A Little Semantic Web Goes a Long Way in Biology

K. Wolstencroft, A. Brass, I. Horrocks, P. Lord, U. Sattler, D. Turi, and R. Stevens

School of Computer Science, University of Manchester, UK

Abstract. We show how state-of-the-art Semantic Web technology can be used in e-Science, in particular, to automate the classification of proteins in biology. We show that the resulting classification was of comparable quality to that performed by a human expert, and how investigations using the classified data even resulted in the discovery of significant information that had previously been overlooked, leading to the identification of a possible drug-target.

1 Introduction

Semantic Web research has seen impressive strides in the development of languages, tools, and other infrastructure. In particular, the OWL ontology language, the Protégé ontology editor, and OWL reasoning tools such as FaCT++ and Racer are now in widespread use.

In this paper, we report on an application of Semantic Web technology in the domain of biology, where an OWL ontology and an OWL classification tool called the *Instance Store* were used to automate the classification of protein data. We show that the resulting classification was of comparable quality to one performed by a human expert, and how investigations using the classified data even resulted in either the discovery of new information or that which had been overlooked.

While this example focuses on a particular protein family and a particular set of model organisms, the technique should be applicable to other protein families, and to data from any sequenced genome—in fact we believe that similar techniques should be applicable to a wide range of investigations in biology, and in e-Science more generally. If this proves to be the case, then Semantic Web technology is set to have a major impact on e-Science.

Background and Motivation. The volume of genomic data is increasing at a seemingly exponential rate. In particular, *high throughput* technology has enabled the generation of large quantities of DNA sequence information. This sequence data, however, needs further analysis before it is useful to most biologists. This process, called *annotation*, augments the raw DNA sequence, and its derived protein sequence, with significant quantities of additional information describing its biological context.

Y. Gil et al. (Eds.): ISWC 2005, LNCS 3729, pp. 786–800, 2005.

One important process during annotation is the classification of proteins into different families. This is an important step in understanding the molecular biology of an organism. Attempts to automate this procedure have, however, not generally matched the gold-standard set by human experts. Human expert classification has been more accurate because their expertise allows them to recognise the properties that are sufficient, for example, to place an individual protein into a specific subfamily class. Automated methods have, in contrast, often failed to achieve the same level of specificity. Our goal, therefore, was to improve the precision of automatic protein classification, and bring it up to the same level as that achieved by human experts.

Overview of Our Technique. Given a set of proteins, each with a (partial) description of its properties, the objective is to find, for each of these proteins, the most specific protein family classes of which it is an instance. To describe protein family classes, we use an OWL-DL ontology; this enables us to specify necessary and sufficient conditions for a protein to be an instance of a given protein class. The ontology models the biology community's view of the current knowledge of protein classification. We then take protein data derived using standard bioinformatics analysis tools, translate these data into OWL-DL instance descriptions that use terms from the ontology, and use the Instance Store to classify these instances.

Empirical Evaluation. We have tested our system using data sets from both the human and *Aspergillus fumigatus* (a pathogenic fungus) genomes. We found that our automatic classification process performed at least as well as a human expert: it allows a fast and repeatable classification process, and the explicit representation of human expert knowledge means that there is a clear and explicit evidence base for the classification. Moreover, the precise and methodical classification of the data led to the discovery of new information about these proteins, including a protein subclass that seems to be specific to pathogenic fungi, and could thus be an important drug-target for pharmaceutical investigations.

2 Science and Technology

In this section, we describe the biology problem we have tackled and the Semantic Web technology that we used to achieve an appropriate solution.

2.1 Classifying Proteins

The process of annotation follows the "central dogma" of molecular biology. In broad outline, this process consists of the following steps: firstly DNA is sequenced; then genes are identified in this DNA; the DNA is then translated into a protein sequence; the proteins are then analysed and annotated with information useful for further biological investigation. As the majority of the functions of a cell are carried out by proteins, it is those proteins in which most

biologists are interested. Proteins are classified into families that both reflect the functions they carry out in the cell, as well as often giving clear indications as to the biological processes in which they are involved. It is this classification, along with other and diverse kinds of information, which makes up the annotation of a protein and makes the large data sets manageable, enabling biologists to perform more thorough investigations.

In the last decade, various steps of this process have been automated, and thus their speed has increased enormously. Sequencing of whole genomes[1] is now routine. Gene discovery is technically challenging, but responds well to the increasing availability of CPU cycles. However, this still leaves a large number of protein sequences—approximately 30 000 in the human genome, a quantity that is more or less in other species. This quantity is far more than that with which the individual biologist can cope.

