ICBO: International Conference on Biomedical Ontology
July 28-30, 2011 · Buffalo, NY, USA

Towards a Semantic Web Application: Ontology-Driven Ortholog Clustering Analysis

Yu Lin, Zuoshuang Xiang, Yongqun He

Center for Computational Medicine and Bioinformatics, Unit of Laboratory Animal Medicine, and Department of Microbiology and Immunology, University of Michigan Medical School, Ann Arbor, MI, USA

Abstract. Ontology is the foundation of Semantic Web applications. The Clusters of Orthologous Groups (COG) system uses evolutionary relationships to cluster proteins from different genomes into different functional categories. In this study, we generated a COG Analysis Ontology (CAO), and used it to develop OntoCOG, an ontology-based Semantic Web application for COG-based gene set enrichment analysis. As a use case, OntoCOG is applied to a list of *B. melitensis* virulence factors retrieved from the Brucellosis Ontology (BO). This OntoCOG analysis confirms and expands current knowledge about *B. melitensis* virulence factors.

Keywords: Semantic Web, Clusters of Orthologous Groups, COG, gene set enrichment analysis, *Brucella* virulence factors

1 Introduction

The Semantic Web is a group of methods and technologies designed to allow machines to understand the meaning – or "semantics" – of information on the World Wide Web (WWW). It comprises standards and tools associated with XML, XML Schema, RDF, RDF Schema and OWL and organized in the Semantic Web Stack. During the last decade the number and scope of Semantic Web applications has remarkably increased.

Ontologies are consensus-based controlled vocabularies of terms and relations with associated definitions, which are logically formulated to promote automated reasoning. In biomedicine, ontologies play important roles such as: (a) knowledge management, including the indexing and retrieval of data and information; (b) data integration, exchange and semantic interoperability; and (c) decision support and reasoning [1]. Machine-readable ontologies play a fundamental role in Semantic Web applications in ensuring that computers can understand the semantics of terms.

The relationships between genes from different genomes are naturally represented as a system of homologous families that include both orthologs and paralogs. Orthologs are genes in different species that have evolved from a common ancestral gene by speciation [2]. Orthologs usually share the same functions in the course of evolution. Therefore, identification of orthologs is critical for reliable prediction of gene function in newly sequenced genomes. The Clusters of Orthologous Groups (COG) database (http://www.ncbi.nlm.nih.gov/COG/) provides a system designed to classify proteins in terms of orthologous relationships based on comparative genomic study [3, 4]. The authors of COG database defined the COGs of proteins by strictly applying all against all BLAST alignments of protein sequences from completely sequenced microbial genomes [5]. Each protein in COG database has been assigned a COG ID, and further clustered into 25 COG functional categories. These 25 Functional Categories belong to four divisions, namely: *Information storage and processing*, *Cellular processes and signaling*, *Metabolism,* and *Poorly characterized*. The COG assignment thus falls into a hierarchy fashion.

Similar with the Gene Ontology (GO; http://www.geneontology.org/), the COG categories can be used to perform functional analysis, i.e., COG-based gene set enrichment analysis. A COG-based gene set enrichment analysis (in short, COG enrichment analysis) serves to identify COG terms that are enriched to a statistically significant degree

among a given list of proteins compared to the distribution of these terms within the organism. Specifically, given a list of k COG annotated proteins with a total of t proteins from one organism. For a given COG category *catA*, there are q proteins within k and m proteins within t associated with it. The data will look like this in a 2×2 table:

	Given list	Not given list	Total
catA	*q*	*m-q*	*M*
non-catA	*k-q*	*t-m-(k-q)*	*t-m*
total	*K*	*t-k*	*T*

The COG enrichment analysis is to find out the statistical significance of the distribution of the data, particularly, the p-value to test whether COG category *catA* annotated protein q is enriched (unevenly distributed) among the given protein list *t*.. A statistical method, for example, Fisher's exact test, Chi squared test, or hyper geometric test, can be used depending on the sample size of the dataset.

There is no platform independent COG enrichment analysis package available yet, here we introduce a Semantic Web application OntoCOG, which is an ontology-oriented COG-based gene enrichment analysis. OntoCOG has a simple interface for scientists to process their data and return a result of COG enrichment analysis in OWL format. The COG enrichment analysis of *Brucella* virulence factors is used as an example to demonstrate the OntoCOG system, including the COG Analysis Ontology (CAO), an essential part for the OntoCOG design and construction. This project provides a clear demonstration on how an important biomedical question can be addressed using ontology-based Semantic Web technology.

