Ontology Algebra for Composition of Protein Data Sources

Amandeep S. Sidhu, Tharam S. Dillon, and Elizabeth Chang
Digital Ecosystems and Business Intelligence Institute,
Curtin University of Technology, Perth, Australia
{Amandeep.Sidhu, Tharam.Dillon, Elizabeth.Chang}@cbs.curtin.edu.au

Abstract

Most biological resources available today on the web provide a good number of cross-links to other resources with relevant information. However, in our opinion, what is still lacking is an integrated view that provides complete coverage of information through a single entry point. The main problem lies in interpreting biological nomenclature because the underlying data sources are inconsistent. In this paper we discuss Protein Ontology (PO) Algebra that we use for composition and interoperability of protein data sources. We outline the existing research in interoperability of biological data sources, before discussing our semantic interoperability approach in detail. The actual implementation of Protein Ontology is also discussed briefly in this paper, which depends on the strength of the Protein Ontology Algebra.

1. Introduction

Life scientists face many challenges in their task of managing biological information. Looking at the bioinformatics workflow at a low level, we observe that data from experiments is entered into databases, where scripts written in languages such as Perl are employed to filter and analyze the data and to compare them with other known samples. Results are then prepared as a combination of numerical data and prose for publication.

At a more conceptual level, the information that is used and generated by life scientists, including chemical pathways, annotated gene sequences, and protein structures, is highly connected in nature. For example, an enzyme that catalyzes some specific pathway has a specific, definite structure and genetic sequence that encodes how to construct it. Connections also exist between any given protein and other proteins that are similar to it, either in terms of functionality or composition. Historically, these different forms of information (e.g., annotations, pathways, structures, sequences, etc.) have been stored in a series of incompatible databases using disparate identifier schemes and distinct data formats. As a result, life scientists have been prevented from working with their information at the desired high level because these databases must be bridged, manipulated and normalized (usually by hand-written Perl scripts) to accomplish all but the simplest of tasks.

Protein Data and Information in structural bioinformatics is organized along the following lines. The structural information about protein molecules – the 3D atomic coordinates of structures – is the core from which all other details are derived; it is a primary resource of structural data and is central to everything else. The files containing atomic coordinates are uninformative to the majority of structural biologists; thus, there are algorithmic tools (applications) that transform, classify, analyze, and model this primary data. The results of the data analysis are often (but not always) stored in other databases, considered to be secondary resources, since they contain value-added information.

Even without a unified approach to biological information management as proposed in earlier works [1-3], one may still note that a huge wealth of information is available publicly. Despite data being universally accessible for any given research area, what distinguishes a novice from an expert is the knowledge of what information is relevant. Knowledge representation technology has enabled some of that expertise to be represented in a machine- readable form, such as with the Protein Ontology (PO) [4] and Gene Ontology (GO) [1]. When concepts are mapped in terms of ontologies, they are then searchable by the machine, potentially expanding the power of query engines. One can imagine a user aggregating several kinds of objects together—gene sequences, literature references, web pages, even the names of noted experts in the field—into custom, user- defined collections. Such collections represent valuable interpretations of

978-1-4244-1604-2/07/$25.00 ©2007 IEEE.

relevance and could then be shared, sent between colleagues, and searched.

Previous work into integrating biological information sources has concentrated on doing so for the purposes of resolving federated queries [5]. Query resolution requires a sophisticated level of mapping in design of domain ontologies in order to work across multiple, disparate databases. We try to address this need by presenting algebra in this paper that will enhance the power of query engine for PO. The focus of this paper is on how we can standardize the naming conventions and provide user interface mechanisms for RDF-encoded biological information; hence, the output of protein ontology-based query resolvers should be usable.

2. Graph Oriented Model for Protein Ontology

Most of the applications especially from Semantic Web Community assume the existence of an ontology that models the domain of the application in an integrated way. However, the majority of existing biomedical data sources are not modeled in a way that relates instances directly to ontology classes and properties like RDF [6] can do, but are modeled as relations or as XML [7]. These data models come with their own schema and SQL or XML Schema [8] and XQuery [9] respectively. We developed BIODB database as our initial work [10-12] using XML Schema, which contains information about Protein Structures in a way that represents the relationships between various Protein Structure elements. It takes into greater consideration formation of the ultimate protein conformation and the relationships that exist in the data, rather then just storing the data. Both SQL and XML representations were made available to the community for BIODB.

