domain principles encoded in the KB. The individual descriptions consistency check is a prerequisite for the following other individual inference services. The most basic one is a test for class membership, i.e., is a given individual an instance of one or more classes declared in a KB. The services can be even more refined because the reasoner can automatically determine the most specific classes that instantiate a given individual. It is important to note that class membership for individuals can usually only automatically be recognized if the class description is complete. The membership of an individual in the superclasses of a given class is immediately implied due to the transitivity of the subsumption relationship. If this service, to determine the most specific classes of an individual, is applied to all individuals declared in a KB, it is traditionally referred to as realization of a KB.

2.2 Reasoning Paradigms

Individual descriptions in a KB usually rely heavily on the classes and properties declared in a KB, although OWL also allows users to introduce names that have not been declared yet. The structure of OWL DL statements about individuals and their relationships with other individuals or values can be compared with relational data descriptions known from relational databases (DBs). The information about individuals resembles, to some extent, a simple database schema, where a one-column table exists for each named class, containing all individual names that are instances of this class, and a two-column table for each property, containing pairs of individuals (object property) or values associated with individuals (datatype property) known to be related by the property. Occurrence in a table is based on either explicit assertions or implicit OWL DL reasoning results. In contrast to standard DBs it is assumed that the information in these tables is incomplete. This principle is called an open-world assumption in contrast to a closed-world assumption from DBs, where the non-occurrence of information is interpreted as "this information does not hold". The open-world assumption

is also closely related to another basic reasoning principle for OWL DL, the monotonicity of reasoning. This means that knowledge derived by inferences can only extend the already known knowledge. It cannot contradict known knowledge and it cannot cause the retraction of known knowledge. These principles could be either considered as advantageous or disadvantageous. In the context of the WWW it makes sense to consider information as incomplete. However, the information about the state of a domain is usually also evolving in a non-monotonic way because previously known facts might not hold anymore. It is important to note that the monotonicity of reasoning holds for a given version of a KB but different versions of KBs might evolve in a non-monotonic way. However, reasoning about such a change between versions is beyond the state of the art of current OWL DL reasoners.

2.3 Querying Individual Descriptions

The open-world assumption also affects how queries about individual descriptions are answered. Besides the basic inference services for individual descriptions some OWL DL reasoners also support query answering with functionality similar to DBs. Again, query answering about OWL DL individual descriptions might involve reasoning in contrast to standard DBs, where query answering mostly involves table look-ups. One of the currently most advanced query languages [26], called nRQL (New RacerPro Query Language), is implemented in the OWL DL reasoner Racer [27] and its successor RacerPro (Racer Systems 2006). The nRQL language supports query answering about OWL DL individual descriptions. The supported query language elements allow one to retrieve all individuals that are instances of a class, all individual pairs that are elements of object properties, and all individual-value pairs that are elements of data type properties and optionally satisfy specified constraints. All these elements can be combined to form complex queries with query operators such as intersection, union, complement, and projection. These operators are similar to standard relational DB operators. The DB join operator is implicitly available in nRQL through the use of query variables and the intersection operator. Moreover, nRQL supports closed-world reasoning over named individuals (sometimes also called negation as failure), which is especially useful for measuring the degree of completeness of modeling the domain of discourse in a KB. The nRQL query language is oriented towards computer scientists and uses a Lisp-like syntax. In order to facilitate the use of nRQL by scientists from other domains the OntoIQ tool has been developed [28]. It offers users a graphical and easy-to-use user interface to compose, execute,

and store nRQL queries. Queries can be also composed with the help of predefined query patterns. nRQL and OntoIQ[1] have been successfully used in the context of a fungal enzyme project [29], [30].

3. POTENTIAL APPLICATIONS OF REASONING PATTERNS

In the previous section we reviewed main OWL DL language elements and discussed OWL DL reasoning principles and services. In this section we come back to the question "how can one make a meaningful use of such reasoning services?" In general, there exist two possible approaches. The first one is applicable if the above-mentioned reasoning services can be directly used to solve the domain-specific application problems. This is usually possible if the necessary domain knowledge can be directly encoded into OWL DL. For instance, this is the case with the study presented in this chapter. The second and more difficult approach requires the translation of the knowledge about the problem domain into OWL DL in such a way as to use the reasoning services as general problem solver. For instance, one might encode the structure of a labyrinth into an OWL DL KB and then use queries to find a path from a certain point within the labyrinth to its exit.

