Protein Ontology Project: 2007 Updates

Amandeep S. Sidhu, Tharam S. Dillon, Elizabeth Chang

*Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, Australia*

**Abstract**

Protein Ontology (PO) provides integration of heterogeneous protein and biological data sources. PO converts the enormous amounts of data collected by geneticists and molecular biologists into information that scientists, physicians and other health care professionals and researchers can use to easily understand the mapping of relationships inside protein molecules, interaction between two protein molecules and interactions between protein and other macromolecules at cellular level. This paper discusses the updates that happened to Protein Ontology Project since it was last presented at Data Mining 2006 Conference.

**Availability:** [http://www.proteinontology.info/](http://www.proteinontology.info/)

**Keywords:** Protein Ontology, Proteomics, Bioinformatics, Protein Informatics, Computational Proteomics, Protein Structure, Biomedical Ontologies, Data Integration, Data Semantics

1 **Introduction**

The ultimate goal of protein annotator framework or Protein Ontology (PO) is to deduce from proteomics data all its biological features and describing all intermediate structures: primary amino acid sequence, secondary structure folds and domains, tertiary three dimensional atomic structure, quaternary active functional sites,
etc. Thus, complete protein annotation for all types of proteins for an organism is a very complex process that requires besides extracting data from various protein databases, integration of additional information: results of protein experiments, analysis of bioinformatics tools, and biological knowledge accumulated over years. This constitutes a huge mass of heterogeneous protein data sources that need to rightly represented and stored. Protein Annotators must be able to readily retrieve and consult these data. Therefore protein databases and man-machine interfaces are very important when defining a protein annotation using protein ontology.

The process of development of a protein annotation based on our protein ontology requires an important effort to organize, standardize and rationalize protein data and concepts. First of all, protein information must be defined and organized in a systematic manner in databases. In this context, PO addresses the following problems of existing protein databases: redundancy, data quality (errors, incorrect annotations, and inconsistencies), lack of standardization in nomenclature etc. The process of annotation relies heavily on integration of heterogeneous protein data. Integration is thus a key concept if one wants to make full use of protein data from collections. In order to be able to integrate various protein data it is important that community agree upon concepts underlying the data. PO provides a framework of structured vocabularies and standardized description of protein concepts that helps to achieve this agreement and achieve uniformity in protein data representation [1-4].

PO consists of concepts (or classes), which are data descriptors for proteomics data and the relations among these concepts. PO has (1) a hierarchical classification of concepts represented as classes, from general to specific; (2) a list of attributes related to each concept, for each class; and (3) a set of relations between classes to link concepts in ontology in more complicated ways then implied by the hierarchy, to promote reuse of concepts in the ontology.

Protein Ontology Instance Store is created for various protein data using the PO format. PO uses data sources like PDB, SCOP, OMIM and various published scientific literature to gather protein data. The PO instance store covers various species of proteins from bacterial and plant proteins to human proteins. Such a generic
representation using PO shows the strength of PO format representation.
In this paper, we discuss the updates that happened to Protein Ontology Project since it was last presented at Data Mining 2006 Conference [5]. Section 2 outlines PO Algebra based on our earlier work on Semantic Relationships in PO [5]. Section 3 discusses results of various data mining algorithms on PO Instance Store. Lastly, Section 4 discusses our future work on building Trustworthy Protein Ontology.

2 PO Algebra

Several approaches for data interoperability identified by Karp [6] have been implemented for biological databases. We extend Karp’s approach for interoperability not only to protein databases but also to knowledge bases and other information sources. This section outlines algebra for protein data source composition based on our earlier work of Semantic Relationships in Protein Ontology discussed in Data Mining 2006 [5].

Our common conceptual model for the internal representation of PO is based on the work done by Gyssens et al [7]. In its core, we represent a protein ontology as a graph. Formally, an ontology \( O = (G, R) \) is represented as a directed labelled graph \( G \) and a set of rules \( R \). The graph \( G = (N, E) \) comprises a finite set of nodes \( N \) and a finite set of edges \( E \). A non-null string gives the label of a node. In the context of protein ontology, the label is often represents a concept defined in protein ontology. The label of an edge is the name of a semantic relationship among the concepts in protein ontology and can be null if the relationship is not known. The set of logical rules \( R \), associated with protein ontology, are rules expressed in a logic-based language used to derive data from existing protein data and information sources.