The automation of the annotation process has, however, lagged behind advances in other parts of this process. To date, automated approaches have proven to be quicker than human expert annotation, but the level of detail is often reduced [26,6]. As a consequence, many protein sequences are not annotated with the accurate, specific information necessary for bioinformatics analyses. Thus useful resources for further biological discovery remain untapped.

In this investigation, we have used one protein family, the *protein phosphatase* family, as a case study to demonstrate a new, ontology-based method for automated annotation. This method was designed to combine the speed of automated annotation with some of the detailed knowledge that experts use in annotation.

Protein phosphatases are a large and varied protein family. Together with another family, the protein kinases, they are critically involve in controlling the activity of many other proteins, thereby forming an essential part of the feedback control mechanism within the cell.

Given this pivotal role, it is perhaps unsurprising that many protein phosphatases have been implicated in various diseases of great medical importance, including diabetes, cancer, and neurodegenerative conditions. Phosphatases are therefore a major subject of medical and pharmaceutical research.

In general, proteins are relatively modular and comprise of a number of different *protein domains*. Using a protein sequence, it is often possible to computationally determine the protein domains of which it is composed. For many protein families, including the protein phosphatases, it is possible to classify their members based on the protein domains of which they are composed. To avoid confusion with interpretation domains or the domain of a property, for the remainder of this paper, we use "p-domain" for protein domain.

The different p-domain composition of proteins suggests the specific function of a protein. , Individual p-domains, however, often have specific and separate functions from the protein as a whole. For example, an enzyme will have a catalytic p-domain that performs the catalysis on the substrate molecule, but it will also contain structural p-domains and binding p-domains that ensure that the substrate can interact with the catalytic p-domain. Therefore, a specific

[1] A genome is the entirety of DNA in a cell.

combination of p-domains is required for a protein to function correctly. In some cases, the presence of a certain p-domain is *diagnostic* for membership in a particular protein family, i.e., some p-domains only occur in a single protein family. If a protein contains one of these diagnostic p-domains, it must belong to that particular family. For example, the protein tyrosine kinase catalytic p-domain is diagnostic for the tyrosine kinases.

Most protein families are, however, defined by a non-trivial combination of p-domains. For example, as you descend the hierarchical structure, extra p-domains (and therefore more specific functional properties) are observed in the protein class definitions. For example, an R5 phosphatase is a type of classical receptor tyrosine phosphatase. As a tyrosine phosphatase, it contains at least one phosphatase catalytic p-domain and, as a receptor tyrosine phosphatase, it contains a transmembrane region. The R5 type actually contains two catalytic p-domains and a fibronectin p-domain, identifying it as an instance of even more specific subclasses.

Identifying the p-domain composition of a protein is, therefore, a first step towards its classification. There are databases describing functional p-domains, for example, PROSITE [17], SMART [20] and InterPro [23], and these databases come with specific tools, such as InterProScan, which can report the presence of these p-domains in a novel protein sequence. Bioinformaticians are, however, usually required to perform the analysis that places a protein (with its set of p-domains) into a particular protein family. The whole process of classifying proteins from a genome can be accomplished with the following steps:

1. Given a genome, we extract DNA gene sequences, which we then translate into the set of protein sequences. If we are interested in a particular protein family, we can sub-select sequences containing p-domains diagnostic of that family.
2. On each of the extracted proteins, we use InterProScan to determine its p-domain composition.
3. For each of these compositions, we identify the protein family or subfamily to which it belongs by comparing them to the available biological knowledge.

The final step currently requires the most human analysis and expert knowledge. Manual classification methods are carried out by protein family experts to interpret these data and use their expert knowledge to classify proteins to a fine-grained level. To the best of our knowledge, no automated method has yet been able to replicate this expert level of detail and precision.

2.2 Ontologies and the Instance Store

Ontologies, with their intuitive taxonomic structure and class based semantics, are widely used in domains like bio- and medical-informatics, where there is a tradition of establishing taxonomies of terms. The recent W3C recommendation of OWL[2] as the language of choice for web ontologies also underlines the long

[2] See `http://www.w3.org/2004/OWL/` or [11].

term vision that ontologies will play a central role in the Semantic Web. Most importantly, as shown in [4], most of the available OWL ontologies can be captured in OWL-DL—a subset of OWL for which highly optimised Description Logic [2] reasoners can be used to support ontology design and deployment.