UniProt [13] is a comprehensive repository of protein sequence and annotation data. The number of known protein sequences has been increasing ever since the creation of the database. UniProt is working towards providing protein sequence and annotation data in RDF format. Challenges still exits to these approaches like similar concepts in different databases by other organizations need to be mapped to each other or unified. Also users must still have detailed knowledge of precisely what classes and properties are available. Given that even a single database may make use of dozens of concepts this is not a trivial task. Some kind of query tool that allows users to browse the schema and construct queries might help.

In order to access such sources via ontologies Semantic Web Query Language or SWQL [14] was developed. SWQL uses an XQuery-like syntax and is like XQuery a declarative, fully compositional, strictly typed functional query language. It supports navigation and selection via its sublanguage SQWLPath similar to XPath [15] and unlike most RDF query languages [16] it supports joins and construction. Although SWQL uses Web Ontology Language or OWL [17] for representation but it still uses concepts from earlier XQuery and XPath implementations. Recently developed OWL Query Language (OWL-QL) is a formal language and protocol for a querying agent and an answering agent to use in conducting a query-answering dialogue using knowledge represented in the OWL. OWL-QL is an updated version of the DAML Query Language or DQL [18].

There still need for a query framework for representing vocabularies forming ontologies as tree structured items to enhance performance and efficiency of data mining algorithms that are used for bioinformatics. Our common conceptual model for the internal representation of PO is based on the work done by Gyssens et al. [19]. In its core, we represent protein ontology as a graph. Following definition of graph is used:

Definition 1

- Ontology O = (G, RL) is represented as a directed labeled graph G and a set of rules RL.
- 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. Label of a node often represents a concept defined in the ontology. Label of an edge is the name of a semantic relationship among the concepts in the ontology and can be null if the relationship is not known. R denotes relationships in our graph model.

Set of logical rules RL, associated with protein ontology, are rules expressed in a logic-based language used to derive data from existing protein data and information sources. We discuss these rules in the next section in context of N, E, and R.

3. Protein Ontology Algebraic Operators

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

978-1-4244-1604-2/07/$25.00 ©2007 IEEE.

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 two unary operators: *Select, Projection and three binary operators: Intersection, Union and Difference*.

Select and Projection Operators

The Select Operator allows us to highlight and select portions of the PO that are relevant to the query at hand. Given the PO structure and a concept to be selected, the select operator selects the instances satisfying the given condition. These instances, which satisfy the given condition, would belong to particular sub trees or are the subset of the instances that belong to one or more sub trees. The Select Operator selects only those edges in the PO that connect nodes in a given set. The Select Operator is defined as:

Definition 2

OS = σ *(NS, ES, RS) where*
NS = Nodes (condition = true)
ES = Edges ($\forall N \in NS$*)*

Here N, E, R represent a set of nodes, edges and relationships for the Protein Ontology graph and NS, ES, RS represent the nodes, edges and relationships of the selection set respectively. We don't discuss the join condition operator here in Protein Ontology as the Protein Ontology Conceptual Framework is structured well and the Select Operator can be used in following forms:

- Simple-Condition: Where the select condition is specified using the simple content types like Generic Concepts in PO and the select operator is value-based;
- Complex-Condition: Where the select condition is specified using complex content types like Derived Concepts in PO and the select operator is structure-based; and,
- Pattern-Condition: Where the select condition is specified using a mix of simple and/or complex content types in the hierarchy with additional constraints, where the select operator is pattern-based.

The Projection Operator allows us to produce a result for a given context (number of attributes of nodes denoted by n) where it has only specified items specified in the item set NP (where, NP $\subset$ n) with: (a) preserved node hierarchy, (b) preserved node order and (c) preserved semantic relationships.

Definition 3

OP = Π *(NP, EP, RP) where*
NP $\subset$ n
EP = Edges ($\forall N \in NP$*)*

Here N, E, R represent a set of nodes, edges and relationships for the Protein Ontology graph and NP, EP, RP represent the nodes, edges and relationships of the selection set respectively. Now we consider the Description of Protein Families in PO to demonstrate the usage of both the Select and Project Operators.

Example 1

Let us consider that a user requires all the information available in the PO in regards to Protein Families. In this case, all the instances of Family Concept are displayed and the SELECT operator is used for this purpose.