2 Methods

2.1 OntoCOG Design and System Architecture

OntoCOG is designed as a Semantic Web service application for COG enrichment analysis. The OntoCOG software takes a given list of protein identifiers as input, performs statistical COG enrichment analysis, and returns COG analysis results as output using RDF/XML format that is modeled by CAO. In OntoCOG, the data obtained from the COG database is stored locally in an OntoCOG relational database management system (RDMS). Each time a user sends requirements through the interface; OntoCOG will retrieve the COG annotation from RDMS and then transform the data set into a RDF/XML file. An OWL reasoner will then be applied to check the consistency and if needed remove duplicated or invalid data. Meanwhile, statistical calculation for the COG enrichment measurement will be performed, and the result is transformed into RDF based on the COG Analysis Ontology (CAO). Output data is also available in plain text format (Fig. 1).

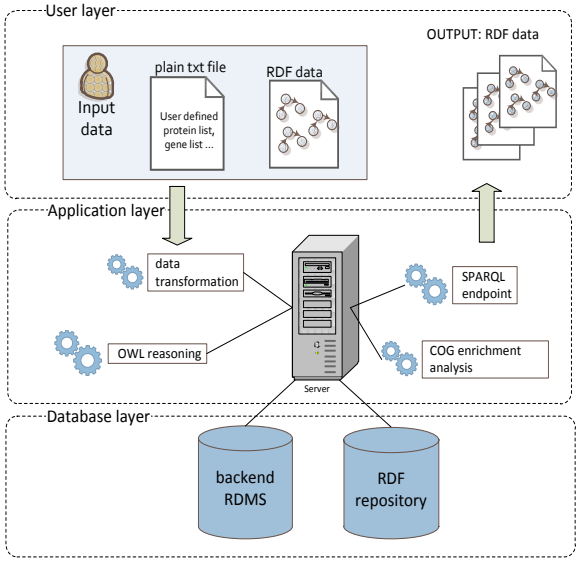

Figure 1. Design & Architecture of OntoCOG.

A conventional three-tier architecture is used for the OntoCOG system development (Fig. 1). For the user layer, a user presents data (plain text data or RDF data) or analysis queries using a front-end web browser via an HTML form or by uploading input data as a file through the interface. The middle tier, also called the application layer, extracts the input data from the user layer and implements the application's functionality. Basic functions in OntoCOG include: data transformation, OWL reasoning, COG enrichment analysis, and SPARQL data retrieval. These processes are executed with PHP scripts against the OntoCOG relational database and any publicly available RDF repository (back-end, database server). The result of each query is presented to a user

through the web browser using HTML and RDF format.

2.2 Development of the COG Analysis Ontology (CAO)

The CAO is developed based on the Semantic Web application's needs. The scopes of CAO include: 1) ontology-based software/service design; 2) supporting data integration and exchange in OWL format. The domains covered by CAO include statistical analysis and protein's COG annotation. OWL is the default format for CAO development. CAO was edited using Protégé 4.1 Beta (build 218) as ontology editor. CAO fully imports the Basic Formal Ontology (BFO; http://www.ifomis.org/bfo/) as its top ontology and the Relation Ontology (RO; http://www.obofoundry.org/ro/) as a collective of core relations. OntoFox, an ontology development tool for importing external terms from existing ontologies [6], was used to import the following groups of ontology terms: (a) statistical analysis related terms, such as use curly quotes Fisher's exact test and Chi square test, from the Ontology for Biomedical Investigation (OBI) [7] ; (b) informatics related terms, such as data item and data set, from the Information Artifact Ontology (IAO) [8]; and (c) Organism terms from NCBITaxon [9].

3 Results

3.1 COG Analysis Ontology (CAO)

The CAO source code is available in sourceforge (http://cao.svn.sourceforge.net/). All the classes in CAO fall into three top classes: 1) *data transformation,* subclass of *planned process*; 2) *material entity*, subclass of *independent continuant*; and 3) *information content entity*, subclass of *generically dependent continuant* (Fig. 2).

