Machine learning with biomedical ontologies

Presented by Sarah Alghamdi, Azza Althagafi, Robert Hoehndorf, Maxat Kulmanov, Sumyyah Toonsi, Fernando Zhapa-Camacho

Learning Outcomes

- Introduce Ontologies and the description logic
- Discuss unsupervised machine learning methods that can “embed” from one structure to another
- Introduce different methods that use ontologies in machine learning models
- Introduce mOWL, a software library for machine learning with ontologies
- Incorporate mOWL in Biomedical data analysis using different approaches

Preliminaries: ontologies

- “ An ontology is an **explicit specification** of a **conceptualization**” … Gruber 1993
- “ An ontology is an explicit **formal** specification of a **shared** conceptualization” … Borst 1997
- “ An ontology is a **logical theory** designed in order to capture the **intended models** corresponding to a certain conceptualization and to **exclude the unintended** ones” … Guarino 2009

Preliminaries: ontologies

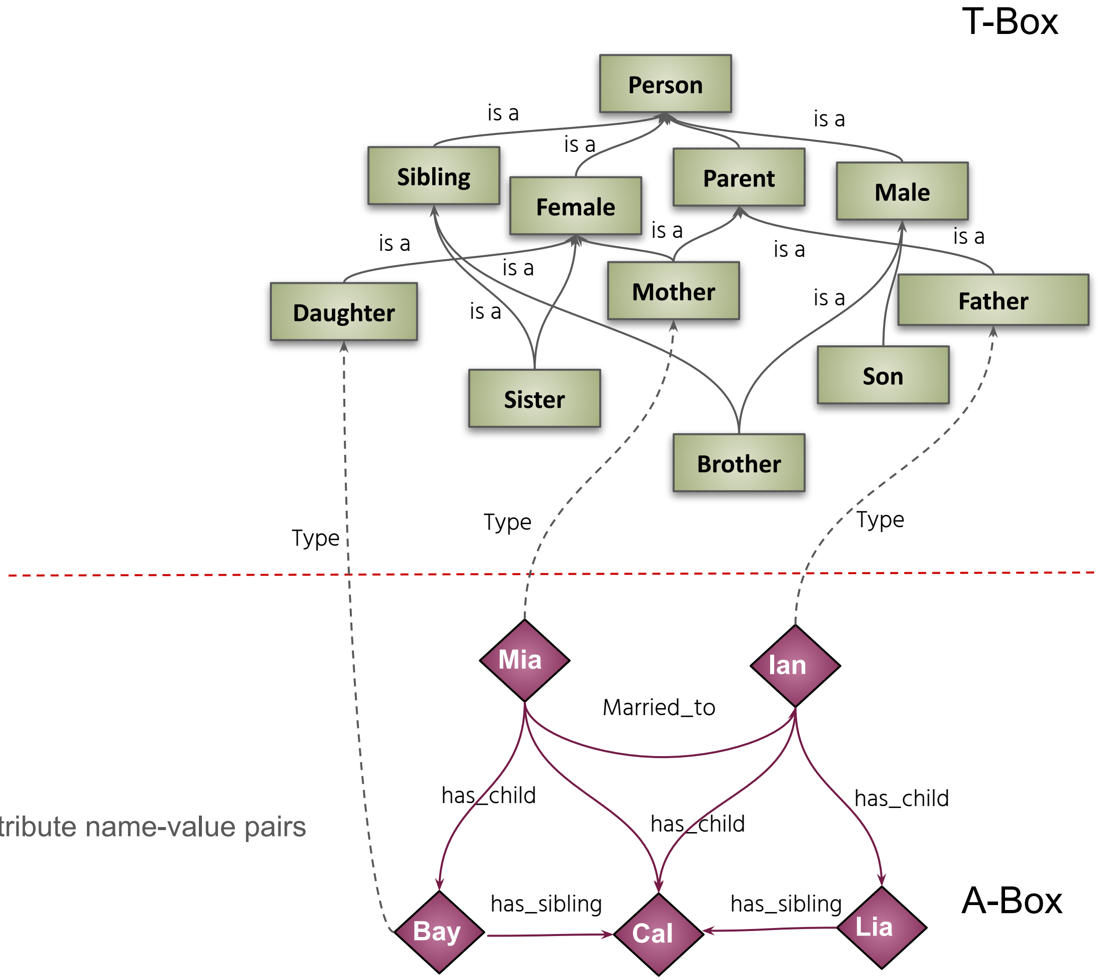
- Ontology consist of :

  O={C,R,I,⊢}

- T-Box
  - Set of terminological Component

- A-Box
  - Set of assertions using T-Box
    - Concept assertions
    - Relation Assertion