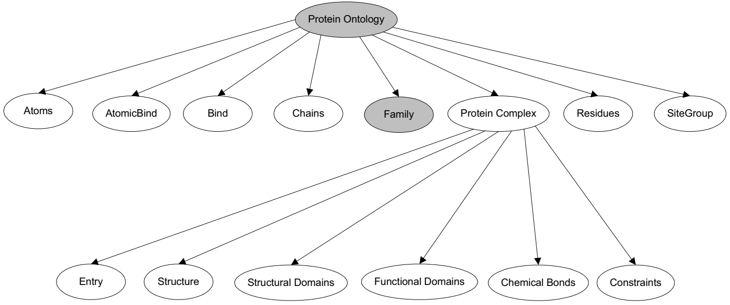


Figure 1: Depicting SELECT Operations

On the other hand, if the user specifically wants all instances of Family Concept in PO but display only in the context of ProteinFamily and ProteinSuperFamily, then the PROJECTION operator is used.

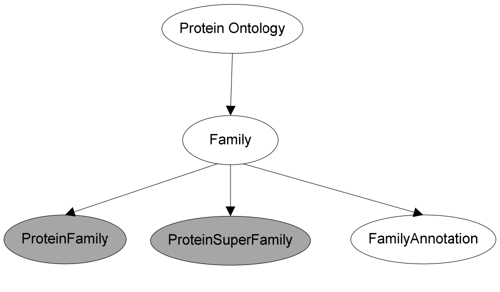


Figure 2: Depicting PROJECTION Operations

978-1-4244-1604-2/07/$25.00 ©2007 IEEE.

Intersection Operator

Intersection is an important and interesting binary operation. Let *O1 = (N1, E1, R1), and O2 = (N2, E2, R2)* be the two parts of PO whose composition will provide the 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 Relationships. The intersection of two parts of the PO with respect to the semantic relationships (SR) of PO is:

Definition 4

$OI\ (1, 2) = O1 \cap_{SR} O2 = (NI, EI, RI)$, *where*
$NI = Nodes\ (SR\ (O1, O2))$,
$EI = Edges\ (E1, NI \cap N1) + Edges\ (E2, NI \cap N2) + Edges\ (SR\ (O1, O2))$, *and*
$RI = Relationships\ (O1, NI \cap N1) + Relationships\ (O2, NI \cap N2) + SR\ (O1, O2) – Edges\ (SR\ (O1, O2))$.

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 modelled as edges and those relationships present in source parts of the ontology that use only concepts that occur in the intersection ontology.

Example 2

Let us consider that a query requires all the information that is common between Protein Entry and Structure descriptions available in the PO. In this case, all the information highlighted in Figure 3 is displayed. Only the ChainRef is common between Entry and Structure. INTERSECTION operation is used for this purpose (Entry ∩ Structure). ChainRef refers to the generic concept of Chains.

Union Operator

The union of two parts of PO, *O1 = (N1, E1, R1), and O2 = (N2, E2, R2)* is expressed as:

Definition 5

$OI\ (1, 2) = O1 \cup_{SR} O2 = (NU, EU, RU)$, *where,*
$NU = N1 \cup N2 \cup NI\ (1, 2)$,
$EU = E1 \cup E2 \cup EI\ (1, 2)$, *and*
$RU = R1 \cup R2 \cup RI\ (1, 2)$, *where,*
$OI\ (1, 2) = O1 \cap_{SR} O2 = (NI\ (1, 2), EI\ (1, 2), RI\ (1, 2))$ *is the intersection of two ontologies.*

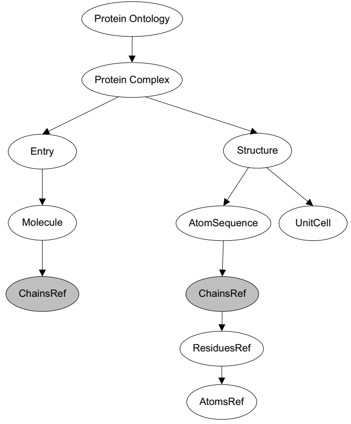


Figure 3: Depicting INTERSECTION Operations

The union operation combines two parts of the ontology retaining only one copy of the concepts in the intersection. Here N, E, R represent set of nodes, edges and relationships for the Protein Ontology graph and NU, EU, RU represent the nodes, edges and relationships of the selection set respectively.

Example 3

Let us consider that a user requires all the information available in the PO in regards to Protein Families and the Protein Structure. In this case, all the information highlighted in Figure 4 is displayed. The UNION operator is used for this purpose (Family ∪ Structure).