The version 1.0 of CAO contains 178 CAO specific terms. CAO supports the design of the OntoCOG web services in aspects of data input and output flow, data modeling, and logic processing of the whole system (Fig. 3). The ontology term *OntoCOG*, asserted as a subclass of *software (IAO_0000010)*, has *COG enrichment analysis data transformation objective* as its objective specification part.

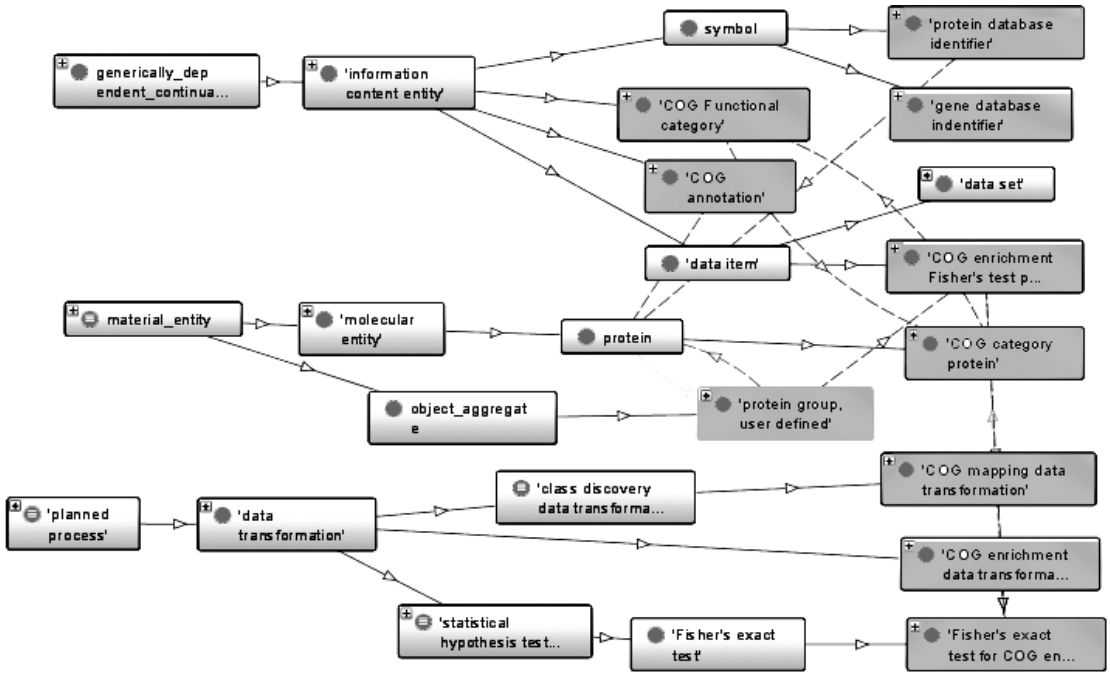

Figure 2. Key terms in the CAO hierarchy.
Gray boxes contain specific CAO terms. The remaining boxes contain terms derived from external ontologies.