- Metadata
  - representation of a resource in terms of attribute name-value pairs
    - Definition
    - Labels
    - …



Preliminaries: ontologies

- Description Logic (DL) is used to formally and explicitly represent ontologies

Name	DL syntax	Semantics
Top concept	$\top$	$\Delta^{\mathcal{I}}$
Bottom concept	$\bot$	$\emptyset$
Concept	C	$C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Concept disjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
Concept conjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
Concept negation	$\neg C$	$\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
Universal restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}((x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$
Existential restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}((x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\}$
Subclass of	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
Subproperty of	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
Equivalent class	$C_1 \equiv C_2$	$C_1^{\mathcal{I}} = C_2^{\mathcal{I}}$
Equivalent property	$R_1 \equiv R_2$	$R_1^{\mathcal{I}} = R_2^{\mathcal{I}}$

Concepts , Roles

Person $\sqsubseteq \top$
Female $\sqcap$ *Male* $\sqsubseteq \bot$
Female $\sqcup$ *Male* $\sqsubseteq \top$
Female $\equiv \neg$ *Male*
Parent $\equiv \exists$ *has_child*. *Person*
Son $\sqsubseteq$ *Male* $\sqcap \exists$ *child_of*. *Person*
Mother $\sqsubseteq$ *Female* $\sqcap$ *Parent*
Sibling $\sqsubseteq \exists$ *has_sibling*. *Person*
has_brother $\sqsubseteq$ *has_sibling*

Preliminaries: ontologies

- Description Logic (DL) is used to formally and explicitly represent ontologies

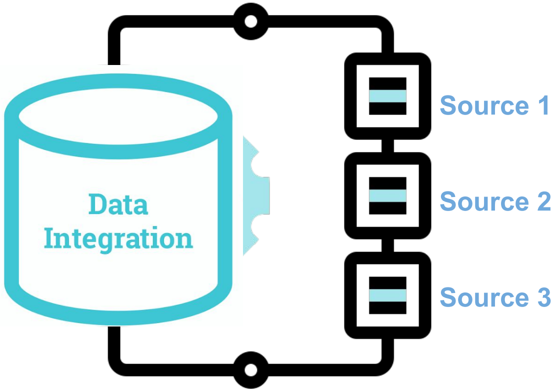
DL Syntax	Manchester Syntax
$C \sqcap D$	C and D
$C \sqcup D$	C or D
$\neg C$	not C
$\exists R.C$	R some C
$\forall R.C$	R only C
$(\geq nR.C)$	R min n C
$(\leq nR.C)$	R max n C
$(= nR.C)$	R exactly n C
$\{a\} \sqcup \{b\} \sqcup \ldots$	{a b ...}

Concepts , Roles

$Person \sqsubseteq \top$
$Female \sqcap Male \sqsubseteq \bot$
$Female \sqcup Male \sqsubseteq \top$
$Female \equiv \neg\, Male$
$Parent \equiv \exists\, has\_child.\, Person$
$Son \sqsubseteq Male \sqcap \exists\, child\_of.\, Person$
$Mother \sqsubseteq Female \sqcap Parent$
$Sibling \sqsubseteq \exists\, has\_sibling.\, Person$
$has\_brother \sqsubseteq has\_sibling$

How Ontologies are used in Databases

- Annotations and data integration
  - Ontologies play a crucial role in facilitating data integration across databases due to their usage of standard identifiers for classes and relations



How Ontologies are used in Databases

- Annotations and data integration

GAF fields

The annotation flat file format is comprised of 17 tab-delimited fields.

Column	Content	Required?	Cardinality	Example	
1	DB	required	1	UniProtKB	
2	DB Object ID	required	1	P12345	
3	DB Object Symbol	required	1	PHO3	
4	Qualifier	required	1 or 2	NOTIinvolved_in	
5	GO ID	required	1	GO:0003993	
6	DB:Reference (IDB:Reference)	required	1 or greater	PMID:2676709	
7	Evidence Code	required	1	IMP	
8	With (or) From	optional	0 or greater	GO:0000346	
9	Aspect	required	1	F	
10	DB Object Name	optional	0 or 1	Toll-like receptor 4	
11	DB Object Synonym (ISynonym)	optional	0 or greater	hToll	Tollbooth
12	DB Object Type	required	1	protein	
13	Taxon(Itaxon)	required	1 or 2	taxon:9606	
14	Date	required	1	20090118	
15	Assigned By	required	1	SGD	
16	Annotation Extension	optional	0 or greater	part_of(CL:0000576)	
17	Gene Product Form ID	optional	0 or 1	UniProtKB:P12345-2	

How Ontologies are used in Databases

- Annotations and data integration

1. UniProtKB
2. A0A024RBG1
3. NUDT4B
4. enables
5. GO:0003723
6. GO_REF:0000043
7. IEA
8. UniProtKB-KW:KW-0694
9. F
10. Diphosphoinositol polyphosphate phosphohydrolase
11. NUDT4B
12. NUDT4B
13. Protein
14. taxon:9606 20221109
15. UniProt
16.
17.