Unfortunately, existing reasoners (and tools), while successful in dealing with the (relatively small and static) class level information in ontologies, fail when presented with the large volumes of instance level data often required by realistic applications, hampering the use of reasoning over ontologies beyond the class level. The system we have used—the instance Store (iS) [14]—addresses this problem using a hybrid database/reasoner architecture: a relational database is used to make instances persistent, while a class level ("TBox" in Description Logic terms) reasoner is used to infer ontological information about the classes to which they belong. Moreover, part of this ontological information is also made persistent in the database. The iS currently only supports a rather limited form of reasoning about individuals: it takes an ontology (without instances), a set of axioms asserting class-instance relationships, and answers queries asking for all the instances of a class description. The classes in both axioms and queries can be arbitrarily complex OWL-DL descriptions, and a DL reasoner is used to ensure that *all* instances (explicit and implicit) of the query concept are returned. In the remainder of this paper, we use "class-level ontology" for an ontology in which no instances occur. From a theoretical perspective, this might seem un-interesting; the iS is, however, able to deal with much larger numbers of individuals than would be possible using a standard DL reasoner. More importantly, this kind of reasoning turns out to be useful in a range of applications, in particular those such as the one presented here where a domain model is used to structure and classify large data sets.[3]

There is a long tradition of coupling databases to knowledge representation systems in order to perform reasoning, most notably the work in [5]. However, in the iS, we do not use the standard approach of associating a table (or view) with each class and property. Instead, we have a fixed and relatively simple schema that is independent of the structure of the ontology and of the instance data. The iS is, therefore, agnostic about the provenance of data, and uses a new, dedicated database for each ontology (although the schema is always the same).

The basic functionality of the iS system are illustrated in Figure 1. At start-up, the `initialise` method is called with a relational database, an OWL-DL class reasoner such as Racer [9] or Fact++ [30], and a class-level OWL-DL ontology. The method creates the schema for the database if needed (i.e., if the iS is new), parses the ontology, and loads it into the reasoner. To populate the iS, the `addAssertion` method is called repeatedly. Each assertion states that an instance (identified by a URI) belongs to class (which is an arbitrary OWL-DL description). Once the iS has been populated with some—possibly millions of—instances, it can be queried using the `retrieve` method. A query again consists of an arbitrary (possibly complex) OWL-DL class description; the result is the set of all instances belonging to the

[3] The iS was initially developed for use in a Web Service registry application, where it was used to classify and retrieve (large numbers of) descriptions of web services.

```
initialise(database: Database, reasoner: OWLReasoner, ontology: OWLOntology)
addAssertion(instance: URI, class: OWLDescription)
retrieve(query: OWLDescription): Set⟨URI⟩
```

Fig. 1. The IS API

query class, and is returned by `retrieve` as a set of URIs. The IS uses database queries to return individuals that are "obviously" instances of the query class, and to identify those instances where the DL reasoner is needed in order to determine if they form part of the answer set.

3 Description of the Experiments Undertaken

The method we present could be applicable in general to many protein families, but to demonstrate the technique and the fine-grained classification possible, we present the analysis of one family, the protein phosphatases, in the human and *Aspergillus fumigatus* genomes.

We have combined automated reasoning techniques [9,14] with elements of a service-oriented architecture [27,19] to produce a system to automatically extract

