

Towards a Semantic Web application: Ontology-driven ortholog clustering analysis

Yu Lin, Zuoshuang Xiang,
Yongqun He

Outline

- ▶ Background of COG (Clusters of Orthologous Groups) database
- ▶ COG-based gene set enrichment analysis
- ▶ COG Analysis Ontology (CAO)
- ▶ OntoCOG, the semantic web application for COG enrichment analysis

Ortholog & COG database

- ▶ ortholog : Orthologs are genes in different species that have evolved from a common ancestral gene by speciation. Orthologs usually share the same functions in the course of evolution.
- ▶ COG database:
 - 1) collections of orthologs
 - 2) clusters orthologs to functional groups.
- ▶ Entry in COG has COG ID, or may have a functional category assignment.

COGs
Phylogenetic classification of proteins encoded in complete genomes

NCBI

Clusters of Orthologous Groups of proteins (COGs) were delineated by comparing protein sequences encoded in complete genomes, representing major phylogenetic lineages. Each COG consists of individual proteins or groups of paralogs from at least 3 lineages and thus corresponds to an ancient conserved domain.

Unicellular clusters		FTP		Initial
66 genomes				
38 orders				
28 classes				
14 phyla				
				version

Eukaryotic Clusters		FTP
Code	Name	Abbreviation
A	<i>Arabidopsis thaliana</i> (thale cress)	ath
C	<i>Caenorhabditis elegans</i> (worm)	cel
D	<i>Drosophila melanogaster</i> (fruit fly)	dme
H	<i>Homo sapiens</i> (human)	hsa
Y	<i>Saccharomyces cerevisiae</i> (baker yeast)	sce
P	<i>Schizosaccharomyces pombe</i> (fission yeast)	spo
E	<i>Encephalitozoon cuniculi</i> (Microsporidia)	ecu

COG vs. GO

- ▶ Same: Classified categories with gene product assigned, provide gene function annotation and classification.
- ▶ Different:
 - Categories
 - Species:
GO: model animals; COG: 66 genomes.
(COG covers more bacteria.)
 - Only *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 both in GO and COG.
GO:0042803 : protein homodimerization activity
COG0408: Coproporphyrinogen III oxidase (Coenzyme transport and metabolism H)

COG enrichment analysis

Contingency table

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

Given a list of k COG annotated proteins with a total of t proteins, for a given COG category A, there are q proteins within k and m proteins within t associated with it.

Fisher's Exact Test

$$P(X = x > q) = \sum_{x=q}^m \frac{\binom{m}{x} \binom{t-m}{k-x}}{\binom{t}{k}}$$

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 lot of GO enrichment analysis services are available, but not the COG enrichment analysis service