1. MGI
2. MGI:1913300
3. 0610009B22Rik
4. enables
5. GO:0001222
6. MGI:MGI:4834177|GO_REF:0000096
7. ISO
8. UniProtKB:P0DI82
9. F
10. RIKEN cDNA 0610009B22 gene
11.
12. protein_coding_gene
13. taxon:10090
14. 20210709
15. MGI
16.
17.

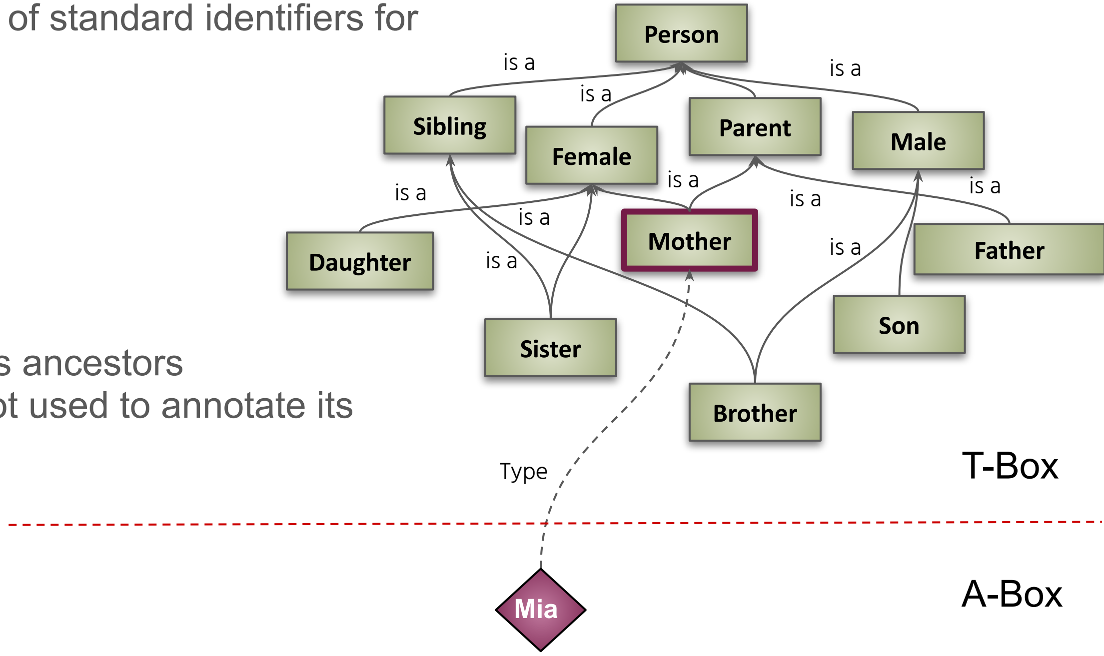
GAF fields

The annotation flat file format is comprised of 17 tab-delimited fields.

Column	Content	Required?	Cardinality	Example
1	DB	required	1	UniProtKB
2	DB Object ID	required	1	P12345
3	DB Object Symbol	required	1	PHO3
4	Qualifier	required	1 or 2	NOT|involved_in
5	GO ID	required	1	GO:0003993
6	DB:Reference (|DB:Reference)	required	1 or greater	PMID:2676709
7	Evidence Code	required	1	IMP
8	With (or) From	optional	0 or greater	GO:0000346
9	Aspect	required	1	F
10	DB Object Name	optional	0 or 1	Toll-like receptor 4
11	DB Object Synonym (|Synonym)	optional	0 or greater	hToll Tollbooth
12	DB Object Type	required	1	protein
13	Taxon(|taxon)	required	1 or 2	taxon:9606
14	Date	required	1	20090118
15	Assigned By	required	1	SGD
16	Annotation Extension	optional	0 or greater	part_of(CL:0000576)
17	Gene Product Form ID	optional	0 or 1	UniProtKB:P12345-2