3.1 Classification Pattern

The classification pattern makes a direct use of the classification mechanisms implemented in OWL DL reasoners. In order to apply this pattern the domain knowledge needs to be encoded as mostly complete class descriptions specifying meaningful sets of entities in the application domain. The solution to an application problem would consist of the inferred class taxonomy, i.e., a problem is solved if selected classes are subsumed by other classes or, in other words, the subsumers of classes describe the problem solution. A biological example of this would be that all protein phosphatases should be subsumed by the class enzyme, and all enzymes should be subsumed by the class protein.

3.2 Realization Pattern

This pattern builds on top of the classification pattern. Besides the class taxonomy useful knowledge is also encoded in individual descriptions. The

[1] OntoIQ download page: http://www.cs.concordia.ca/FungalWeb/Downloads.html

problem solution results from computing for selected individuals their most specific instantiators, i.e., the most specific (complete) classes that instantiate these individuals. This pattern usually also requires that the envisioned instantiators have complete descriptions. This is the pattern that was successfully employed in the case study reported in this chapter. Protein phosphatase class descriptions were constructed from the types and numbers of p-domains they contained. By analyzing the p-domains in the individuals, and comparing them to the class descriptions, the most specific class instantiating an individual could be identified.

3.3 Query Pattern

The query pattern can be used independently of the previous patterns or in addition to the realization pattern. This pattern partially views individual descriptions as stored in a deductive DB and query results are interpreted as solutions for the application problem. A typical use of the query pattern would be to add functionality to the realization pattern by allowing more complex query conditions that can be utilized to encode problem solutions. For instance, arbitrary queries allow one to query (possibly cyclic) individual graph structures where the edges of a graph consist of properties holding between pairs of individuals. The realization pattern can be often considered as queries enforcing individual tree structures only. Both query pattern variants might collapse into one pattern if a query involves enumerated classes. The successful use of the query pattern is reported elsewhere [29],[30].

4. USING OWL DL IN BIOLOGICAL CLASSIFICATION

The previous section introduced OWL DL, the notion of reasoning, and some common reasoning patterns. This section details the practical application of these technologies to the biological case study, and goes on to discuss the implications of this for the biological community.

4.1 The Ontology Classification Method

This study combined automated reasoning techniques with traditional bioinformatics sequence analysis techniques to automatically extract and

classify the set of protein phosphatases from an organism. Figure 3 shows the components in our protein classification experiment.

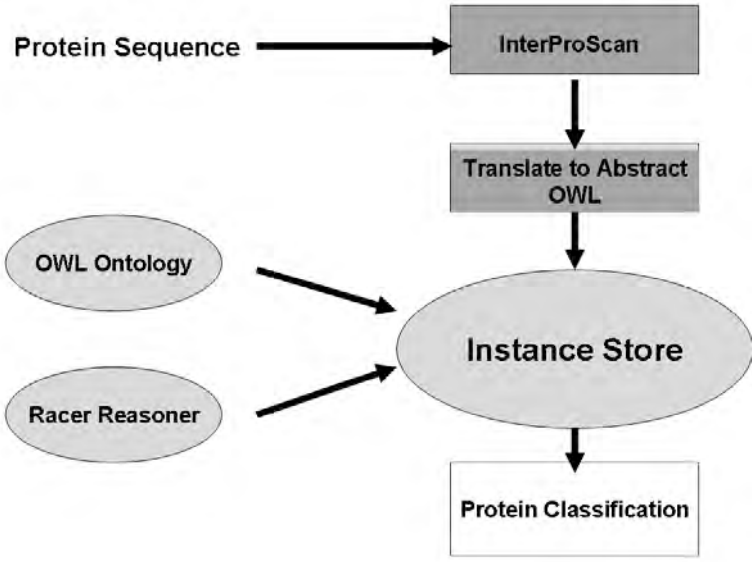

Figure #-3. The Architecture of the ontology classification method

The method includes the following stages:

1. An OWL class-level ontology describes the protein phosphatase family and the different domain architectures for members of different subfamilies. This ontology is pre-loaded into the Instance Store.
2. Protein instance data is extracted from the protein set of a genome by first screening for diagnostic phosphatase domains and then analyzing the p-domain composition of each using InterproScan.
3. The p-domain compositions are then translated into OWL descriptions and compared to the OWL definitions for protein family classes using the Instance Store which, in turn, uses a Description Logic reasoner, Racer, to classify each instance. For every protein sequence, it returns the most specific classes from the ontology that this protein could be found to be an instance of.