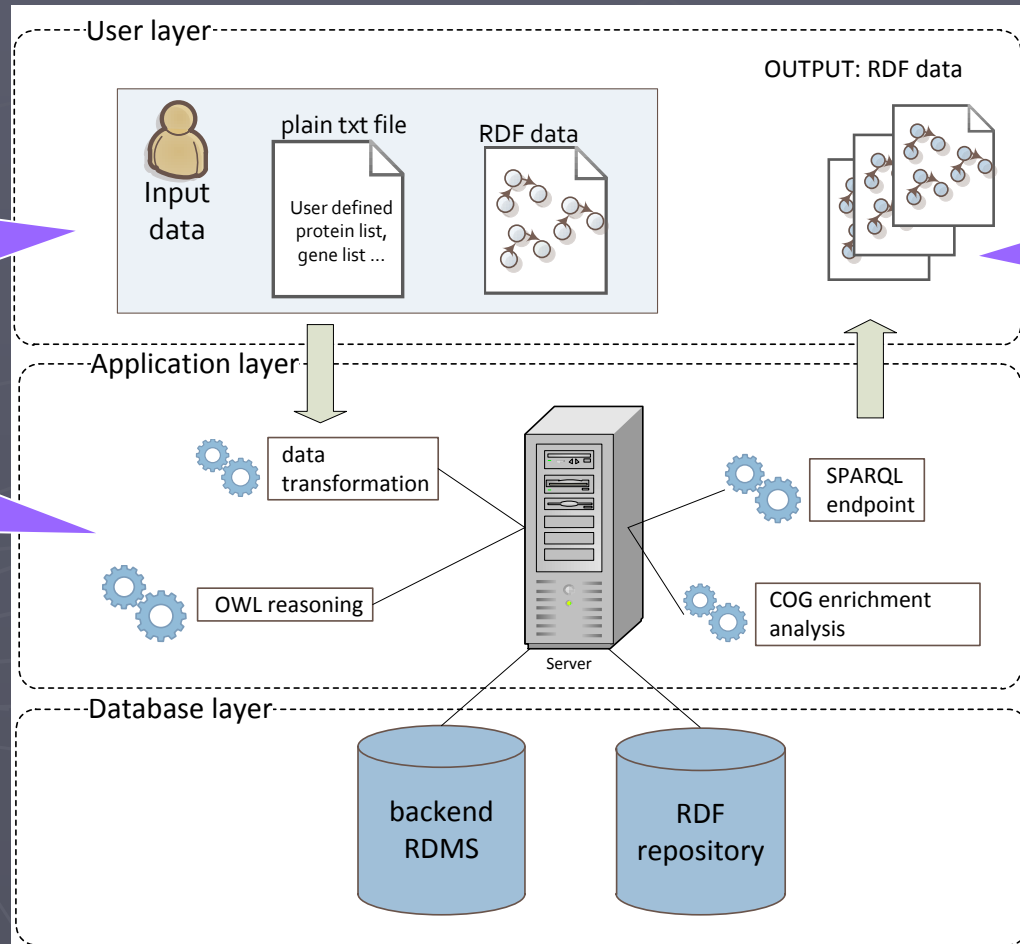
Design of OntoCOG

OntoCOG : a Semantic Web service application for COG enrichment analysis.

Input data: a list of protein defined by user for COG enrichment analysis

CAO (COG Analysis Ontology) supported

Output data: proteins grouped by COG category with a p-value in OWL format.



COG Analysis Ontology (CAO)