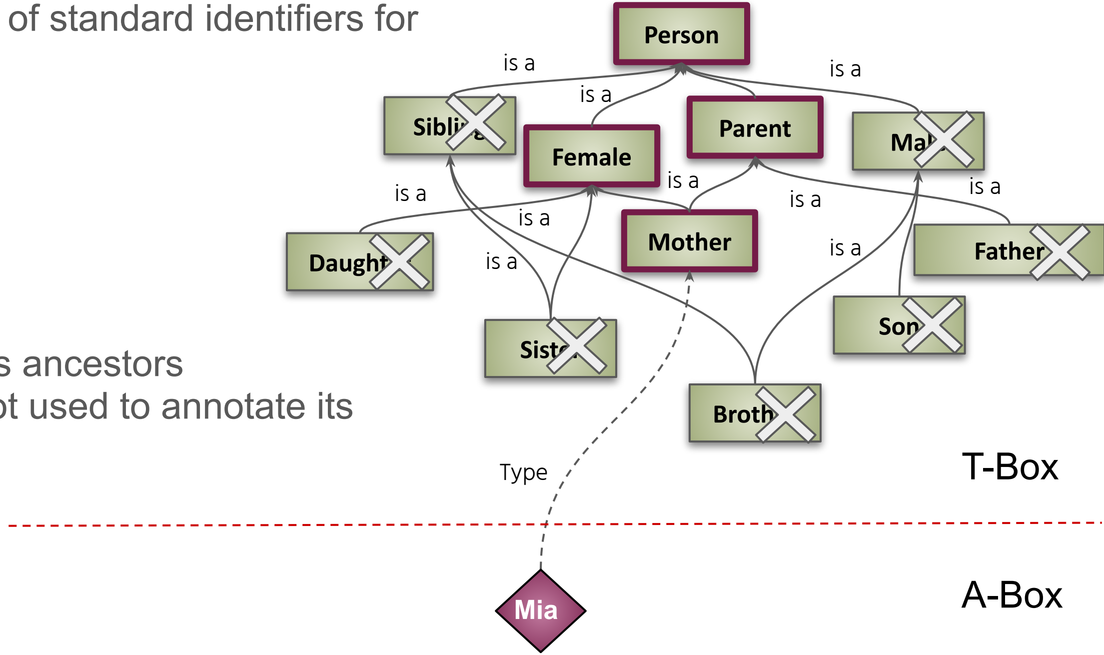
How Ontologies are used in Databases

- Annotations and data integration
  - Ontologies play a crucial role in facilitating data integration across databases due to their usage of standard identifiers for classes and relations
- True path rule:
  - Annotation for a class is passed to its ancestors
  - Unannotated entities for a class is not used to annotate its descendants



How Ontologies are used in Databases

- Annotations and data integration
  - Ontologies play a crucial role in facilitating data integration across databases due to their usage of standard identifiers for classes and relations
- True path rule:
  - Annotation for a class is passed to its ancestors
  - Unannotated entities for a class is not used to annotate its descendants

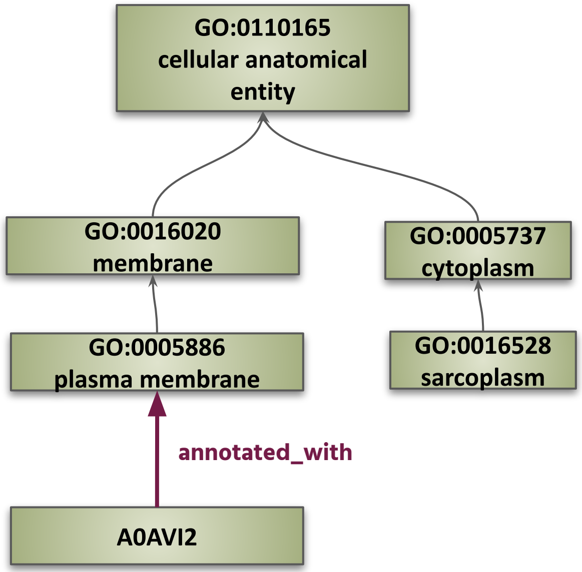


Transforming GO Annotations to OWL axioms

Example:

Annotating protein **A0AVI2** To Gene Ontology

- Annotations to T-Box
  - $\textbf{\textit{A0AVI2}} \sqsubseteq \exists\ annotated\_with.\ \textbf{\textit{GO:0005886}}$