4.2 The Ontology

All the data used for developing the phosphatase family ontology was extracted from peer-reviewed literature from protein phosphatase experts. The human protein phosphatases have been well characterized experimentally, and detailed reviews of the classification and family composition are available [18],[19]. These reviews represent the current community knowledge of the relevant biology. If, in the future, new subfamilies are discovered, the ontology can easily be changed to reflect these changes in knowledge.

The differences between phosphatase subfamilies can be expressed by the differences of their p-domain compositions. These p-domain architectures represent 'rules' for protein subfamily membership, and these rules can be expressed as class definitions in an OWL-DL ontology. The use of an ontology to capture the understanding of p-domain composition enables the automation of the final analysis and classification step which had previously required human intervention, thus allowing for full automation of the complete process.

More precisely, for each class of phosphatase, the ontology contains a (necessary and sufficient) definition. For this family of proteins, the definition is, in most cases, a conjunction of p-domain compositions. For example, figures 4 and 5 show two classes from the phosphatase ontology. Figure 4 shows a tyrosine receptor phosphatase, instances of which have at least one tyrosine phosphatase catalytic domain and at least one transmembrane domain. The former gives the enzyme its catalytic activity and the latter anchors the protein to a cell membrane. A specific kind of receptor tyrosine phosphatase would have other domains and these are specified in subclasses of this class. These two domains are, however, sufficient to recognize any particular protein sequence to be a member of this class. The ability of OWL to model incomplete knowledge, through its open world assumption, is very useful at this point.

```
Class ReceptorTyrosinePhosphatase Complete
        (Protein and
                (hasDomain some tyrosinePhosphataseCatalyticDomain ) and
                (hasdomain some TransmembraneDomain))
```

Figure #-4. The complete OWL class description for a receptor tyrosine phosphatase. Note the possibility that other domains may be added.

Figure 5 shows an R5 phosphatase. This has many more p-domains. They are necessary for R5 phosphatase activity and the presence of all is sufficient to recognize any sequence as a member of the class. Note that there is a closure axiom stating that these are the only kinds of domain that can be present. This is to ensure that a sequence that has the p-domain architecture shown in Figure 5 plus additional p-domains will not be recognized as an R5 phosphatase. For example, the LAR protein (leukocyte antigen related protein, accession number P10586) contains two tyrosine phosphatase catalytic p-domains, one transmembrane p-domain, nine fibronectin p-domains and three immunoglobulin p-domains. The tyrosine phosphatase catalytic p-domains and the transmembrane p-domain are sufficient for the protein to belong to the receptor tyrosine phosphatase class, but the extra immunoglobulin p-domains and the lack of a carbonic anhydrase p-domain means that it cannot belong to the R5 phosphatase class. This protein is another type of receptor tyrosine phosphatase. From figure 2 we can deduce it is an R2B.

```
Class R5Phosphatase Complete
      (Protein and
            (hasDomain two tyrosinePhosphataseCatalyticDomain ) and
            (hasdomain some TransmembraneDomain) and
            (hasDomain some fibronectinDomain) and
                  (hasDoman some carbonicAnhydraseDomain) and
                        hasDomain only
                  (TyrosinePhosphataseCatalyticDomain and
                  TransmebraneDomain and fibronectinDomain and
                  carbonicAnhydraseDomain))
```

Figure #-5. A complete description of an R5 phosphatase. Note the closure axiom restricting the kinds of domain that might appear in instances of this class.