The key to scalability of PO conceptual model is the systematic and effective composition of data and information. In this section, we present PO ontology algebra that allows composition of multiple levels of information stored in the ontology for information retrieval. By retaining a log of composition process, we can also, with minimal adaptations, replay the composition whenever any of the underlying data sources that PO integrates change. The algebra
has one unary operator: Select, and three binary operations: Intersection, Union and Difference.

Select Operator allows us to highlight and select portions of the PO that are relevant to query at hand. Given the PO structure and a concept to be selected, the select operator selects the sub tree rooted at that concept. Given the PO structure and a set of concepts, the select operator selects only those edges in the PO that connect nodes in a given set. Select Operator is defined as:

\[ OS = \sigma (NS, ES, RS) \text{ where} \]
\[ NS = \text{Nodes (condition = true)} \]
\[ ES = \text{Edges (} \forall N \in NS \text{)} \]

Intersection Operator is the most important and interesting binary operation. Let \( O1 = (NI, EI, RI) \), and \( O2 = (N2, E2, R2) \) be the two parts of PO whose composition will provide answer to the query submitted by the user. Here N is the set of nodes or concepts of PO, E is the set of edges or the PO hierarchy, and R is set of Semantic Relationships. The intersection of two parts of PO with respect to semantic relationships (SR) of PO is:

\[ OI (1, 2) = O1 \cap_{SR} O2 = (NI, EI, RI), \text{ where} \]
\[ NI = \text{Nodes (SR (O1, O2))}, \]
\[ EI = \text{Edges (} EI \text{, } NI \cap NI \text{) + Edges (} E2 \text{, } NI \cap N2 \text{) + Edges (SR (} O1 \text{, } O2 \text{))}, \text{ and} \]
\[ RI = \text{Relationships (} OI \text{, } NI \cap NI \text{) + Relationships (} O2 \text{, } NI \cap N2 \text{) + SR (} O1 \text{, } O2 \text{) – Edges (SR (} O1 \text{, } O2 \text{))}. \]

Note that SR is different from R, as it does not include sequences. The nodes in the intersection ontology are those nodes that appear in the semantic relationships, SR. The edges in the intersection ontology are the edges among nodes that are either present in the source parts of the ontology or have been established as a semantic relationship, SR. Relationships in the intersection ontology are the relationships that have not been already been modelled as edges and those relationships present in source parts of the ontology that use only concepts that occur in intersection ontology.
Union Operator combines two parts of the ontology retaining only one copy of the concepts in the intersection. The union of two parts of PO, $O_1 = (N_1, E_1, R_1)$, and $O_2 = (N_2, E_2, R_2)$ with respect to semantic relationships (SR) of PO is expressed as:

$$O_1 \cup_{SR} O_2 = (NU, EU, RU), \text{ where,}$$
$$NU = N_1 \cup N_2 \cup NI (1, 2),$$
$$EU = E_1 \cup E_2 \cup EI (1, 2), \text{ and}$$
$$RU = R_1 \cup R_2 \cup RI (1, 2), \text{ where,}$$
$$O_1 \cap_{SR} O_2 = (NI (1, 2), EI (1, 2), RI (1, 2)) \text{ is the intersection of two ontologies.}$$

The difference of two parts of PO - $O_1$ and $O_2$, computed by Difference Operator, written as $O_1 - O_2$, includes portions of the first part that are not common to the second part. The difference can be rewritten as $O_1 - (O_1 \cap_{SR} O_2)$. The nodes, edges and relationships that are not in intersection but are present in the first part comprise the difference.

One of the objectives of computing the difference is to optimise the maintenance of PO. As the PO instance store is huge and so many people add instances to it, difference will suggest that instances are not entered properly or there is change in underlying data sources that PO integrates. Change suggested by difference is forwarded to the administrator. If the change happens to be in difference between structures of parts considered, then it does not occur in intersection and is not related to any semantic relationships that establish bridged between the parts of the ontology. Therefore Semantic Relationships do not need to be changed. If the changes is because of changes happened to underlying data sources that PO integrates, then set of concepts and semantic relationships need to be checked for any changes required to remove the difference.