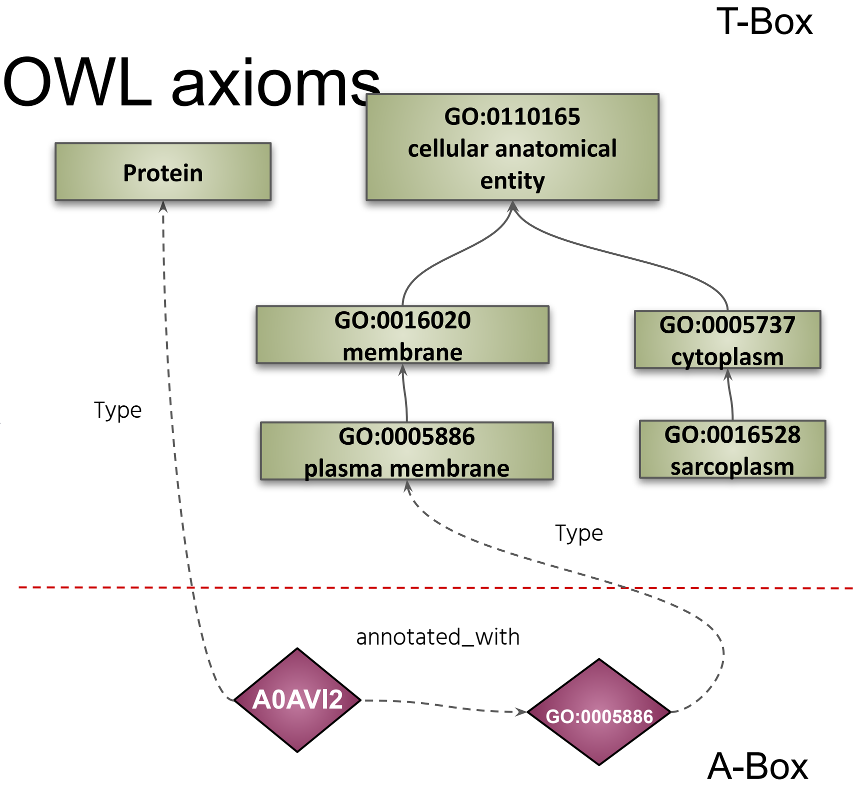
Transforming GO Annotations to OWL axioms

Example:

Annotating protein **A0AVI2** To Gene Ontology

- Annotations to T-Box
  - ***A0AVI2*** $\sqsubseteq \exists$ *annotated_with.* ***GO:0005886***

- Annotations to A-Box
  - Protein(***A0AVI2***)
  - annotated_with(***A0AVI2, GO:0005886***)