4.3 The Instance Store

We use the Instance Store application in this study [31]. The Instance Store combines a Description Logic reasoner with a relational database. The reasoner in this case performs the task of classification; that is, from the OWL instance descriptions given, it determines the appropriate ontology class for an instance description. The relational database provides the stability, scalability and persistence necessary for this work

4.4 The Data Sets

This study focuses on protein phosphatases from two organisms, human and a pathogenic fungus, *Aspergillus fumigatus*. The human phosphatases have already been identified and extensively described in previous studies [18]. They have been carefully hand-classified by domain experts and form a control group to assess the performance of the automated classification method. The *Aspergillus* proteins have been less well characterized and the protein phosphatases in this organism required identification and extraction from the genome before classification could proceed.

Previous classification of human phosphatases by domain experts provides a substantial test-set for the ontology. If the ontology can classify the proteins as well as the human experts have, studies on new, unknown genomes can be undertaken with greater confidence. The *Aspergillus fumigatus* genome offers a unique insight into the comparison between the automated method and the manual. The *A. fumigatus* genome has been sequenced and annotation is currently underway by a team of human experts [32].

5. RESULTS

The purposes of performing the studies with the human and *A. fumigatus* sequence data differed. The human study was a proof of concept to demonstrate the automated ontology classification method could be effective, and the *A. fumigatus* study was focused on biological discovery.

For the human phosphatases, the classification of proteins obtained by the automated ontology method was compared with the human expert classification. For each subclass of protein phosphatases, the numbers of individual proteins in the human classification were compared to the number obtained from the automated method. The results were the same number of individuals for each class.

The comparison between the classifications clearly demonstrated that the performance of the automated ontology classification system was equal to that of the human annotated original. The ontology class definitions were sufficient to identify the differences between protein subfamilies and demonstrate the usability of the system on uncharacterized genomes.

An interesting result from the analysis was that, using the ontology, we were able to identify additional functional domains in two dual specificity phosphatases, presenting the opportunity to refine the classification of the subfamily into further subtypes.

Alonso et al [18], describe the 'atypical' dual specificity phosphatases as being divided into seven subtypes. The largest of these have the same p-domain architecture; they contain tyrosine phosphatase and dual specificity catalytic p-domains alone. However, several proteins have additional functional domains that have been shown to confer functional specificity [33]. Classifying the proteins using the ontology highlighted more of these 'extra' p-domains. For example, the dual specificity phosphatase 10 protein (DUS10, Uniprot accession: Q9Y6W6) contains a disintegrin domain. The UniProt record reflects this, but the domain does not appear in any phosphatase characterization/classification studies. The domain architecture of DUS10 is conserved in other species (figure 6), which suggests a specific function for the domain, but current experimental evidence does not explain what this might be.

Figure #-6. The domain architecture of the dual specificity phosphatase 10 protein across different organisms

The results of the classification of phosphatases for the *A. fumigatus* genome were more interesting from a biological perspective.

The *A. fumigatus* genome has been partially annotated. It has been sequenced, and is being annotated by human experts. Therefore, the protein data currently consists of both predicted and known proteins. The predicted

proteins may contain descriptions based upon automated similarity searches, producing entries termed 'hypothetical' or 'putative', but their annotation is limited.

Using the ontology system to classify the phosphatases allowed a comparison between the proteins already annotated and those with partial annotation from similarity *searching*. The classification also enabled a comparison between the protein phosphatases in the human and *A. fumigatus* genomes. Figure 7 shows the differences in protein family composition.