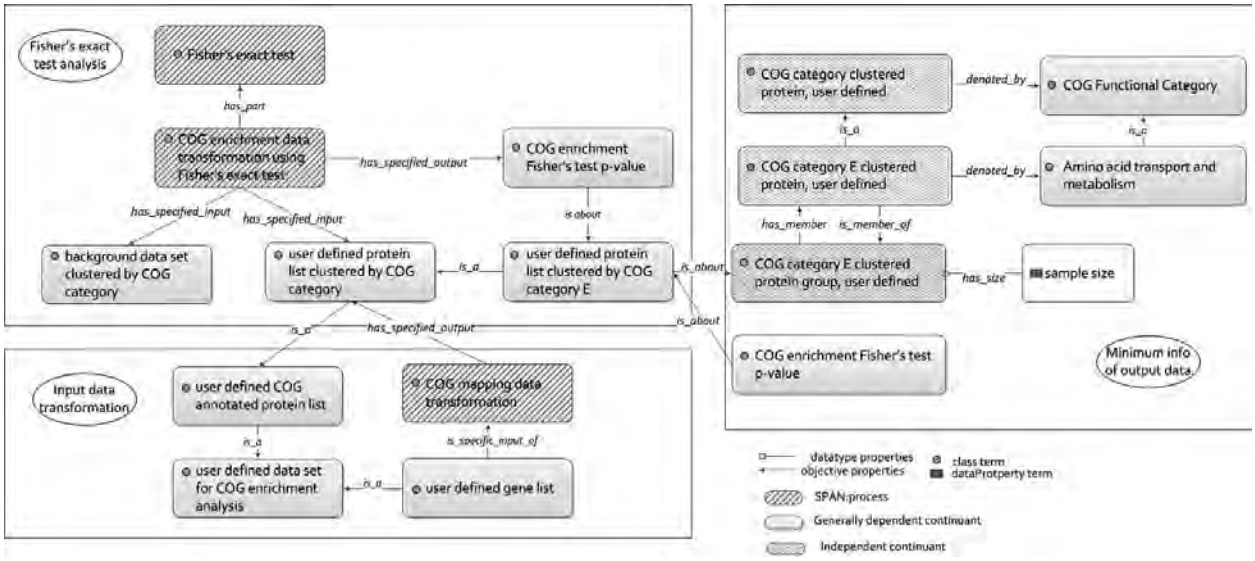

Figure 3. Design of CAO under the scope of the OntoCOG application.
CAO includes models for major components of the OntoCOG application: input data transformation, Fisher's exact Test analysis, and minimum information of output data. Terms in boxes with lines, white boxes, and boxes with dots denote *processes*, *generically dependent continuants,* and *independent continuants,* respectively.

Modeling OntoCOG input data transformation in CAO. The input dataset of OntoCOG is a list of proteins or gene identifiers submitted by a user. CAO uses IAO term *symbol* to represent the protein identifiers and gene identifiers. Both *gene database identifier* and *protein database identifier* are subclasses of *symbol*. *NCBI protein GI* and *NCBI protein accession* are subclasses of *protein database identifier*. The IAO relation *is about* has been used here to denote the relation between *information content entity* and the *material entity*. For example, a *NCBI protein GI* is about a *protein* that is a *material entity*.

By default, the input of OntoCOG is a list of NCBI protein GIs. Users may also submit a list of NCBI protein accessions or NCBI gene GIs. The server of OntoCOG will map those lists to their NCBI protein GIs, and then map the protein list with the backend COG database installed in the server. CAO models this process as a *COG mapping data transformation* process, which has a *user-defined protein (*or *user defined gene list)* as a specific input, and a protein list assigned by COG categories as one of the two specific outputs. Another specific output of this process is the sub list of proteins grouped by each COG category.

Modeling OntoCOG Fisher's exact analysis in CAO. In the CAO ontology, *COG enrichment data transformation using Fisher's exact test* is a subclass of *data transformation* (OBI_0200000). Two specified inputs are participants of this process: *background data set clustered by COG category* and *user defined protein sub list clustered by COG category*. Background data set is all the proteins assigned with COG Categories from the same species. This data set has been preinstalled into OntoCOG server as a copy of the COG database.

User defined protein list clustered by COG category is an *information content entity*, and it *is about* the material entity's aggregate: *user defined protein group*. In the following use case, an example of user defined protein group includes all the proteins annotated by one COG category.

A Perl library

Text::NSP::Measures::2D::Fisher::twotailed

that runs COG enrichment analysis by Fisher's exact test has been asserted as a subclass of *algorithm (IAO_0000064)*.

The specified output of *COG enrichment data transformation using Fisher's exact test* is *COG enrichment Fisher's test p-value*.

Modeling OntoCOG output data in CAO. CAO captures the minimum information for a COG enrichment analysis. CAO specifies relevant COG categories of proteins and a p-value that explains the significance of the distribution of the list of input proteins compared to that of the whole protein list in the same organism.

In the bioinformatics field, "proteins" are often treated as data, or the system ignores the reality of a protein as a material entity. However, in ontology, "protein as material entity" and "protein as data" are distinct from each other. Recognizing this distinction will avoid the vagueness and inconsistency found in many Semantic Web applications. While a protein molecule is a material entity, a protein list is a type of datum. In CAO, several terms such as *user defined COG annotated protein list* are generated to represent data set instead of material entity.