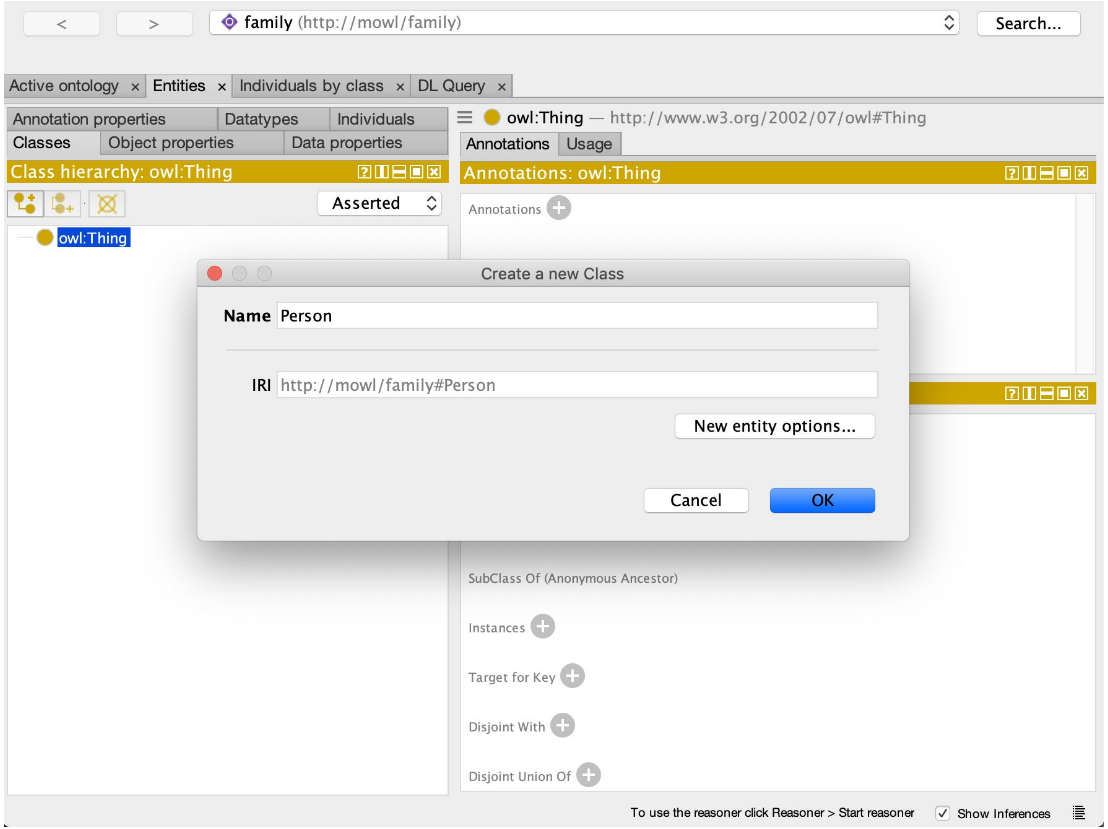


Hands on (1):

- Creating the family ontology
  - WebProtégé
    - https://webprotege.stanford.edu/#login
    - Download from: https://protege.stanford.edu/

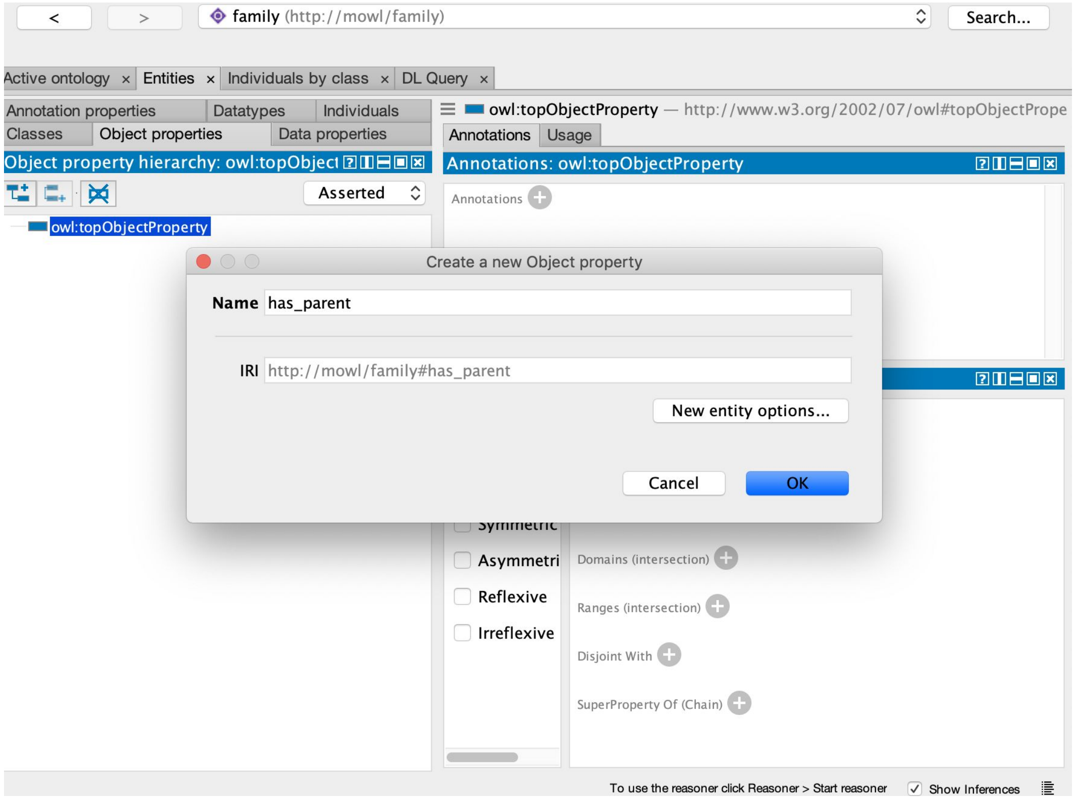
Hands on (1):