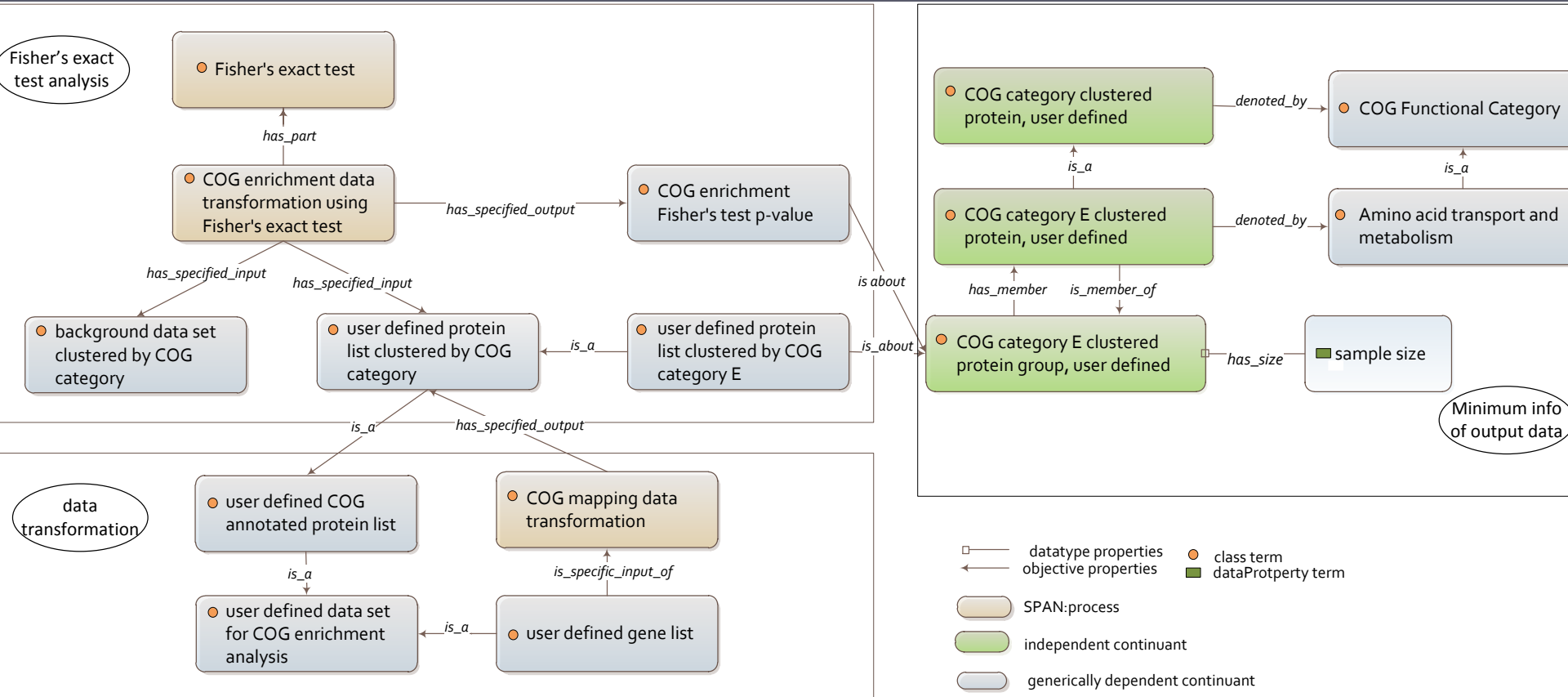
- ▶ Scope

- 1) ontology-based software/service design;
- 2) supporting data integration and exchange in OWL format.

- ▶ Domain

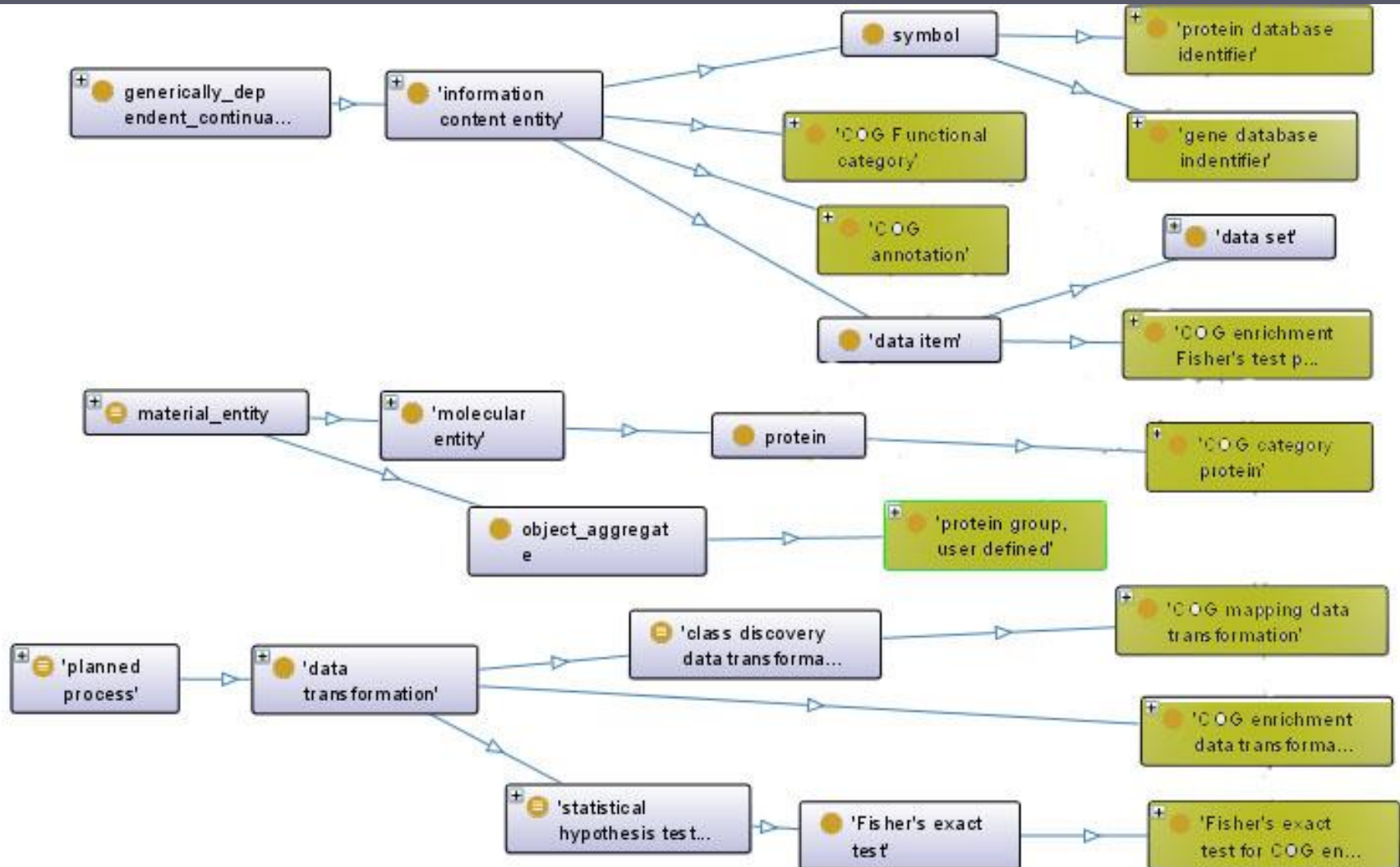
statistical analysis
protein's COG annotation