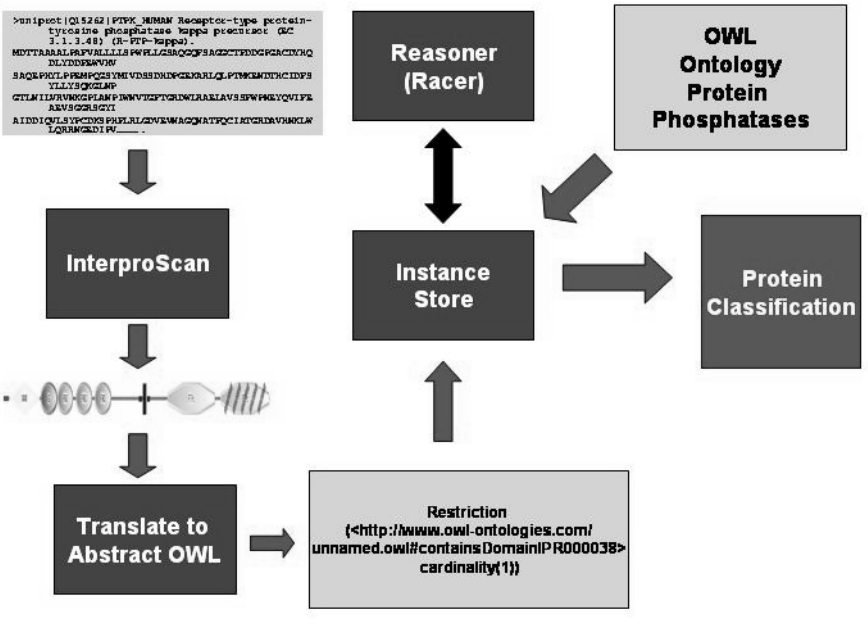

Fig. 2. The Ontology Classification System Architecture

and classify the set of protein phosphatase from an organism.[4] Figure 2 shows the components in our protein classification experiment. An OWL class-level ontology describes the protein phosphatase family, and this ontology is pre-loaded into the Instance Store. Protein instance data is extracted from the protein set of a genome, and the p-domain composition is determined using InterProScan. These 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 DL reasoner (Racer in this case), to classify each such instance. For each protein class from our ontology, it returns those proteins that can be inferred to be an instance of this class.

In the remainder of this section, we will describe the relevant components of this architecture in more detail, and explain the outcomes of this experiment from a biology perspective. In the next section, we describe the experience gained and lessons learnt from a computer science perspective.

3.1 The Ontology

In this section, we describe how we capture the expert knowledge for phosphatase classification in an OWL-DL ontology. All the information used for developing our ontology comes from peer-reviewed literature from protein phosphatase experts. The family of human protein phosphatases has been well characterised experimentally, and detailed reviews of the classification and family composition are available [1,7,18]. 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; we will comment on this in Section 4.

Fortunately for this application, there are precise rules,[5] based on p-domain composition, for protein family membership, and we can express these rules 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 step which had previously required human intervention, thus allowing for full automation of the complete process. In biology, the use of ontologies to capture human knowledge of a particular domain and to answer complex queries is becoming well established [8,28]. Less well established is the use of reasoning systems for data interpretation. In this study, we present a method which makes use of ontology reasoning and illustrates the advantages of such an approach.

The ontology was developed in OWL-DL using the Protégé editor,[6] and currently contains 80 classes and 39 properties; it is available at `(http://www.bioinf.man.ac.uk/phosphabase/download)`. Part of the subsumption hierarchy inferred from these descriptions can be seen in the left-hand panel of Figure 3, which shows the OWL ontology in the Protégé editor.

[4] Due to the relatively small test-set used, the case study reported here could have been carried out using Racer [9] only, i.e., without the iS. However, larger sets of protein data will necessitate the use of iS or a similar tool.

[5] We use "rules" here in a completely informal way.

[6] We used Protégé 3.0 with OWL plugin 1.3, build 225.1.

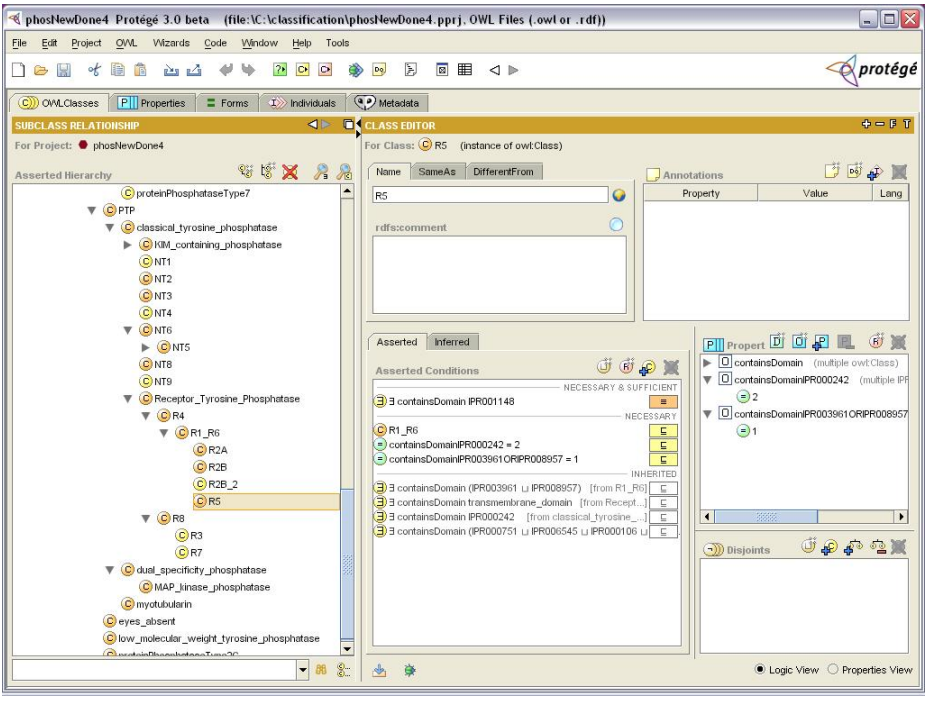

Fig. 3. A screenshot of the phosphatase ontology in the OWL ontology editor Protégé

More precisely, for each class of phosphatase, this ontology contains a (necessary and sufficient) definition. For this family of proteins, this definition is, in most cases, a conjunction of p-domain compositions, i.e., a typical case of a phosphatase class definition looks as follows, where X_i are p-domains:

If a Y protein contains at least n_1 p-domains of type X_1 and ... and at least n_m p-domains of type X_m, then this protein also belongs to class Z.