- Protégé:
  - Add new classes



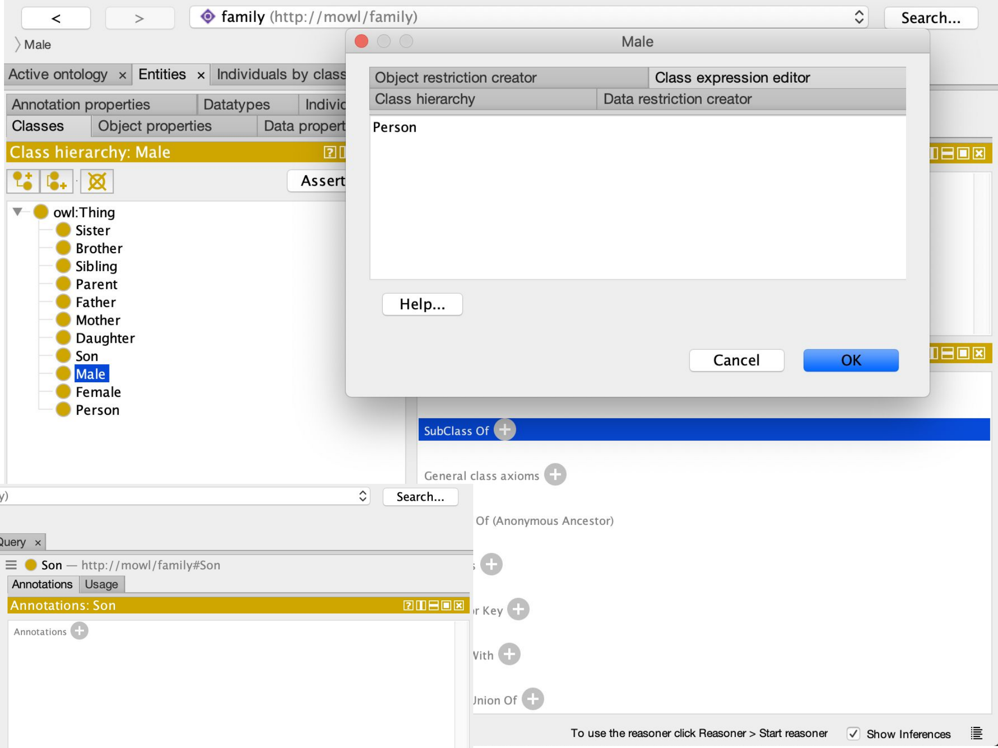
Hands on (1):

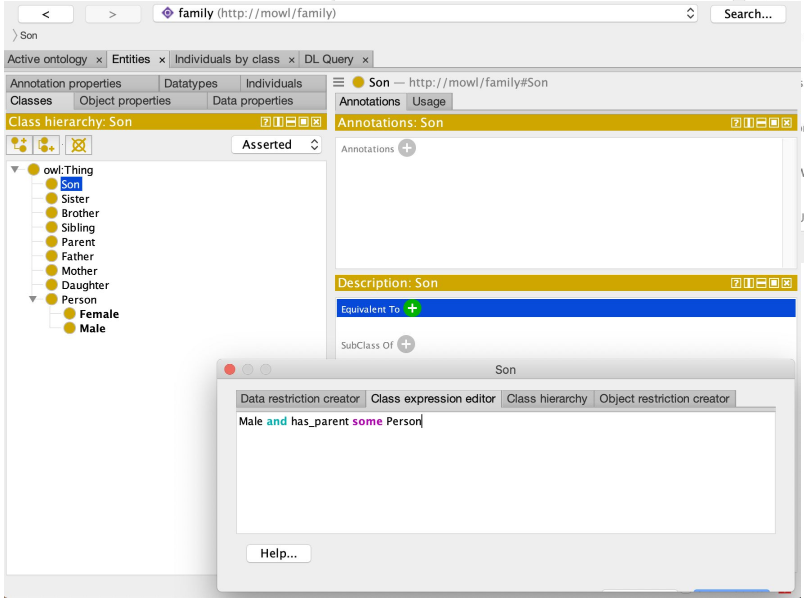
- Protégé:
    - Add new classes
    - Add new object properties

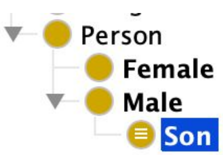


Hands on (1):

- Protégé:
  - Add new classes
  - Add new object properties
  - Adding axioms
    - SubclassOf
    - Equivalent
    - ….