Design of CAO



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 yellow, light purple, and green boxes denote *processes*, *generically dependent continuants*, and *independent continuants*, respectively.

COG Analysis Ontology (CAO) : Core Terms



Information captured by CAO

- ▶ The given list
- ▶ The proteins grouped by COG categories
- ▶ The size of each category in the given list *
- ▶ The p-value of each category in the given list *

It captures more information than traditional COG enrichment analysis (non-SW technology supported)

```
Text output
H: 1.49713616709924e-07; 35*
E: 7.45290117727196e-11; 72*
P: 0.0430093063719378; 7*
J: 6.95603641934163e-13; 52*
F: 1.32399203293874e-05; 21*
I: 0.00262575594696207; 1*
G: 1.00000000000047; 16
C: 0.221275273142313; 11
D: 1; 2
R: 5.22697546843457e-07; 7*
S: 8.81150233068625e-10; 1*
O: 0.0082668319976629; 3*
L: 0.0082668319976629; 3*
K: 3.25387512255916e-05; 2*
B: 0.162656083050577; 1
Q: 0.231179805591791; 2
```

The traditional output of COG enrichment analysis.

*Format:
Category: p-value; size;
(* denotes p-value < 0.05 significant)*

New relations in CAO

▶ denoted_by

- describes a relation of an independent entity and a data item
an independent entity "denoted_by" a data item
- Not a reverse relation of "denotes"

▶ is_member_of has_member

- Reverse relations
- Describes relation of object and object aggregate

Axioms in CAO

- ▶ *COG category clustered protein, user defined \equiv user defined protein and (denoted_by some COG Functional category)*
- ▶ *COG category E clustered protein, user defined \equiv COG category protein and (denoted_by min 1 COG Amino acid transport and metabolism)*
- ▶ *COG category E clustered protein group, user defined \equiv protein group and (has_member only COG category E protein)*

Validation of CAO

The screenshot displays the Protege ontology editor interface. The main window shows the 'cao' ontology with a list of classes on the left, including 'COG category A clustered protein, user defined' through 'COG category U clustered protein, user defined'. The 'COG category E clustered protein, user defined' class is selected and highlighted in blue. A central dialog box titled 'Explanation for protein17987454 Type 'COG category E clustered protein, user defined'' is open, listing several axioms:

- 'Amino acid transport and metabolism' **SubClassOf** metabolism
- 'COG category E clustered protein, user defined' **EquivalentTo** 'COG category clustered protein, user defined' **and** (denoted_by **min** 1 'Amino acid transport and metabolism')
- 'COG category clustered protein, user defined' **EquivalentTo** 'user defined protein' **and** (denoted_by **some** 'COG Functional category')
- ◆ **E Type** 'Amino acid transport and metabolism'

Below the dialog box, a list of protein instances is shown, with 'protein17987454' selected and highlighted in blue. To the right, the 'Description: protein17987454' panel shows the type hierarchy: 'user defined protein' (parent), 'COG category E clustered protein, user defined' (child), and 'material_entity' (parent). The 'Property assertions: protein17987454' panel shows object property assertions, including 'denoted_by E' and 'denoted_by E'.

At the bottom right, the status bar indicates 'Reasoner state out of sync with active ontology' and 'Show Inferences' is checked.

Summaries on CAO

- ▶ An ontology to represent COG enrichment analysis
- ▶ An ontology to represent the COG enrichment analysis service : OntoCOG
- ▶ It is a use case of IAO (Information Artifact Ontology) and OBI (Ontology for Biomedical Investigation)
- ▶ It supports OntoCOG.

OntoCOG

<http://ontobat.hegroup.org/ontocog/index.php>

Ontobat

Home SPARQL Query OntoConvert **OntoCOG** Introduction Tutorial

OntoCOG

(1) Select one species
Brucella melitensis

(2) Input protein GIs: [Example 1](#), [Example 2 \(Protein Virulence factor\)](#)

17986368
17987110
17986900
17986900
17987216
17988169
17987771
17987399

Get OWL (RDF/XML) Output File Reset

FAQs References Links Contact Acknowledge

OntoCOG analysis of *Brucella* virulence factors

Ontobat

[Home](#)[SPARQL Query](#)[OntoConvert](#)[OntoCOG](#)[Introduction](#)[Tutorial](#)

OntoCOG Query Result

[OWL\(RDF/XML\) output file](#) generated.

Text output

E: 0.463287665317927; 32
I: 0.0829671299066097; 3
H: 0.323329281729191; 7
G: 0.0282030340169236; 24*
F: 0.00526801376681896; 14*
L: 0.236978256542513; 6
P: 0.455906297607988; 10
J: 0.0115540005944313; 5*
K: 0.162325010535853; 20
R: 0.00634131762994135; 14*
C: 0.394976297636098; 11
O: 0.324698940577407; 14
T: 0.111098524579059; 11
M: 1.000000000000005; 14
N: 0.0113452455156726; 8*
U: 0.0521202164142756; 9
V: 0.999999999999995; 3
Q: 0.12568302563995; 1
S: 7.7059734062508e-07; 3*

```
<owl:NamedIndividual rdf:about="&obo;CAO_cog_group_N">
  <rdf:type rdf:resource="&obo;CAO_0000313"/>
  <rdfs:label>N clustered protein group, user defined</rdfs:label>
  <obo:CAO_0000051 rdf:datatype="&xsd:int">8</obo:CAO_0000051>
  <obo:CAO_0000052 rdf:resource="&obo;CAO_protein_17988379"/>
  <obo:CAO_0000052 rdf:resource="&obo;CAO_protein_17988510"/>
  <obo:CAO_0000052 rdf:resource="&obo;CAO_protein_17988494"/>
  <obo:CAO_0000052 rdf:resource="&obo;CAO_protein_17988498"/>
  <obo:CAO_0000052 rdf:resource="&obo;CAO_protein_17989429"/>
  <obo:CAO_0000052 rdf:resource="&obo;CAO_protein_17988503"/>
  <obo:CAO_0000052 rdf:resource="&obo;CAO_protein_17988495"/>
  <obo:CAO_0000052 rdf:resource="&obo;CAO_protein_17986369"/>
  <obo:CAO_0000117 rdf:resource="&obo;CAO_fisher_test_p_value_N"/>
</owl:NamedIndividual>
```

```
<owl:NamedIndividual rdf:about="&obo;CAO_fisher_test_p_value_N">
  <rdf:type rdf:resource="&obo;CAO_0000040"/>
  <rdfs:label>0.0113452455156726</rdfs:label>
</owl:NamedIndividual>
```