In CAO, *protein* and *protein group* represent the major subtypes of *material entity* in the ontology. If a *protein* has been assigned by a COG functional category, it will be classified as a COG *functional category clustered protein*. For example, a *COG category E clustered protein, user defined* infers that this protein has been assigned for COG category E: *amino acid transport and metabolism*, and is a member of *user-defined COG category E clustered protein group* (Fig. 2). The size of this group is represented as a datatype property of the group. Both *user-defined COG category E clustered protein list* (a subclass of *clustered dataset*) and *COG enrichment Fishers test p-value* are information entity about this group of protein.

CAO includes several specific objective properties. Three CAO-specific relations have been created. The term *denoted_by* describes a relation between an independent entity and a data item. The domain of this property is *information content entity*, and the range is *independent entity*. Its range and domain are opposite to those of *is about*. However, *denoted_by* is not the inverse property of *is about* because there exists a many-to-many relation between an entity and its associated information. Examples of the usage of this new relation in CAO includes: a *COG category protein* is *denoted by* some *COG functional category*, and a *protein* is *denoted by* some *COG functional category*.

The OntoCOG relations *has_member* and *is_member_of* are a pair of inverse properties. They describe the relations of a collective of entities (*object_aggregate*) and the entities within this collectivity. Both collective and individual entities are independent continuants. Both *has_member* and *is_member_of* are relations at the instance level, meaning that all the entities within one collective must be one kind. For example, an instance of the *COG category E clustered protein group* has and only has members from all the instances of class *COG category E clustered protein, user defined*.

3.2 Validation of CAO

CAO was validated by inputting real data as instances in CAO using Protege 4. The OWL reasoner HemiT1.3.3 (http://hermit-reasoner.com/) was used to check the consistency and axioms defined in CAO.

Two types of data were used for the validation of CAO: 1) a list of protein identifiers followed by COG functional category annotations; 2) a list of COG category clustered protein groups followed by pre-calculated COG enrichment p-value.

Data consistency checking:
In CAO, a protein assigned by a COG functional category is represented as following: *protein17987454 denoted_by E*, where *E* is an instance of COG *amino acid transport and metabolism* (*i.e.*, COG category E).

Reasoning experiments were performed to classify individual proteins into different classes: *COG category protein* and its subclasses. The term *protein17987454* will be inferred as an instance of *COG category E clustered protein, user defined*.

Axiom validation of CAO:
Three axioms have been validated in CAO by using a reasoner to perform the classification of input data (Fig. 4):