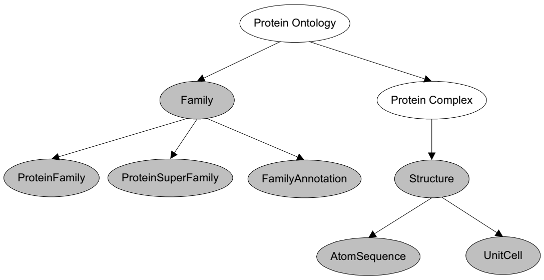


Figure 4: Depicting UNION Operations

Difference Operator

The difference of two parts of PO, O1 and O2, written as O1 – O2, includes portions of the first part that are not common to the second part. The difference

978-1-4244-1604-2/07/$25.00 ©2007 IEEE.

can be rewritten as $O1 - (O1 \cap_{SR} O2)$. The nodes, edges and relationships that are not in the intersection but are present in the first part comprise the difference.

Example 4

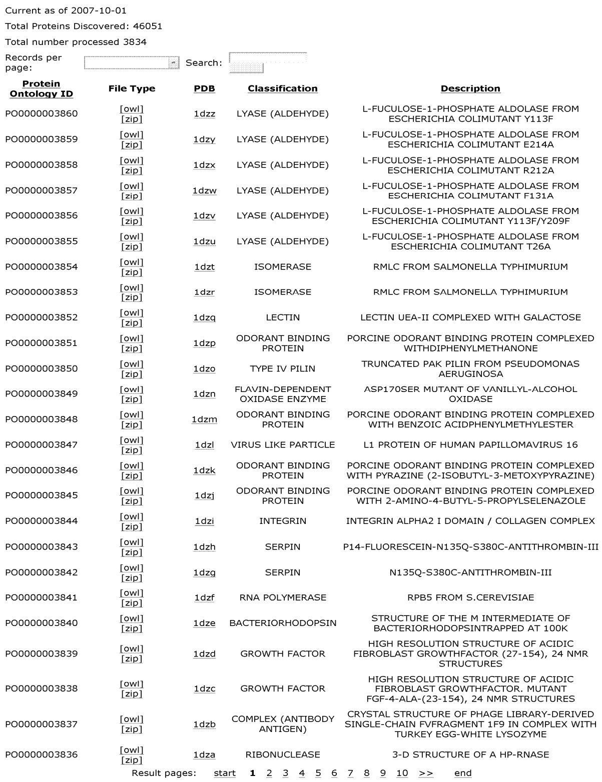
Let us consider a query that requires all the information about the Protein Entry excluding the commonality between Protein Entry and Structure descriptions available in the PO. In this case, all the information that is not highlighted for Protein Entry in Figure 3 is displayed. As only the ChainRef is common between Entry and Structure, all other information except ChainRef is displayed for Entry using DIFFERENCE operation (Entry - (Entry ∩ Structure)).

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

4. Protein Ontology Implementation

The process of acquiring data and knowledge from proteomics domain is the first Stage of the development, which applies algorithms and methods analyzing protein data files and proteomics domain texts. The terminology used by domain experts is defined in protein ontology. In this study, to collect a glossary of concepts (classes) for the proteomics domain, first, an analysis was performed on 4 major protein data sources: PDB [20], SCOP [21], SWISS-PROT [22] and PIR [23]. An interface constructed using Java is used to parse the data from various protein data sources and unify them in the PO format **(Figure 5)**. Protein data is parsed according OWL schema specifications.

In Stage 2 Protein Ontology is formally represented using an ontology language [24] such as Ontology Web Language (OWL) or Resource Description Format (RDF). This stage involves the formalization of each term and the constraints used by the ontology. Some of the formalisms not provided by OWL in which Protein Ontology are defined using Protein Ontology Algebra. Terms are represented through classes, relations, functions, and instances. Queries to extract Protein Ontology Concepts and Instances are also formulated using Protein Ontology Specification **(Figure 6)**.