[Download CAO ontology.](#)

[FAQs](#)[References](#)[Links](#)[Contact](#)[Acknowledge](#)

Result

The screenshot shows the Protege OWL editor interface. The main window displays the class hierarchy on the left, the members list in the center, and the property assertions on the right. The class hierarchy shows a tree structure starting with 'object' and 'object_aggregate', leading to 'protein group, user defined', which contains various 'COG category' classes. The 'Members list' shows the selected class and its members. The 'Property assertions' panel shows several 'has_member' assertions with protein GI values and a 'denoted_by' assertion with a numerical value. A 'has size' data property assertion is also visible.

caio_output (http://purl.obolibrary.org/obo/caio_output.owl) - [C:\Users\asiyah\Documents\ontologies\test2_OntoCOG.owl]

File Edit View Reasoner Tools Refactor Window Help

caio_output (http://purl.obolibrary.org/obo/caio_output.owl)

Active Ontology | Entities | Classes | Object Properties | Data Properties | Individuals | OWLviz | DL Query | OntoGraf

Class hierarchy | Class hierarchy (inferred)

Class hierarchy: 'COG category N clustered protein group, user defined'

- object
- object_aggregate
 - 'protein group, user defined'
 - 'COG category A clustered protein group, user defined'
 - 'COG category B clustered protein group, user defined'
 - 'COG category C clustered protein group, user defined'
 - 'COG category D clustered protein group, user defined'
 - 'COG category E clustered protein group, user defined'
 - 'COG category F clustered protein group, user defined'
 - 'COG category G clustered protein group, user defined'
 - 'COG category H clustered protein group, user defined'
 - 'COG category I clustered protein group, user defined'
 - 'COG category J clustered protein group, user defined'
 - 'COG category K clustered protein group, user defined'
 - 'COG category L clustered protein group, user defined'
 - 'COG category M clustered protein group, user defined'
 - 'COG category N clustered protein group, user defined'
 - 'COG category O clustered protein group, user defined'
 - 'COG category P clustered protein group, user defined'
 - 'COG category Q clustered protein group, user defined'
 - 'COG category R clustered protein group, user defined'
 - 'COG category S clustered protein group, user defined'
 - 'COG category T clustered protein group, user defined'
 - 'COG category U clustered protein group, user defined'
 - 'COG category V clustered protein group, user defined'
 - 'COG category W clustered protein group, user defined'

Members list | Members list (inferred)

Members list: 'N clustered protein group, user defined'

'N clustered protein group, user defined'

Annotations | Usage

Annotations: 'N clustered protein group, user defined'

Annotations +

Description: 'N clustered protein group, user defined'

Types +

- 'COG category N clustered protein group, user defined'

Same individuals +

Different individuals +

Property assertions: 'N clustered protein group, user defined'

Object property assertions +

- has_member protein_gi_17988498
- has_member protein_gi_17986369
- has_member protein_gi_17988494
- denoted_by 0.0113452455156726
- has_member protein_gi_17989429
- has_member protein_gi_17988379
- has_member protein_gi_17988503
- has_member protein_gi_17988510
- has_member protein_gi_17988495

Data property assertions +

- 'has size' "8"^^int

Negative object property assertions +

Negative data property assertions +

To use the reasoner click Reasoner -> Start Reasoner Show Inferences

Final Conclusion

- ▶ OntoCOG provide a platform independent server for COG enrichment analysis
- ▶ CAO ontology supports the design and workflow of OntoCOG.
- ▶ OntoCOG is the first semantic web application used for such purpose.
- ▶ Future work: interface developing; expand to other statistical analysis; output data visualization.

Acknowledgement

- The OntoCOG project is supported by NIH grant 1R01AI081062.
- People:
 - Yu Lin
 - Yongqun “Oliver” He
 - Zuoshuang “Allen” Xiang
- Special thanks to ICBO Committee
- Thank Dr. Barry Smith for correcting the English in our manuscript.