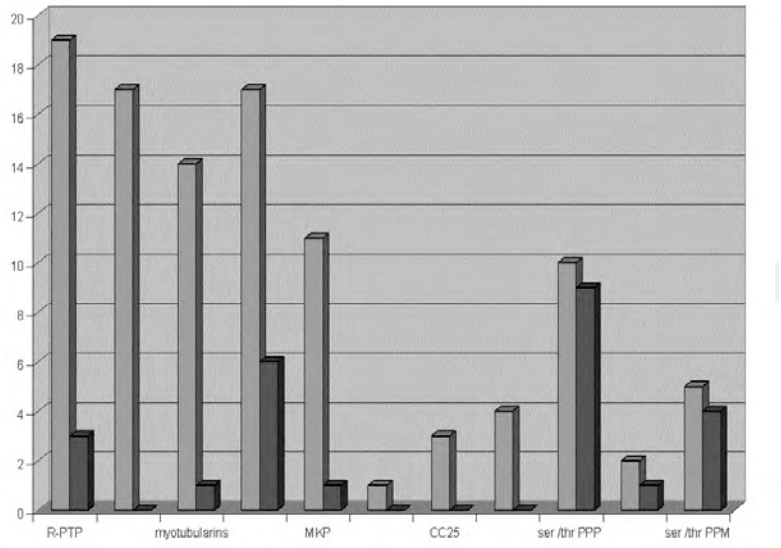

Figure #-7. The number of protein phosphatases the in human and *A.fumigatus* genomes. Human proteins are shown in pale grey, *A.fumigatus* in dark grey. These numbers represent the higher level classes of phosphatase. For example, the R5 phosphatase from figure 4 is a subclass of receptor tyrosine phosphatase, and so is a child of the R-PTP class.

In the case of the *A.fumigatus* proteins, the most interesting results were proteins that did not fit into any of the defined subfamily classes. These proteins represented differences between the human and *A.fumigatus* protein families and therefore potential differences in metabolic pathways. Since *A. fumigatus* is pathogenic to humans, these differences are important avenues of investigation for potential drug targets. The most interesting discovery in the *A. fumigatus* data set was the identification of a novel type of calcineurin phosphatase. Calcineurin is well conserved throughout evolution and

performs the same function in all organisms. However, in *A. fumigatus*, it contains an extra functional domain. The ontology classification method highlighted this difference by failing to classify the protein into any of the defined subfamily classes. Further bioinformatics analyses revealed that this extra domain also occurs in other pathogenic fungus species, but in no other organisms, suggesting a specific functional role for this extra p-domain.

6. DISCUSSION

This study demonstrates the use of the reasoning capabilities of description logic ontologies to perform protein classifications. By harnessing this technology, classifications that had previously relied on human interpretation steps could be derived from definitions of ontological classes and simple sequence analysis data alone.

Bioinformaticians perform protein classification by analyzing sequences using a series of bioinformatics tools and interpreting their results based on prior knowledge. Automating the use of the tools can be a trivial problem compared with automating the interpretation step. Users may require local implementations of tools and databases or data files for analysis, or they may perform these analyses using middleware services and workflows. However, the process of inserting and collecting data is a mechanical one and can be scripted.

Automating the biological interpretation of bioinformatics results is where the difficulty lies. An analysis of the functional domains in a given protein, using InterProScan for example, produces a list of domains. The number of each domain and, potentially, the order could also be captured, but it is the bioinformatician that infers that the presence of domains x, y and z, for example, indicates membership of a particular family. Capturing the knowledge used to perform these inferences, using defined classes in an ontology allows this final step to also be automated, increasing the speed at which proteins from a particular family can be extracted from a genome and classified. The most useful application for this technology is the analysis of protein families from genomes as and when they are sequenced, enabling fast comparisons between what is known to be present in other species. In the pharmaceutical industry in particular, this has implications for the discovery of new drug targets. Bioinformatics has been increasingly used to quicken the pace of target identification [34]. Performing *in silico* experiments on publicly available data is faster and much less expensive than many laboratory experiments. The automated classification technique enables whole protein families from many species, perhaps pathogenic and

non-pathogenic to be analyzed in unison, identifying differences that could be easily exploited when targeting pharmaceuticals.

The automated classification technique has proven to produce biologically significant results in the protein phosphatase domain and work is continuing to analyze protein phosphatases in other species, currently, the trypanosomes. The work has also been expanded to analyze different protein families, the potassium ion channels and the ADAMTS proteins.

In the future, there are plans to increase the expressivity of the protein class descriptions. As work on other protein families continues, new considerations are emerging. For example, for the protein phosphatases, the order of p-domains was not important, simply counting the number of each was sufficient to distinguish between proteins from different subfamilies. However, extending this work to other protein families would require ontology class descriptions to specify the order of p-domains.

The automated classification method presented here focuses on protein family classification using protein domain architectures; however, it is not confined to such relationships. Any analysis which uses sequence data alone can potentially use the ontology-driven method. For example, substrate recognition or protein-protein binding interactions.

The biological significance of the results obtained from the small proof of principle study in this work demonstrates that it is a powerful application of ontology reasoning, and since classification and data annotation are now slower than data production, it could have far-reaching implications on bioinformatics data analysis.

