OWL primer for GO users

David Osumi-Sutherland
What is OWL?

- **Web Ontology Language**
  - Can express everything in OBO and more.

- **Certified web standard**

- **Fast reasoning software allows:**
  - automated ontology building; error checking; queries
I’m an annotator, why should I care about OWL?

- OBO is OWL
  - OBO 1.4 spec defines OBO as a mapping to OWL.
- OWL reasoning is used by GO to:
  - Automate classification during ontology building
  - Check for errors (inconsistencies) in ontology and annotations
  - Drive TermGenie
  - **Tell you how annotation extensions fold**
I’m an annotator, why should I care about OWL?

- OBO is OWL
  - OBO 1.4 spec defines OBO as a mapping to OWL.
- OWL reasoning is used by GO to:
  - Automate classification during ontology building
  - Check for errors (inconsistencies) in ontology and annotations
  - Drive TermGenie
  - Tell you how annotation extensions fold

YOU CAN’T KNOW HOW AN ANNOTATION EXTENSION WILL FOLD WITHOUT ASKING A REASONER!
What is an ontology?

A classification

- **Classes**
  - **appendage**
    - antenna
    - leg
    - wing
    - forewing
    - hindwing

- appendage
  - wing
  - forewing
  - hindwing
OBO-OWL cheat sheet: classification

**OWL:** antenna `SubClassOf` appendage

**OBO:** antenna `is_a` appendage
Relationships record necessary conditions for class membership

Being part of a thoracic segment is a necessary condition of being in the class leg

‘leg’ **SubClassOf** part_of some thoracic segment
Class - class relationships are quantified

- Class:Class relationships are many to many
  - Does the relation apply to all or just some of the class?
    - we specify this with quantifiers:
      - *some* \( \exists \): there exists,
      - \( \forall \): for all, *all, only, every*
relationships between classes use quantifiers

- OBO (quantifiers hidden)
  - name: leg
  - relationship: part_of thoracic segment

- OWL (MS):
  - leg SubClassOf part_of some ‘thoracic segment’
Directionality and quantifiers

‘wing’ SubClassOf part_of some thoracic segment ✓

‘thoracic segment’ SubClassOf has_part some ‘wing’ ✗
Defining necessary and sufficient conditions for class membership

- **English**
  - Any appendage that is part of some thoracic segment is a thoracic appendage

- **OWL**
  - thoracic appendage
  - `EquivalentTo` ‘appendage’
  - `and part_of some` thoracic segment

- **OBO**
  - `name`: thoracic appendage
  - `intersection_of`: appendage
  - `intersection_of`: `part_of` thoracic segment
appendage + part_of some 'thoracic segment' + thoracic appendage + appendage thoracic appendage part_of some 'thoracic segment' thoracic appendage wing
How automated classification helps ontology building

• Record
  • one classification (is_a)
  • relationships (necessary conditions for class membership)

• Reason:
  • To find what other classifications are applicable

+ve: It is very hard to find all relevant classifications by hand.
-ve: Only works if relations well defined and understood
We don’t need to make a class to express a concept in OWL

- Just as in annotation extensions
Some relations entail others

\[ \text{negatively_regulates} \]

\[ \text{'negatively_regulates'} \]

\[ \text{'positively_regulates'} \]

\[ \text{regulates} \]

\[ \text{some} \text{ ‘cell division’} \]

\[ \text{negatively_regulates} \]

\[ \text{some} \text{ ‘cell division’} \]
Rules

\[ IF \ X \text{ regulates} \ Y \text{ AND} \ Y \text{ part_of} \ Z \rightarrow X \text{ regulates} \ Z \]
IF $X$ regulates $Y$
AND $Y$ part_of $Z$

$X$ regulates $Z$
Annotation extension -> OWL

<table>
<thead>
<tr>
<th>Gene Product</th>
<th>GO term (c5)</th>
<th>C16</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC1</td>
<td>apoptotic process</td>
<td>occurs in photoreceptor</td>
</tr>
</tbody>
</table>

**Query (class expression)**

'apoptotic process' **and** 'occurs in' **some** 'photoreceptor cell'

**Query results**

Super classes (1)

- 'neuron apoptotic process'

Descendant classes (2)

- 'retinal cone cell apoptotic process'
- 'retinal rod cell apoptotic process'
Folding

Query (class expression)

'apoptotic process' and 'occurs in' some 'photoreceptor cell'

Execute  Add to ontology

Query results

Super classes (1)

• 'neuron apoptotic process'

Descendant classes (2)