In this section we covered the PO ontology algebra that allows composition of multiple levels of information stored in the protein ontology for information retrieval. The PO approach supports precise composition of information from multiple diverse sources providing semantic relationships between among such sources. This approach allows reliable exploitation of protein information sources without any imposition on the sources themselves. The PO
algebra based on semantic relationships allows systematic composition, which unlike integration is more scalable.

3 Mining PO Instance Store

Tree Mining has attracted lots of interest among the data mining community, due to the increasing use of semi-structured data sources for more meaningful knowledge representations. This is particularly evident in areas such as Bioinformatics, XML Mining, Web applications, scientific data management, and more generally in any area where the knowledge is represented in a tree-structured form. Many powerful tree-mining algorithms have been developed to aid in structural comparisons, association rule discovery and in general, for mining of tree structured knowledge representations. Our work in the area of frequent subtree mining is characterized by adopting a Tree Model Guided (TMG) candidate generation [8] as opposed to the join approach, which is commonly used. This non-redundant systematic enumeration model ensures only valid candidates are generated which conform to the actual tree structure of the data.

Here we apply the MB3 algorithm to the Prions database of PO [9] in order to extract the frequently occurring subtrees. Prion (short for proteinaceous infectious particle) is a type of infectious agent. Prions are abnormally structured forms of host protein, which are able to convert normal molecules of protein into abnormally structured form. Prions dataset describes Protein Ontology (PO) database for Human Prion proteins in XML format [4, 10]. The experiments were run on 3Ghz (Intel-CPU), 2Gb RAM, Mandrake 10.2 Linux machine and compilation was performed using GNU g++ (3.4.3) with -g and -O3 parameters. Occurrence-match support definition was used. The total run-time and memory usage of the MB3 algorithm is displayed in Figure 1, for varying support thresholds.
Figure 1: MB3$^R$ run-time and memory usage profile

We also used some standard hierarchical and tree mining algorithms [8] on the PO instance store. We compared our MB3-Miner (MB3) algorithm with X3-Miner (X3), VTreeMiner (VTM) and PatternMatcher (PM) for mining embedded subtrees and our IMB3-Miner (IMB3) with FREQT (FT) for mining induced subtrees of PO instance store. Figure 2 shows the time performance of different algorithms. Our original MB3 has the best time performance for this data.
Quite interestingly, with Prion dataset of PO the number of frequent candidate subtrees generated is identical for all algorithms (Figure 3). Another observation is that when support is less than 10, PM aborts and VTM performs poorly. The rationale for this could be because the utilized join approach enumerates additional invalid subtrees. Note that original MB3 is faster than IMB3 due to additional checks performed to restrict the level of embedding.
4 Trustworthy Protein Ontology

Here we describe a conceptual framework that we are working on, to engineer Trustworthy Protein Ontology [11]. It is termed as 'Trustworthy Protein Ontology' as the final engineered ontology is trustworthy in the sense that it is accurate and precise. The final engineered ontology does not contain any redundant, inconsistent, and incorrect data or relationships.

Consider the scenario where we have 'N' Research Assistants. Each of these Research Assistants enters the data into an Intermediate Protein Ontology (IPO). IPO is mirror of the Original PO and contains same concepts in an exactly similar structured hierarchy as PO. However the research assistants may not be necessarily the experts in field of proteomics for which the ontology is being engineered. Hence we propose that instead of allowing research assistants to make changes directly to the Original PO, changes should be entered into the IPO. PO administrator then goes through IPO to check if the concepts, relationships and instances entered by research assistants. PO administrator is a person who is an expert in the field of proteomics for which trustworthy PO is engineered. PO administrator has knowledge about data formats of diverse protein data and knowledge sources. After research assistants enter the data in IPO, PO administrator goes through IPO in order skim out concepts, relationships and instances, which are redundant, inconsistent, and incorrect. This is done by running syntax and semantic checks on IPO, to check its validity in regards to concepts, relationships and instances already present in Original PO. There are two ways in which PO administrator may choose to skim through IPO.