Ontology use in the bioinformatics community has grown steadily over recent years. As data and information sources reached sizes that could not be realistically managed manually, and as the need for large-scale integration and interoperation between these resources increased, computational methods were sought to help address these issues. In this work, the application of ontologies to classifying protein family information has been presented. The resources produced have demonstrated the utility of such technologies and the distinct advantages gained by their use. It is hoped that this system can be employed and exploited in future work for drug target identification and new genome annotation.

7. ACKNOWLEDGEMENTS

This work was funded by an MRC PhD studentship and myGrid e-science project, University of Manchester with the UK e-science programme EPSRC grant GR/R67743. Preliminary sequence data was obtained from The Institute for Genomic Research website at http://www.tigr.org from Dr

Jane Mabey-Gilsenan. Sequencing of *A. fumigatus* was funded by the National Institute of Allergy and Infectious Disease U01 AI 48830 to David Denning and William Nierman, the Wellcome Trust, and Fondo de Investicagiones Sanitarias.

8. REFERENCES

[1] Ouzounis, C. A., and A. Valencia. 2003. Early bioinformatics: the birth of a discipline--a personal view. Bioinformatics **19**:2176-2190

[2] Borsani, G., A. Ballabio, and S. Banfi. 1998. A practical guide to orient yourself in the labyrinth of genome databases. Hum Mol Genet **7**:1641-1648.

[3] Wheeler, D. L., T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, W. Helmberg, D. L. Kenton, O. Khovayko, D. J. Lipman, T. L. Madden, D. R. Maglott, J. Ostell, J. U. Pontius, K. D. Pruitt, G. D. Schuler, L. M. Schriml, E. Sequeira, S. T. Sherry, K. Sirotkin, G. Starchenko, T. O. Suzek, R. Tatusov, T. A. Tatusova, L. Wagner, and E. Yaschenko. 2005. Database resources of the National Center for Biotechnology Information. Nucleic Acids Res **33**:D39-45.

[4] Ouzounis, C. A., and P. D. Karp. 2002. The past, present and future of genome-wide re-annotation. Genome Biol **3**:COMMENT2001.

[5] Ge, H., A. J. Walhout, and M. Vidal. 2003. Integrating 'omic' information: a bridge between genomics and systems biology. Trends Genet **19**:551-560.

[6] Altschul, S. F., T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res **25**:3389-3402.

[7] Mulder, N. J., R. Apweiler, T. K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bradley, P. Bork, P. Bucher, L. Cerutti, R. Copley, E. Courcelle, U. Das, R. Durbin, W. Fleischmann, J. Gough, D. Haft, N. Harte, N. Hulo, D. Kahn, A. Kanapin, M. Krestyaninova, D. Lonsdale, R. Lopez, I. Letunic, M. Madera, J. Maslen, J. McDowall, A. Mitchell, A. N. Nikolskaya, S. Orchard, M. Pagni, C. P. Ponting, E. Quevillon, J. Selengut, C. J. Sigrist, V. Silventoinen, D. J. Studholme, R. Vaughan, and C. H. Wu. 2005. InterPro, progress and status in 2005. Nucleic Acids Res **33**:D201-205

[8] Gilks, W. R., B. Audit, D. De Angelis, S. Tsoka, and C. A. Ouzounis. 2002. Modeling the percolation of annotation errors in a database of protein sequences. Bioinformatics **18**:1641-1649.

[9] Hulo, N., C. J. Sigrist, V. Le Saux, P. S. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. De Castro, P. Bucher, and A. Bairoch. 2004. Recent improvements to the PROSITE database. Nucleic Acids Res **32**:D134-137.

[10] Bateman, A., L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. Sonnhammer, D. J. Studholme, C. Yeats, and S. R. Eddy. 2004. The Pfam protein families database. Nucleic Acids Res **32**:D138-141.

[11] Stevens, R., C. Goble, I. Horrocks, and S. Bechhofer. 2002a. OILing the way to machine understandable bioinformatics resources. IEEE Trans Inf Technol Biomed 6:129-134.

[12] Gene Ontology Consortium. 2001. Creating the gene ontology resource: design and implementation. Genome Res **11**:1425-1433.