• 'retinal cone cell apoptotic process'
• 'retinal rod cell apoptotic process'

folding
**Unfolding**

<table>
<thead>
<tr>
<th>Gene Product</th>
<th>GO term (c5)</th>
<th>C16</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC1</td>
<td>B cell apoptotic process</td>
<td></td>
</tr>
</tbody>
</table>

**Description:** 'B cell apoptotic process'

**Equivalent To**

- 'apoptotic process' and 'occurs in' some 'B cell'

<table>
<thead>
<tr>
<th>Gene Product</th>
<th>GO term (c5)</th>
<th>C16</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC1</td>
<td>apoptotic process</td>
<td>occurs in B cell</td>
</tr>
</tbody>
</table>
## OWL translation of C16 with multiple clauses

<table>
<thead>
<tr>
<th>Gene Product</th>
<th>GO term (c5)</th>
<th>C16</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASQ2</td>
<td>sequestering of calcium ion</td>
<td>occurs_in sarcoplasmic reticulum,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>occurs_in cardiac muscle cell</td>
</tr>
</tbody>
</table>

**Note – not nested**

Sometimes reasoning will give classifications you might not expect.
OWL translation of C16 finding equivalent GO terms

<table>
<thead>
<tr>
<th>Gene Product</th>
<th>GO term (c5)</th>
<th>C16</th>
</tr>
</thead>
<tbody>
<tr>
<td>xyz10</td>
<td>apoptotic process</td>
<td>occurs_in some B cell</td>
</tr>
</tbody>
</table>

Query (class expression)

'apoptotic process' and 'occurs in' some 'B cell'

Execute  Add to ontology

Query results

Equivalent classes (1)

'B cell apoptotic process'

Super classes (1)

'lymphocyte apoptotic process'

AE equivalent to existing term
TermGenie examples

TG: regulation of cell adhesion involved in retrograde extension

is_a: GO:0010975 {is_inferred="true"} ! regulation of neuron projection development
is_a: GO:0030155 {is_inferred="true"} ! regulation of cell adhesion
The reasons for inferred classification can be complicated

<table>
<thead>
<tr>
<th>Query (class expression)</th>
</tr>
</thead>
<tbody>
<tr>
<td>'biological regulation' <strong>and</strong> regulates <strong>some</strong> 'cell adhesion involved in retrograde extension'</td>
</tr>
</tbody>
</table>

### Query results

**Super classes (2)**

- ≡ 'regulation of cell adhesion'
- ≡ 'regulation of neuron projection development'

### Explanation for 'biological regulation' and (regulates some 'cell adhesion involved in retrograde extension') SubClassOf 'regulation of neuron projection development'

1. regulates **o** 'part of' SubPropertyOf regulates
2. 'retrograde extension' SubClassOf 'neuron projection development'
3. 'cell adhesion involved in retrograde extension' SubClassOf 'part of' **some** 'retrograde extension'
4. 'regulation of neuron projection development' EquivalentTo 'biological regulation' **and** (regulates **some** 'neuron projection development')
Using Protégé to find usage of relations

- 'alpha-beta T cell proliferation' EquivalentTo 'cell proliferation' and (acts_on_population_of some 'alpha-beta T cell')
- 'B cell homeostasis' EquivalentTo 'homeostasis of number of cells' and (acts_on_population_of some 'B cell')
- 'B cell homeostatic proliferation' EquivalentTo 'cell proliferation' and ('part of' some 'homeostatic process') and (acts_on_population_of some 'B cell')
- 'B cell proliferation' EquivalentTo 'cell proliferation' and (acts_on_population_of some 'B cell')
- 'B-1 B cell homeostasis' EquivalentTo 'homeostasis of number of cells' and (acts_on_population_of some 'B-1 B cell')
Take home messages

• If you understand the relations you use then the classification should look after itself.

• Except in the most trivial cases, you can’t work out how an annotation extension will fold.

• Tooling support may be necessary to help understand the implications of annotation extensions.
Tooling support for Annotation Extensions

- A web tool to allow annotators to find how extensions will fold, check for equivalence, etc?
PROBABLY WON’T USE SLIDES FROM HERE ON
ERROR MESSAGES ARE YOUR FRIENDS!
– They tell you you’ve screwed up before you get embarrassing emails complaining that you’ve screwed up
Some classes don’t intersect

OWL `DisjointWith`  OBO: `disjoint_from`
Some classes don’t intersect

- nucleus is_a kinase activity
- nucleus subClassOf kinase activity