Current as of 2007-10-01

Total Proteins Discovered: 46051

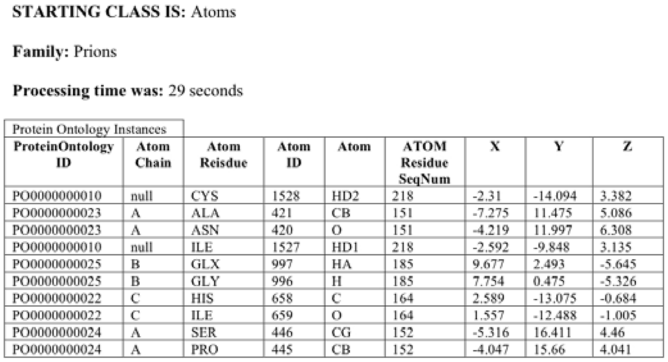
Total number processed 3834

Records per page: Search:

Protein Ontology ID	File Type	PDB	Classification	Description
PO0000003860	[owl] [zip]	1dzz	LYASE (ALDEHYDE)	L-FUCULOSE-1-PHOSPHATE ALDOLASE FROM ESCHERICHIA COLIMUTANT Y113F
PO0000003859	[owl] [zip]	1dzy	LYASE (ALDEHYDE)	L-FUCULOSE-1-PHOSPHATE ALDOLASE FROM ESCHERICHIA COLIMUTANT E214A
PO0000003858	[owl] [zip]	1dzx	LYASE (ALDEHYDE)	L-FUCULOSE-1-PHOSPHATE ALDOLASE FROM ESCHERICHIA COLIMUTANT R212A
PO0000003857	[owl] [zip]	1dzw	LYASE (ALDEHYDE)	L-FUCULOSE-1-PHOSPHATE ALDOLASE FROM ESCHERICHIA COLIMUTANT F131A
PO0000003856	[owl] [zip]	1dzv	LYASE (ALDEHYDE)	L-FUCULOSE-1-PHOSPHATE ALDOLASE FROM ESCHERICHIA COLIMUTANT Y113F/Y209F
PO0000003855	[owl] [zip]	1dzu	LYASE (ALDEHYDE)	L-FUCULOSE-1-PHOSPHATE ALDOLASE FROM ESCHERICHIA COLIMUTANT T26A
PO0000003854	[owl] [zip]	1dzt	ISOMERASE	RMLC FROM SALMONELLA TYPHIMURIUM
PO0000003853	[owl] [zip]	1dzr	ISOMERASE	RMLC FROM SALMONELLA TYPHIMURIUM
PO0000003852	[owl] [zip]	1dzq	LECTIN	LECTIN UEA-II COMPLEXED WITH GALACTOSE
PO0000003851	[owl] [zip]	1dzp	ODORANT BINDING PROTEIN	PORCINE ODORANT BINDING PROTEIN COMPLEXED WITHDIPHENYLMETHANONE
PO0000003850	[owl] [zip]	1dzo	TYPE IV PILIN	TRUNCATED PAK PILIN FROM PSEUDOMONAS AERUGINOSA
PO0000003849	[owl] [zip]	1dzn	FLAVIN-DEPENDENT OXIDASE ENZYME	ASP170SER MUTANT OF VANILLYL-ALCOHOL OXIDASE
PO0000003848	[owl] [zip]	1dzm	ODORANT BINDING PROTEIN	PORCINE ODORANT BINDING PROTEIN COMPLEXED WITH BENZOIC ACIDPHENYLMETHYLESTER
PO0000003847	[owl] [zip]	1dzl	VIRUS LIKE PARTICLE	L1 PROTEIN OF HUMAN PAPILLOMAVIRUS 16
PO0000003846	[owl] [zip]	1dzk	ODORANT BINDING PROTEIN	PORCINE ODORANT BINDING PROTEIN COMPLEXED WITH PYRAZINE (2-ISOBUTYL-3-METOXYPYRAZINE)
PO0000003845	[owl] [zip]	1dzj	ODORANT BINDING PROTEIN	PORCINE ODORANT BINDING PROTEIN COMPLEXED WITH 2-AMINO-4-BUTYL-5-PROPYLSELENAZOLE
PO0000003844	[owl] [zip]	1dzi	INTEGRIN	INTEGRIN ALPHA2 I DOMAIN / COLLAGEN COMPLEX
PO0000003843	[owl] [zip]	1dzh	SERPIN	P14-FLUORESCEIN-N135Q-S380C-ANTITHROMBIN-III
PO0000003842	[owl] [zip]	1dzg	SERPIN	N135Q-S380C-ANTITHROMBIN-III
PO0000003841	[owl] [zip]	1dzf	RNA POLYMERASE	RPB5 FROM S.CEREVISIAE
PO0000003840	[owl] [zip]	1dze	BACTERIORHODOPSIN	STRUCTURE OF THE M INTERMEDIATE OF BACTERIORHODOPSINTRAPPED AT 100K
PO0000003839	[owl] [zip]	1dzd	GROWTH FACTOR	HIGH RESOLUTION STRUCTURE OF ACIDIC FIBROBLAST GROWTHFACTOR (27-154), 24 NMR STRUCTURES
PO0000003838	[owl] [zip]	1dzc	GROWTH FACTOR	HIGH RESOLUTION STRUCTURE OF ACIDIC FIBROBLAST GROWTHFACTOR. MUTANT FGF-4-ALA-(23-154), 24 NMR STRUCTURES
PO0000003837	[owl] [zip]	1dzb	COMPLEX (ANTIBODY ANTIGEN)	CRYSTAL STRUCTURE OF PHAGE LIBRARY-DERIVED SINGLE-CHAIN FVFRAGMENT 1F9 IN COMPLEX WITH TURKEY EGG-WHITE LYSOZYME
PO0000003836	[owl] [zip]	1dza	RIBONUCLEASE	3-D STRUCTURE OF A HP-RNASE