For example, receptor tyrosine phosphatases contain one or two phosphatase catalytic p-domains, and receptor tyrosine R2B phosphatases contain exactly 2 tyrosine phosphatase catalytic p-domains, one transmembrane p-domain, at least one fibronectin p-domain, and at least one immunoglobulin p-domain. In some cases, X_i is a disjunction of p-domains. P-domains come with a rather "flat" structure, i.e., only few p-domains are specialisations of others. Clearly, "counting" statements such as the one above go beyond the expressive power of OWL since they would require (the OWL equivalent of) *qualified* cardinality restrictions [10], whereas OWL only provides unqualified cardinality restrictions through its `restriction(UminCardinality(n))` and `restriction(UmaxCardinality(n))` constructs. In contrast, this kind of expressive means was provided by DAML+OIL [15], i.e., we could have defined the above mentioned receptor tyrosine phosphatases using the expression

```
IntersectionOf(Restriction(contains minCardinality(1) PhCatalDoms)
               Restriction(contains maxCardinality(2) PhCatalDoms)
```

To overcome this problem, we used a well-known work-around.[7] For each X_i that we would have liked to use in a qualified number restriction, we introduced a subproperty `containsX_i` of `contains`, and set the range of `containsX_i` to the class X_i. In addition, we added sub-property assertions so that the hierarchy of newly introduced properties `containsX_i` reflects the class hierarchy of the classes X_i used. Unfortunately, this work-around is not always correct. That is, assume there are two ontologies, one with qualified number restrictions and one that resulted from the application of this work-around. Then there are cases where the first one implies a subsumption relationship between two classes, whereas the second one does not imply this subsumption. Similarly, a class may be unsatisfiable w.r.t. the first one, but satisfiable w.r.t. the second. We used this work-around believing that it was correct and, when we learned that it sometimes is not, were quite surprised—we had "cluttered" our ontology with a large number of new properties without this guaranteeing the desired effect. However, we then checked that, in the special case of our experiment, this work-around is indeed correct, even though we are not going to prove this here. We will comment more on this in Section 4 and 5.

Having captured the expert knowledge in this way, we are left with the problem of dealing with the potentially very large numbers of protein instances that need to be classified according to the corresponding ontology. This requirement motivated our use of the IS.

3.2 The Data Sets

This study focuses on the previously identified and described human phosphatases [1,24], and the less well characterised *A.fumigatus* protein phosphatases. The human phosphatases, having been carefully hand-classified, form a control group for our automated protein phosphatase classification. Previous classification of human phosphatases by biological experts provides a substantial test-set for our approach. If the IS can classify the characterised proteins (at least) as well as human experts, then this would increase our confidence when using our method on unknown genomes. The *A.fumigatus* genome falls between these extremes, and thus 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 [22]. We have considered 118 human phosphatases and 45 from *A.fumigatus*.

Pre-Screening. Isolation of the protein phosphatase sequences from the protein set of the genome was achieved by screening for diagnostic phosphatase motifs, i.e. for specific patterns. These are

1. the protein tyrosine phosphatase active site motif H-C-X(5)-R
2. the protein serine/threonine phosphatase motif [LIVMN]-[KR]-G-N-H-E

[7] See, e.g., `http://www.cs.vu.nl/~guus/public/qcr.html`

3. the protein phosphatase C signature motif [LIVMFY]-[LIVMFYA]-[GSAC]-[LIVM]-[FYC]-D-G-H-[GAV].

The EMBOSS program, PatMatDB [25] was used to perform the pre-screening process. Performing an InterProScan on every protein sequence from the genome would also have isolated the protein phosphatase sequences, but each InterProScan can take several minutes. PatMatDB can screen the whole genome in the time taken to run one InterProScan, so we decided to use InterProScan only for the detailed analysis of each sequence identified as being a protein phosphatase.