**Method 1:** PO administrator goes through the whole IPO to which changes have been submitted by the Research Assistants to determine those concepts, relationships and instances which are redundant, inconsistent, and incorrect. PO administrator then removes or fixes these concepts, relationships and instances to create the final engineered IPO. Once all discrepancies have been removed from the final engineered IPO, and it has been checked for validity with the Original PO, all the changes made to IPO are integrated into the Original PO. This method compares structure and relationships of IPO and Original PO. This method is tedious...
and requires a lot of time and effort by the PO administrator. PO administrators can alternatively choose Method 2 as a means to engineer trustworthy ontology, which is quick, effective and does all the checks.

**Method 2:** PO administrator uses an administration console to skim through IPO using a defined set of rules that denotes what a correct concept would be, what a correct relationship between those concepts would be and what a correct instance of the concept would be. These set of rules utilize structure and semantics of PO to facilitate validation of any changes made to IPO by research assistants. PO structured vocabulary briefly outlined in Section 2 has 92 pre-defined concepts that belong to set of valid concepts, SET V. Of these 92 concepts, 12 concepts are necessary to define the basic information to enter protein complex data into the PO framework. These mandatory concepts belong to SET M. SET M is a subset of SET V. Semantic Relationships among the concepts of PO framework are discussed in Section 3. These Semantic Relationships belong to set of valid relationships, SET R.

To run structure and semantic checks using this method is followed:

1. For a concept entered in IPO by research assistants to be valid (c) it should be within the scope of SET V and must belong to SET M.

2. For a relationship entered in IPO by research assistants to be valid (r) it must belong to SET R.

3. Every tuple (c, r) in IPO belongs to a frameset F. These concepts and relationships are necessary and must be integrated with Original PO.

4. Every tuple (c', r) in IPO belongs to frameset F'. Here c' is a concept that does not belong to SET M. These concepts are checked further to see if they belong to SET V. If they do belong to SET V, then the tuple (c', r) is valid and must be integrated with Original PO.

5. All the tuples that do not belong to F and F' are discarded.
Thus, Method 2 is much quicker and efficient way to engineer a trustworthy PO, but it adds to the complexity of the algorithm. The approach proposed here for generating Trustworthy Protein Ontology is currently being implemented to provide a non-redundant, accurate and precise PO framework for future.

5 Concluding Remarks

Protein Ontology is a part of Standardized Biomedical Ontologies available through the National Center for Biomedical Ontologies [12] along with Gene Ontology [13], Flybase [14], and others (http://clioapprd.stanford.edu/ncbo/faces/pages/ontology_list.xhtml). More information about Protein Ontology can be found on Protein Ontology Website (http://www.proteinontology.info/). We are in process of adding Protein Ontology to Open Biomedical Ontologies or OBO (http://obo.sourceforge.net/).

Also different research groups are using Protein Ontology for different purposes. Y. Wang et al. [15] shows Protein Ontology as an example of a structured approach for knowledge modeling providing solid inference and retrieval functionalities. F. Porto [16] discusses Protein Ontology in his report under Ontologies for Bioinformatics. H. Tan et al. [8] use Protein Ontology generated data set to evaluate their algorithms. A. Kupfer [17] use Protein Ontology along with Gene Ontology to understand concepts when discussing a coevolution approach for database schemas. N. Bolshakova et al. [18] discuss protein ontology under a section on Biomedical Ontologies while comparing data based and ontology based approaches for cluster validation of microarrays. IQube [19] references Protein Ontology when reviewing Ontology Development in their white paper. L. Dhanapalan and Jake Y. Chen [20] discuss protein ontology in detail when doing case study of integrating protein interaction data using semantic web technology. K. Pinagé and V. Brilhante [21] used Protein Ontology for Protein Structure Homology Modeling. Just Recently researchers [22] discuss in detail Protein Ontology along with other major biomedical ontologies, while proposing a text mining based ontology construction methodology for Protein Data mainly for PIR database. A. Kupfer et al. [23] reuses the concept of chains from Protein Ontology when proposing database ontology for
signal transduction pathways. Lastly, Z. Lacroix et al. [24] discuss Protein Ontology briefly when proposing a semantic model to integrate biological resources.

References