Result pages: start **1** 2 3 4 5 6 7 8 9 10 >> end

Figure 5: Snapshot of PO Instances



STARTING CLASS IS: Atoms

Family: Prions

Processing time was: 29 seconds

Protein Ontology Instances

ProteinOntology ID	Atom Chain	Atom Reisdue	Atom ID	Atom	ATOM Residue SeqNum	X	Y	Z
PO0000000010	null	CYS	1528	HD2	218	-2.31	-14.094	3.382
PO0000000023	A	ALA	421	CB	151	-7.275	11.475	5.086
PO0000000023	A	ASN	420	O	151	-4.219	11.997	6.308
PO0000000010	null	ILE	1527	HD1	218	-2.592	-9.848	3.135
PO0000000025	B	GLX	997	HA	185	9.677	2.493	-5.645
PO0000000025	B	GLY	996	H	185	7.754	0.475	-5.326
PO0000000022	C	HIS	658	C	164	2.589	-13.075	-0.684
PO0000000022	C	ILE	659	O	164	1.557	-12.488	-1.005
PO0000000024	A	SER	446	CG	152	-5.316	16.411	4.46
PO0000000024	A	PRO	445	CB	152	-4.047	15.66	4.041

Figure 6: Sample PO Query Results

5. Conclusions

In this paper we covered 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 sources themselves. PO algebra based on semantic relationships allows systematic composition, which unlike integration is more scalable.

References

[1] M. Ashburner, C. A. Ball, J. A. Blake, H. Butler, J. C. Cherry, J. Corradi, and K. Dolinski, "Creating the Gene Ontology Resource: Design and Implementation," *Genome Research*, vol. 11, pp. 1425-1433, 2001.

[2] B. A. Eckman, Z. Lacroix, and L. Raschid, "Optimized seamless integration of biomolecular data," presented at IEEE 2nd International Symposium on Bioinformatics and Bioengineering Conference, Bethesda, MD, USA, 2001, pp. 23-32.

[3] A. S. Sidhu, T. S. Dillon, and E. Chang, "Integration of Protein Data Sources through PO," presented at 17th International Conference on Database and Expert Systems Applications (DEXA 2006), Poland, 2006, pp. 519-527.

[4] A. S. Sidhu, T. S. Dillon, and E. Chang, "Ontological Foundation for Protein Data Models," presented at 1st IFIP WG 2.12 & WG 12.4 International Workshop on Web Semantics (SWWS 2005), In conjunction with On The Move Federated Conferences (OTM 2005), Agia Napa, Cyprus, 2005, pp. 916-925.

[5] P. G. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens, "TAMBIS - transparent access to multiple bioinformatics information sources," presented at 6th International Conference on Intelligent Systems for Molecular Biology, Montreal, Canada, 1998, pp. 25-34.

[6] W3C-RDF, "RDF Primer," in *W3C Recommendation 10 February 2004*, F. Manola, E. Miller, and B. McBride, Eds.: World Wide Web Consortium, 2004.

[7] W3C-XML, "Extensible Markup Language (XML) 1.0," in *W3C Recommendation 16 August 2006, edited in place 29 September 2006*, T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, Eds., 4th ed: World Wide Web Consortium, 2006.

[8] W3C-XMLSchema, "XML Schema Part 0: Primer," in *W3C Recommendation 28 October 2004*, D. C. Fallside and P. Walmsley, Eds., 2nd ed: World Wide Web Consortium, 2004.