3.3 Queries Asked and Results

The purposes of the human and *A.fumigatus* studies were different. The human study was a proof of concept to demonstrate the effectiveness of the automated method. The *A.fumigatus* study was more focused towards biological discovery.

For the human phosphatases, we were interested in comparing the automated classification with the thorough, human expert classification. Therefore, we browsed the class hierarchy of our phosphatase ontology and, for each class, we retrieved those proteins for which the iS inferred that this class was the most specific one. We were also interested in identifying instances that did not fit any of the ontology class definitions (i.e., whose most specific class was the top class).

For the *A.fumigatus* phosphatases, we browsed the class hierarchy in a similar way but, as the phosphatases from this organism were less well characterised, we were particularly interested in the differences between the human and *A.fumigatus* set, i.e., we were interested in finding classes that had instances of the human proteins, but not of the *A.fumigatus* proteins, and vice versa. All these queries could be answered easily and quickly using the iS.

The results of this experiment were three-fold. Firstly, we found that the automated classification of the human protein phosphatases performed as well as the manual classification by phosphatase experts. Since the same protein instances were used in the automated and manual studies, we could compare these two classifications, and it turned out that both classifications put almost all phosphatases into the same place in the class hierarchy. This evidence shows proof of concept, and suggests that the automated approach could be used to solve the current annotation bottleneck. Secondly, in the few cases where the automatic and the manual classification differed, detailed investigations by a domain expert revealed that the automatic one was actually "more correct": we discovered two proteins for which no appropriate class was available, i.e. they were classified by the automatic classification as instances of the top phosphatases class.

This discovery lead to a modification of the ontology, and thus of the expert knowledge on proteins. One of these phosphatases was DUSP10 (Dual specificity phosphatase 10). It was found to contain an extra p-domain, a *disintegrin*. This particular p-domain is not found in any other protein phosphatase and poses interesting questions about possible protein functions to the biologists. Our automated classification method was able to find these mis-classifications because the iS applied the expert knowledge systematically and consistently.

The automated classification of the *A.fumigatus* phosphatases revealed large differences from the human phosphatases. Not only were there fewer individual proteins, but whole subfamilies were missing. Some of these differences can be attributed to the differences in the two organisms. Many phosphatases in the human classification were tissue-specific variations of tissue-types that do not occur in *A.fumigatus*. 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, i.e., again, a phosphatase that was classified automatically only as an instance of the top class. Calcineurin is well conserved throughout evolution and performs the same function in all organisms. However, in *A.fumigatus*, it contains an extra functional p-domain. Further bioinformatics analyses revealed that this extra p-domain also occurs in other pathogenic fungus species, but in no other organisms, suggesting a specific functional role for this extra p-domain. Previous studies have identified divergences in the mechanism of action of calcineurin in pathogenic fungi as being linked to virulence, so this protein is an interesting drug-target for future study.

4 Lessons Learnt

As we have seen, we have successfully used Semantic Web technology in a bioinformatics application. Besides finding new protein families that are of interest to biologists, we have shown that automated classification can indeed compete with manual classification, and is sometimes even superior. Our approach to automated classification combines the advantages of speed of the automated methods and accuracy of human expert classification, the latter being due to the fact that we captured the expert knowledge in an OWL ontology. The combination of the two, namely speed and expert knowledge, provides a quick and efficient method for classifying proteins on a genomic scale, and offers a solution to the current annotation bottleneck.

Our approach was made possible by the development of state-of-the-art Semantic Web technology, such as the OWL ontology language, the Protégé OWL ontology editor, the OWL Instance Store, and the Racer OWL reasoner; this technology did not emerge overnight, but is based on decades of research in logic-based knowledge representation and reasoning. Although neither Racer nor the iS support all of OWL-DL,[8] these tools proved more than adequate for our experiment.

In contrast, a limitation in the expressive power of OWL-DL *did* cause considerable problems: the lack of qualified number restrictions (also called qualified cardinality restrictions). In order to overcome this limitation, we had to employ a work around and verify that this work around, even though not correct in general, was correct for our ontology and instance data. This work around introduced a significant overhead, and was only possible through a close co-operation between the

[8] Racer does not support individual names in complex class descriptions (so-called nominals—see [16]), and the current version of iS does not support role assertions between individuals.

biologists and computer scientists. We, therefor, cannot recommend such an approach in general. Additionally, we observe that, from a theoretical and practical perspective, this work around should not be necessary since (a) reasoners such as Racer and Fact [9,13] support qualified number restrictions, (b) for all Description Logics we are aware of that support (unqualified) number restrictions, the worst-case complexity of reasoning remains the same when they are extended with qualified number restrictions (see, e.g., [29]), and (c) the latest version of Protégé-OWL now supports qualified number restrictions. Hence we can, in the future, run similar experiments without having to resort to this work around, provided that we are willing to diverge from the current OWL standard.