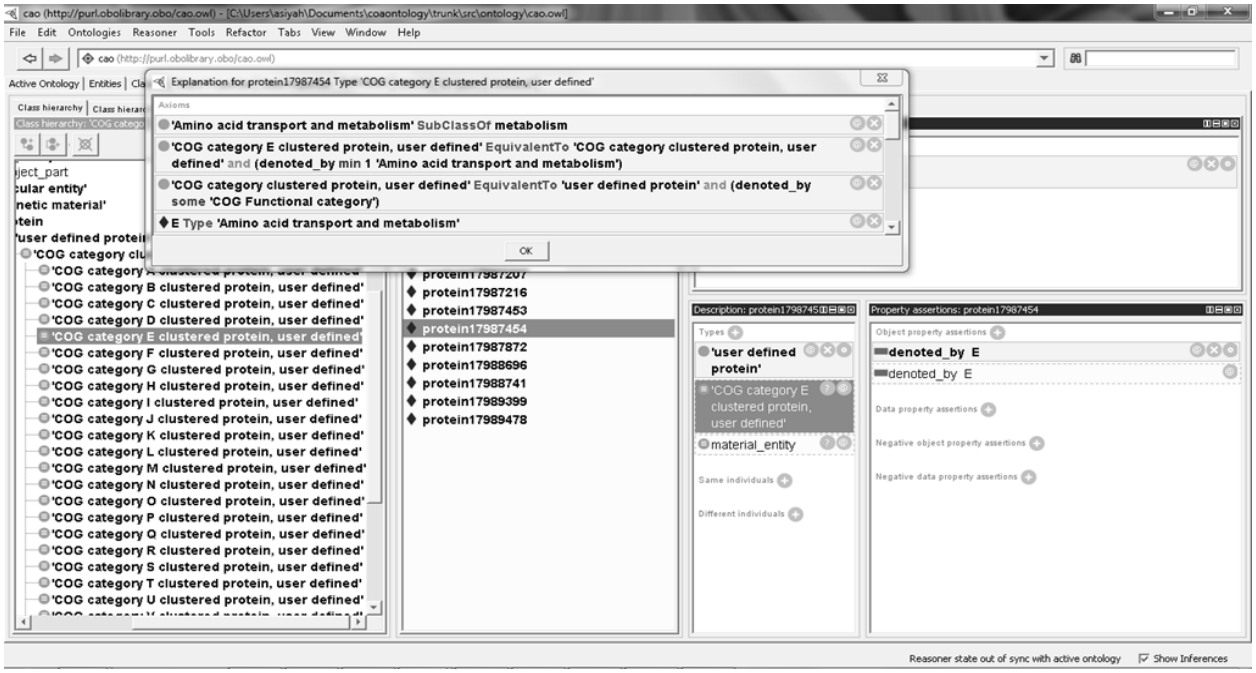

Figure 4. Automatic classification by reasoning.
A *protein17987454* has been annotated as *E*, a instance of *COG amino acid transport and metabolism* (COG category E)).
By Axiom 2, this protein is classified as an instance of *COG category E clustered protein, user defined*.

Axiom 1: a *COG category clustered protein, userdefined* is a protein that has been annotated by a COG functional category in COG database:

COG category clustered protein, user defined ≡ *user defined protein* and *(denoted_by* some *COG Functional category)*

Axiom 2: a *COG category E clustered protein, user defined* is a protein from the given list that has at least 1 annotation of *COG Amino acid transport and metabolism*

COG category E clustered protein, user defined ≡ *COG category protein* and *(denoted_by min 1 COG Amino acid transport and metabolism)*

Axiom 3: a *COG category E clustered protein group, user defined* is a group of proteins that includes only the instances of *COG category E clustered protein, user defined*

COG category E clustered protein group, user defined ≡ *protein group* and *(has_member* only *COG category E protein)*

Our studies found that all axioms and constraints in CAO are effective and efficient for data consistency checking.

3.3 Testing OntoCOG with *Brucella* Protein Virulence Factors

A list of 209 protein virulence factors from *Brucella melitensis* was obtained from the Brucellosis Ontology (BO) [10], and was submitted to OntoCOG via web interface. The COG enrichment analysis result returned by OntoCOG is shown in Table 1.

COG Category	*Proteins*	*Fisher's exact test p-value*
S: Function unknown	3	7.706e-07*
F: Nucleotide transport and metabolism	14	0.005*
R: General function prediction only	14	0.006*
N: Cell motility	8	0.011*
J: Translation	5	0.012*
G: Carbohydrate transport and metabolism	24	0.028*
U: Intracellular trafficking and secretion	9	0.052
I: Lipid transport and metabolism	3	0.083
T: Signal transduction mechanisms	11	0.111
Q: Secondary metabolites biosynthesis, transport and catabolism	1	0.126
K: Transcription	20	0.162
L: Replication, recombination and repair	6	0.237
H: Coenzyme transport and metabolism	7	0.323
O: Posttranslational modification, protein turnover, chaperones	14	0.325
C: Energy production and conversion	11	0.395
P: Inorganic ion transport and metabolism	10	0.456
E: Amino acid transport and metabolism	32	0.463
V: Defense mechanisms	3	1.000
M: Cell wall/membrane biogenesis	14	1.000
* Statistically significant (p<0.05)		

Table 1. The COG enrichment analysis of 209 *B. melitensis* virulence factors

The OntoCOG analysis identified six COG categories significantly enriched (p-value < 0.05). In total, 38 *B. melitensis* virulence factors were found to play an important role in transport and metabolism of various metabolites, including nucleotides, carbohydrates, lipids, and amino acids. Many virulence factors are components of cell motility, intracellular trafficking and secretion. These results are consistent with previous reports [11], and the p-value reports provide new statistical support. The output data can be downloaded as an RDF/OWL file that uses the CAO ontology as import ontology. The following is one synapse from the output file:

```
    <!--
http://purl.obolibrary.org/obo/CAO_cog_group_J -->
      <owl:NamedIndividual
rdf:about="&obo;CAO_cog_group_J">
        <rdf:type
rdf:resource="&obo;CAO_0000309"/>
        <rdfs:label>J clustered protein group, user
defined</rdfs:label>
        <obo:CAO_0000051
rdf:datatype="&xsd;int">5</obo:CAO_0000051>
        <obo:CAO_0000117
rdf:resource="&obo;CAO_fisher_test_p_value_J"/>
        <obo:CAO_0000052
rdf:resource="&obo;CAO_protein_17986440"/>
        <obo:CAO_0000052
rdf:resource="&obo;CAO_protein_17986559"/>
        <obo:CAO_0000052
rdf:resource="&obo;CAO_protein_17986763"/>
        <obo:CAO_0000052
rdf:resource="&obo;CAO_protein_17986899"/>
        <obo:CAO_0000052
rdf:resource="&obo;CAO_protein_17988198"/>
      </owl:NamedIndividual>
      <!--
http://purl.obolibrary.org/obo/CAO_fisher_test_p_va
lue_J -->
      <owl:NamedIndividual
rdf:about="&obo;CAO_fisher_test_p_value_J">
        <rdf:type
rdf:resource="&obo;CAO_0000040"/>

<rdfs:label>0.0115540005944313</rdfs:label>
      </owl:NamedIndividual>
```