[9] W3C-XQuery, "XQuery 1.0: An XML Query Language," in *W3C Proposed Recommendation 21 November 2006*, S. Boag, D. Chamberlin, M. F. Fernández, D. Florescu, J. Robie, and J. Siméon, Eds.: World Wide Web Consortium, 2006.

[10] A. S. Sidhu, T. S. Dillon, H. Setiawan, and B. S. Sidhu, "Comprehensive Protein Database Representation," presented at 8th International Conference on Research in Computational Molecular Biology 2004 (RECOMB 2004), San Diego, California, 2004, pp. 427-429.

[11] A. S. Sidhu, T. S. Dillon, B. S. Sidhu, and H. Setiawan, "A Unified Representation of Protein Structure Databases," in *Biotechnological Approaches for Sustainable Development*, M. S. Reddy and S. Khanna, Eds. India: Allied Publishers, 2004, pp. 396-408.

[12] A. S. Sidhu, T. S. Dillon, B. S. Sidhu, and H. Setiawan, "An XML based semantic protein map," presented at 5th International Conference on Data Mining, Text Mining and their Business Applications (Data Mining 2004), Malaga, Spain, 2004, pp. 51-60.

[13] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, and L. S. Yeh, "UniProt: The Universal Protein knowledgebase.," *Nucleic Acids Research*, vol. 32, pp. 115-119, 2004.

[14] P. Lehti and P. Frankhauser, "SWQL - A Query Language for Data Integration Based on OWL," presented at First IFIP WG 2.12 & WG 12.4 International Workshop on Web Semantics (SWWS 2005) in conjunction with OTM 2005., Agia Napa, Cyprus, 2005, pp. 926-935.

[15] W3C-XPath, "XML Path Language (XPath) Version 1.0," in *W3C Recommendation 16 November 1999*, J. Clark and S. DeRose, Eds.: World Wide Web Consortium, 1999.

[16] P. Haase, J. Broekstra, A. Eberhart, and R. Volz, "A comparison of rdf query languages," presented at 3rd International Semantic Web Conference, Hiroshima, Japan, 2004, pp. 502-517.

[17] W3C-OWL, "OWL Web Ontology Language Overview," in *W3C Recommendation 10 February 2004*, D. L. McGuinness and F. Harmelen, Eds.: World Wide Web Consortium, 2004.

[18] DQL, "DAML Query Language (April 2003)," in *http://www.daml.org/2003/04/dql/ - DQL released by the Committee on 1 April 2003. It replaces the DQL (August 2002) release.*, R. Fikes, P. Hayes, I. Horrocks, H. Boley, M. Dean, B. Grosof, F. van Harmelen, S. Hawke, J. Heflin, O. Lassila, D. L. McGuinness, P. F. Patel-Schneider, and L. Stein, Eds.: Joint US/EU ad hoc Agent Markup Language Committee, 2003.

978-1-4244-1604-2/07/$25.00 ©2007 IEEE.

[19] M. Gyssens, P. Paredaens, and D. Gucht, "A graph-oriented object database model," presented at 9th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, Nashville, Tennessee, 1990, pp. 417-424.

[20] J. Westbrook and P. M. D. Fitzgerald, "The PDB format, mmCIF formats and other data formats," in *Structural Bioinformatics*, P. E. Bourne and H. Weissig, Eds. Hoboken, NJ: John Wiley & Sons, Inc., 2003, pp. 161-179.

[21] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A Structural Classification of Proteins Database for the Investigation of Sequences and Structures," *Journal of Molecular Biology*, vol. 247, pp. 536-540, 1995.

[22] A. Bairoch and R. Apweiler, "The SWISS-PROT protein sequence data bank and its supplement TrEMBL," *Nucleic Acids Research*, vol. 25, pp. 31-36, 1997.

[23] W. C. Barker, J. S. Garavelli, D. H. Haft, L. T. Hunt, C. R. Marzec, and B. C. Orcutt, "The PIR-International Protein Sequence Database," *Nucleic Acids Research*, vol. 26, pp. 27-32, 1998.

[24] A. S. Sidhu, T. S. Dillon, E. Chang, and B. S. Sidhu, "Protein Ontology Development using OWL," presented at 1st Workshop on OWL: Experiences and Directions (OWLED 2005), Galway, Ireland, 2005, pp. 6 pages.