The ability to run such experiments is of considerable importance since there is a wealth of unannotated and partially annotated data in the public domain, to which we plan to apply our approach. New genomes are being sequenced continually, and some existing genomes have not been annotated to any degree of detail. Now that the ontology system architecture is in place, new proteins can be quickly and successfully classified as members of protein phosphatase subfamilies. Development of other ontologies, would enable the application of this technique to some of the 1,000's of other protein families.

This paper demonstrates a proof of concept for the automated classification of proteins using automated reasoning technologies. From a study involving a single protein family and two species, we were able to identify a new protein subclass. As this class of protein appears to be specific to pathogenic fungi, it is potentially useful for further pharmaceutical investigations. Automated reasoning over instance data has therefore enabled us to generate new hypotheses which will require significiant further laboratory experimentation, which, in turn, will potentially improve our understanding of protein phosphorylation.

Finally, we would like to point out that the ontology definitions are produced from expert protein family knowledge. Therefore, they reflect what is currently known in the research community, and are made explicit in a machine-understandable format, namely OWL-DL. This has several important consequences. Firstly, the construction of such an ontology can help in the development of a consensus from within the community [3], and even if the community fails to agree on a single ontology, automated classification could be used to enable "parallel" alternative annotations. Secondly, if the community knowledge of the protein family changes, the ontology can easily be altered, and the protein instances can be re-classified accordingly. Lastly, if the definitions are based on what is known, proteins that do not fit into any of the defined classes are easily identified, making the discovery of new protein subfamilies possible.

5 Outlook and Future Work

Our plans for future work are manifold. Basically, we want to do more "automated" biology, but we are thereby pushing the current state-of-art in logic-based knowledge representation, automated reasoning, and Semantic Web technology. Within this section, we only discuss three of the related issues.

Firstly, we observe that a protein is a sequence of amino acids, and thus sequences can be seen as strings over a twenty letter alphabet since there are only twenty amino acids. In our current ontology, we do not capture this sequence information, and thus cannot answer queries related to these sequences. From a biology perspective, however, queries such as "give me all proteins whose amino acid sequence contains an M followed by some arbitrary sub string, which is then followd by a NEN" would be really valuable. From a computer science perspective, we could easily express (and query over) these strings using a simple form of concrete domains, so-called datatypes [21,12]. However, the datatypes currently available in OWL do not provide predicates that compare a given string with a regular expression, a comparison that would reflect the above example query.

Secondly, we are currently concerned with a single class of components of an organism, namely the proteins. In the future, we want to use the available technology to automate investigations into their interaction, and also represent and reason about larger structures such as genomes and cells. We could easily model interactions between proteins using a property `interact` to make statements such as "proteins of class `X` only interact with proteins of class `Y`". However, we would also need to make statements on an instance level such as "this protein instance interacts with that protein instance", which is possible in OWL-DL, but goes beyond the capabilities of the current iS. We are currently extending the iS to handle statements of this kind, and we will see if this extension is able to cope with the large volumes of data that will be needed in biology applications.[9]

Thirdly, we will "roll back" the work-around we used to cope with the absence of qualified number restrictions, both in our ontology and in the instance data, instead using the form of qualified number restrictions provided by Protégé, Racer, and the iS. This will greatly enhance the interpretability of the current ontology and also make its extension to other families of proteins more straight-forward.

Acknowledgements. This work was funded by an MRC PhD studentship, the myGrid e-science project, University of Manchester with the UK e-science programme EPSRC grant GR/R67743 and the ComparaGRID project, BBSRC grant BBS/B/17131. 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.

References

1. A. Alonso, J. Sasin, N. Bottini, I. Friedberg, I. Friedberg, A. Osterman, A. Godzik, T. Hunter, J. Dixon, and T. Mustelin. Protein tyrosine phosphatases in the human genome. *Cell*, 117(6):699–711, 2004.

[9] Racer can already handle such statements, but can only deal with a relatively small number of individuals.

2. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
3. M. Bada, D. Turi, R. McEntire, and R. Stevens. Using Reasoning to Guide Annotation with Gene Ontology Terms in GOAT. *SIGMOD Record (special issue on data engineering for the life sciences)*, 2004.
4. S. Bechhofer and R. Volz. Patching syntax in OWL ontologies. In *Proc. of the 3rd International Semantic Web Conference (ISWC)*, 2004.
5. A. Borgida and R. J. Brachman. Loading data into description reasoners. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 217–226, 1993.
6. K. Carter, A. Oka, G. Tamiya, and M. I. Bellgard. Bioinformatics issues for automating the annotation of genomic sequences. *Genome Inform Ser Workshop Genome Inform*, 12:204–11, 2001.
7. P. T. Cohen. Novel protein serine/threonine phosphatases: variety is the spice of life. *Trends Biochem Sci*, 22(7):245–51, July 1997.
8. Gene Ontology Consortium. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.
9. V. Haarslev and R. Möller. RACER system description. In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR-01)*, volume 2083 of *Lecture Notes in Artificial Intelligence*, pages 701–705. Springer-Verlag, 2001.
10. B. Hollunder and F. Baader. Qualifying number restrictions in concept languages. In *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning (KR-91)*, pages 335–346, 1991.
11. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1), 2003.
12. I. Horrocks and U. Sattler. Ontology reasoning in the SHOQ(D) description logic. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.
13. I. Horrocks. Using an expressive description logic: FaCT or fiction? In *Proceedings of the Sixth International Conference on the Principles of Knowledge Representation and Reasoning (KR-98)*, pages 636–647, 1998.
14. I. Horrocks, L. Li, D. Turi, and S. Bechhofer. The instance store: DL reasoning with large numbers of individuals. In *Proc. of the 2004 Description Logic Workshop (DL 2004)*, 2004. available at CEUR, `www.ceur.org`, see also `instancestore.man.ac.uk`.
15. I. Horrocks, P. Patel-Schneider, and F. van Harmelen. Reviewing the design of DAML+OIL: An ontology language for the semantic web. In *Proc. of the 18th Nat. Conf. on Artificial Intelligence (AAAI 2002)*, pages 792–797. AAAI Press, 2002.
16. I. Horrocks and U. Sattler. A tableaux decision procedure for SHOIQ. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, 2005.
17. N. Hulo, C. J. Sigrist, V. Le Saux, P. S. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. De Castro, P. Bucher, and A. Bairoch. Recent improvements to the prosite database. *Nucleic Acids Res*, 32:134–7, 2004.
18. P. J. Kennelly. Protein phosphatases–a phylogenetic perspective. *Chem Rev*, 101(8):2291–312, 2001.
19. K. Wolstencroft, P. Lord, L. Tabernero, A. Brass, and R. Stevens. Intelligent classification of proteins using an ontology. Submitted, 2005.

20. I. Letunic, R. R. Copley, S. Schmidt, F. D. Ciccarelli, T. Doerks, J. Schultz, C. P. Ponting, and P. Bork. Smart 4.0: towards genomic data integration. *Nucleic Acids Res*, 32:142–4, 2004.
21. C. Lutz. Description logics with concrete domains—a survey. In *Advances in Modal Logics Volume 4*. World Scientific Publishing Co. Pte. Ltd., 2003.
22. J. E. Mabey, 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. Cadre: the central *aspergillus* data repository. *Nucleic Acids Res*, 32:401–5, 2004.
23. N. J. Mulder, R. Apweiler, T. K. Attwood, et al. Interpro, progress and status in 2005. *Nucleic Acids Res*, 33:201–5, 2005.
24. T. Mustelin, T. Vang, and N. Bottini. Protein tyrosine phosphatases and the immune response. *Nat Rev Immunol*, 5(1):43–57, January 2005.
25. P. Rice, I. Longden, and A. Bleasby. EMBOSS: the European molecular biology open software suite. *Trends Genet*, 16(6):276–7, June 2000.
26. T. F. Smith and X. Zhang. The challenges of genome sequence annotation or "the devil is in the details". *Nat Biotechnol*, 15(12):1222–3, 1997.
27. R. Stevens, H. Tipney, C. Wroe, T. Oinn, M. Senger, P. Lord, C. Goble, A. Brass, and M. Tassabehji. Exploring Williams Beuren Syndrome Using MyGrid. In *Bioinformatics*, volume 20, pages 303–310, 2004. Intelligent Systems for Molecular Biology (ISMB) 2004.
28. R. Stevens, C. Wroe, P. Lord, and C. Goble. Ontologies in bioinformatics. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 635–657. Springer, 2003.
29. S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, 2001. electronically available at `http://www.bth.rwth-aachen.de/ediss/ediss.html`.
30. D. Tsarkov and I. Horrocks. Efficient reasoning with range and domain constraints. In *Proceedings of the 2004 Description Logic Workshop (DL 2004)*. CEUR, 2004. Available from `ceur-ws.org`.