[13] Horrocks, I. Patel-Schneider, P.F, van Harlem, F. 2003. From SHIQ and RDF to OWL: The making of a web ontology language. J. of Web Semantics, **1(1):**7-26

[14] Bollen, M., and W. Stalmans. 1992. The structure, role, and regulation of type 1 protein phosphatases. Crit Rev Biochem Mol Biol **27**:227-281.
[15] Kile, B. T., N. A. Nicola, and W. S. Alexander. 2001. Negative regulators of cytokine signaling. Int J Hematol **73**:292-298.
[16] Cohen, P. 2002a. The origins of protein phosphorylation. Nat Cell Biol **4**:E127-130.
[17] Cohen, P. 1992. Signal integration at the level of protein kinases, protein phosphatases and their substrates. Trends Biochem Sci **17**:408-413.
[18] Alonso, A., J. Sasin, N. Bottini, I. Friedberg, I. Friedberg, A. Osterman, A. Godzik, T. Hunter, J. Dixon, and T. Mustelin. 2004. Protein tyrosine phosphatases in the human genome. Cell **117**:699-711.
[19] Cohen, P. T. 1997. Novel protein serine/threonine phosphatases: variety is the spice of life. Trends Biochem Sci **22**:245-251.
[20] Andersen, J. N., O. H. Mortensen, G. H. Peters, P. G. Drake, L. F. Iversen, O. H. Olsen, P. G. Jansen, H. S. Andersen, N. K. Tonks, and N. P. Moller. 2001. Structural and evolutionary relationships among protein tyrosine phosphatase domains. Mol Cell Biol **21**:7117-7136.
[21] Goldstein, B. J. 2001. Protein-tyrosine phosphatase 1B (PTP1B): a novel therapeutic target for type 2 diabetes mellitus, obesity and related states of insulin resistance. Curr Drug Targets Immune Endocr Metabol Disord **1**:265-275
[22] Schonthal, A. H. 2001. Role of serine/threonine protein phosphatase 2A in cancer. Cancer Lett **170**:1-13
[23] Zhang, Z. Y. 2001. Protein tyrosine phosphatases: prospects for therapeutics. Curr Opin Chem Biol **5**:416-423.
[24] Tian, Q., and J. Wang. 2002. Role of serine/threonine protein phosphatase in Alzheimer's disease. Neurosignals **11**:262-269.
[25] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P.F., editors. 2003. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2003.
[26] Wessel, M., and Möller, R. 2005. A High Performance Semantic Web Query Answering Engine. In I. Horrocks, U. Sattler, and F. Wolter, editors, Proc. International Workshop on Description Logics, 2005.
[27] Haarslev, V., and Möller, R. 2001. RACER system description. In Proceedings of the International Joint Conference on Automated Reasoning (IJCAR-01), volume 2083 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2001, pp. 701–705.
[28] Baker, C., Su, X., Butler, G., and Haarslev, V. 2006a. Ontoligent Interactive Query Tool. In Proceedings of the Canadian Semantic Web Working Symposium, June 6, 2006, Québec City, Québec, Canada, Series: Semantic Web and Beyond: Computing for Human Experience, Vol. 2, Springer Verlag, 2006, pp. 155-169.
[29] Shaban-Nejad, A., Baker, C., Haarslev, V., and Butler, G. 2005. The FungalWeb Ontology: Semantic Web Challenges in Bioinformatics and Genomics. In Semantic Web Challenge - Proceedings of the 4th International Semantic Web Conference, Nov. 6-10, Galway, Ireland, Springer-Verlag, LNCS, Vol. 3729, 2005, pp. 1063-1066, (2. Prize in the Semantic Web Challenges competition).
[30] Baker, C., Shaban-Nejad, A., Su, X., Haarslev, V., and Butler, G. 2006b. Semantic Web Infrastructure for Fungal Enzyme Biotechnologists. Journal of Web Semantics, (4)3, 2006.
[31] Horrocks et al, Instance Store paper
[32] Mabey, J. E., M. J. Anderson, P. F. Giles, C. J. Miller, T. K. Attwood, N. W. Paton, E. Bornberg-Bauer, G. D. Robson, S. G. Oliver, and D. W. Denning. 2004. CADRE: the Central Aspergillus Data REpository. Nucleic Acids Res **32**:D401-405.