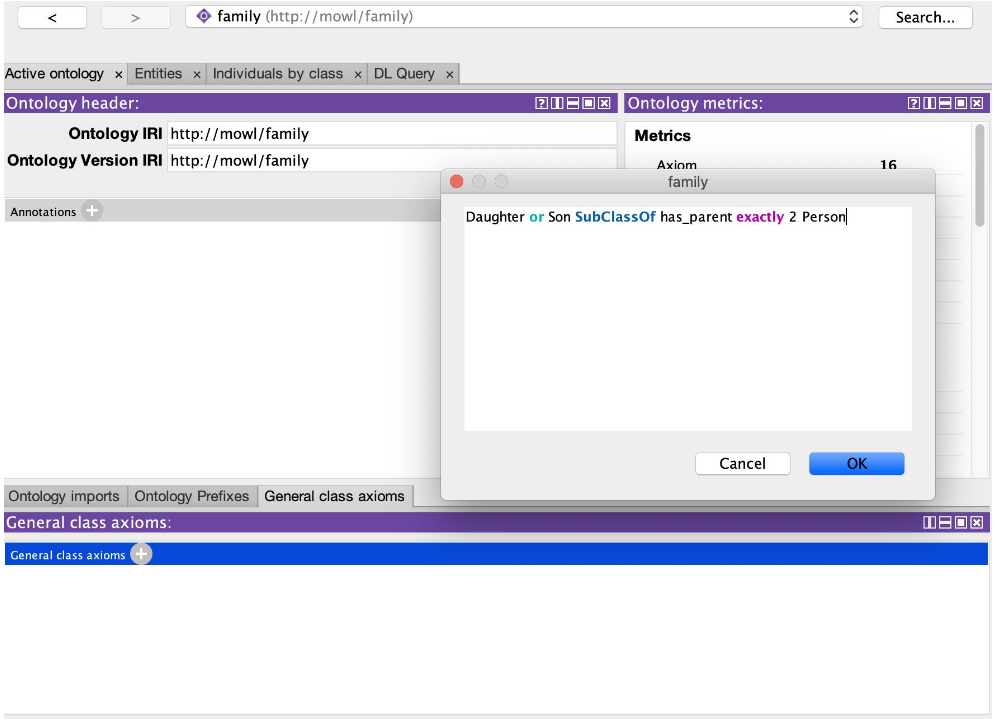






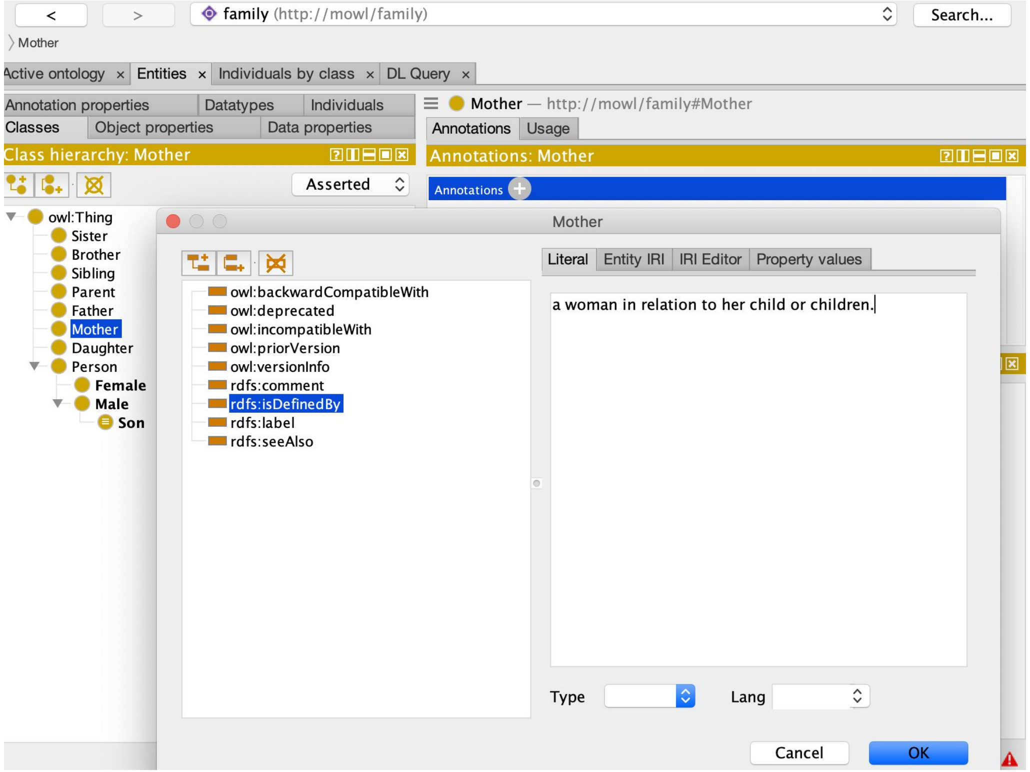
Hands on (1):

- Protégé:
    - Add new classes
    - Add new object properties
    - Adding axioms
    - Adding GCIs



Hands on (1):

- Protégé:
  - Add new classes
  - Add new object properties
  - Adding axioms
  - Adding GCIs
  - Adding definitions, synonyms



Hands on (1):

- Protégé:
  - Add new classes
  - Add new object properties
  - Adding axioms
  - Adding GCIs
  - Adding definitions, synonyms
  - Using reasoners