4 Discussion

Both GO [12] and COG provide gene function annotation and classification. However, only a few of prokaryotic and eukaryotic species, such as Schizosaccharomyces *pombe* (fission yeast), *Saccharomyces cerevisiae* (baker's yeast) and *E. coli,* have both COG and GO annotations. In *Brucella*, only one gene BMEI0467 in *B. melitensis* has been annotated in GO. On the contrary, COG includes annotation of all genes in *Brucella*

melitensis and many other bacteria. Many existing web services (*e.g.*, DAVID and GOEAST) can be used for GO enrichment analysis. However, no web service for COG enrichment analysis exists yet. OntoCOG is the first web application for COG enrichment analysis.

Furthermore, OntoCOG is developed as an ontology-based Semantic Web application. OntoCOG provides CAO-based RDF/XML output data, which is more expressive and more flexible in terms of data integration. For example, users can export the p-value and the list of categories according to the enrichment measurement as other web service did. The users can also explore the attributes of specific members of each category from the given list. The use of the RDF/XML data format also allows flexibility in visualization of the data.

Future work on CAO and OntoCOG includes: 1) CAO and web interface development to allow multiple types of data input, data query, and result retrieval. 2) Provide additional statistics calculations other than Fisher's exact test. 3) Development of more advanced visualization features.

Acknowledgments

This project is supported by NIH grant 1R01AI081062. We gratefully acknowledge the critical review and editing of this manuscript by Dr. Barry Smith at the State University of New York at Buffalo.

References

1. Bodenreider O: Biomedical ontologies in action: role in knowledge management, data integration and decision support. *Yearb Med Inform* 2008:67-79.
2. Fitch WM: Distinguishing homologous from analogous proteins. *Syst Zool* 1970, 19(2):99-113.
3. Tatusov RL, Koonin EV, Lipman DJ: A genomic perspective on protein families. *Science* 1997, 278(5338):631-637.
4. Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, Koonin EV, Krylov DM, Mazumder R, Mekhedov SL, Nikolskaya AN *et al*: The COG database: an updated version includes eukaryotes. *BMC Bioinformatics* 2003, 4:41.
5. Kaufmann M: The role of the COG database in comparative and functional genomics. *Curr Bioinform* 2006, 1(3):291-300.
6. Xiang Z, Courtot M, Brinkman RR, Ruttenberg A, He Y: OntoFox: web-based support for ontology reuse. *BMC Res Notes* 2010, 3:175.
7. Brinkman RR, Courtot M, Derom D, Fostel JM, He Y, Lord P, Malone J, Parkinson H, Peters B, Rocca-Serra P *et al*: Modeling biomedical experimental processes with OBI. *J Biomed Semantics* 2010, 1 Suppl 1:S7.
8. IAO ontology: http://code.google.com/p/information-artifact-ontology/
9. Sayers EW, Barrett T, Benson DA, Bryant SH, Canese K, Chetvernin V, Church DM, DiCuccio M, Edgar R, Federhen S *et al*: Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res* 2009, 37(Database issue):D5-15.
10. Brucellosis Ontology (BO) http://sourceforge.net/projects/bo-ontology.
11. Xiang Z, Zheng W, He Y: BBP: Brucella genome annotation with literature mining and curation. *BMC Bioinformatics* 2006, 7:347.
12. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT *et al*: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* 2000, 25(1):25-29.