Ontologies and their role in the biomedical domain

Ontology features

Independent of the actual definition of what an ‘ontology’ is, most artifacts labeled ‘ontologies’, as well as some ‘vocabularies’ and ‘thesauri’, provide several main features, and these features are used in almost all their applications:

- classes and relations
- a domain vocabulary
- textual definitions and descriptions
- formal definitions and axioms

- **classes and relations**, referred to by an identifier such as an Internationalized Resource Identifier (IRI), a Uniform Resource Identifier (URI), or a database identifier string;
- **a domain vocabulary**, i.e. a list of terms associated with the ontology's classes and relations
- **textual definitions and descriptions** → additional information about what kind of things a class or relation refers to,
- **formal definitions and axioms** → computational counterpart to textual definitions and that can be accessed and exploited automatically using specialized software (i.e. automated reasoners) and axioms about a domain, i.e. statements that are considered to be true within that domain and which provide background knowledge about a domain.

Use of Ontology features

- **Classes and relations**

standard identifiers for classes and relations in ontologies enables data integration across multiple databases

- **Domain vocabulary**

labels associated with classes and relations provide a domain vocabulary that can be exploited for applications ranging from natural language processing, creation of user interfaces, etc.

Use of Ontology features

- **Metadata and descriptions**

Textual definitions, descriptions, examples and further metadata associated with classes → understand the precise meaning of class in the ontology.

The definitions and related metadata should allow consistent understanding of the meaning of classes in ontologies.

- **Axioms and formal definitions**

Formal definitions and axioms enable automated and computational access to (some parts of) the meaning of a class or relation.

Ontology principal components: Classes and relations

- A ‘class’ is an entity that refers to a set of entities in the world

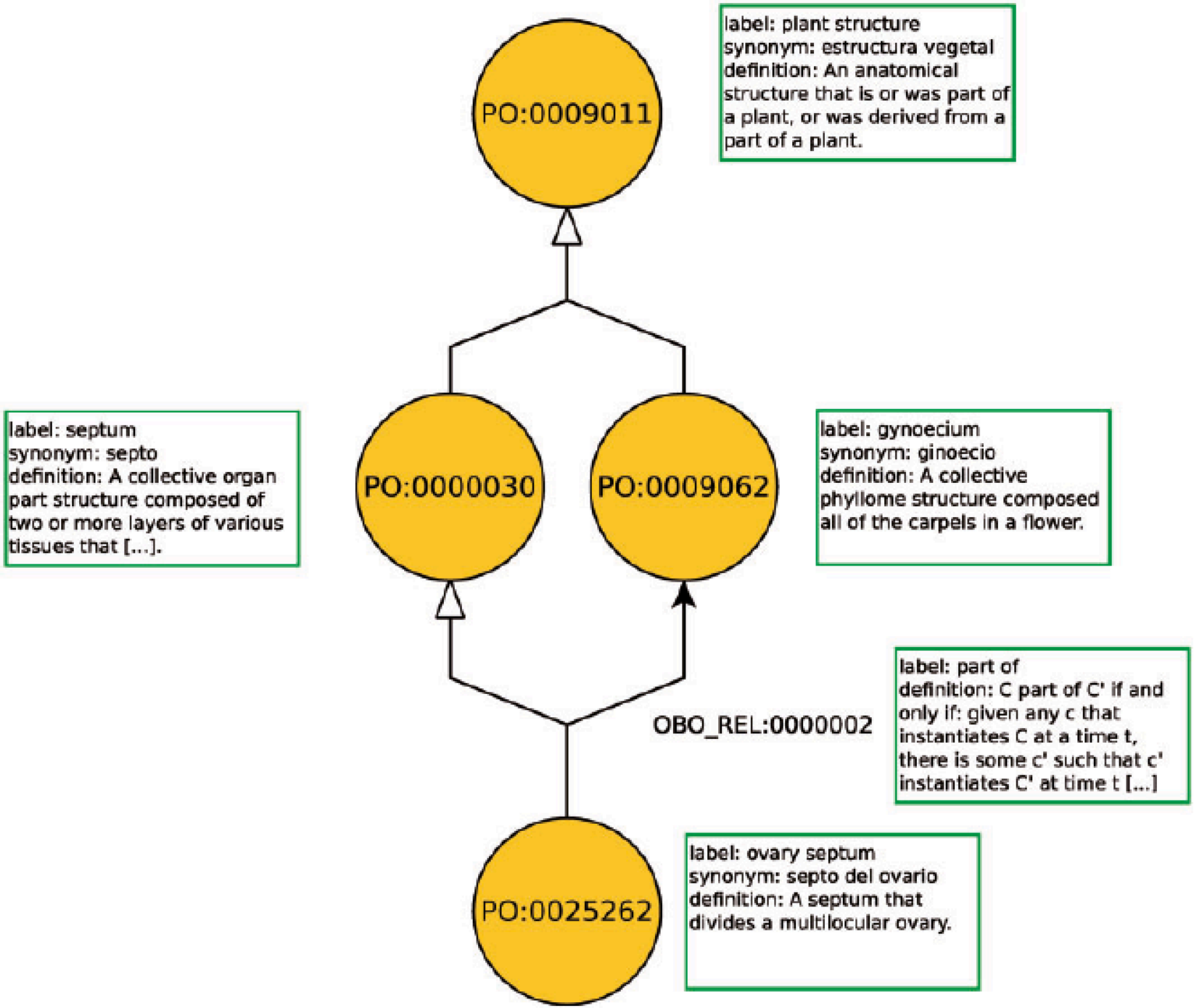
e.g. the class ‘Protein’ (referring to the set of all proteins), ‘Apoptosis’ (referring to the set of all apoptotic processes) or ‘Red’ (referring to the set of all red qualities).

- However, in contrast to sets that are defined by their extension (i.e. the entities that are part of the set), classes in ontologies are defined ‘intensionally’ by specifying the properties, features and relations that the entities belonging to a class must have [6, 9]

- Relations are similar to classes but hold for two or more entities. Examples are the relations ‘part of’, ‘participates in’ or ‘quality of’

Unique Identifiers

- classes and relations are commonly referred to using a unique identifier.
- In the Semantic Web [16], this identifier is an IRI, which is a URI supporting Unicode characters
- It is still common to use database identifier strings in biomedical databases to refer to classes and relations.
  - E.g. PO:0009011, OBO_REL:0000002
- In communities in which database identifiers are still widely used, transformation policies that standardize how database identifiers are transformed into IRIs may be adopted
  - e,g PO:0009011 → IRI http://purl.obolibrary.org/obo/PO_0009011


PO:0009011
label: plant structure
synonym: estructura vegetal
definition: An anatomical structure that is or was part of a plant, or was derived from a part of a plant.
PO:0000030
label: septum
synonym: septo
definition: A collective organ part structure composed of two or more layers of various tissues that [...].
PO:0009062
label: gynoecium
synonym: ginoecio
definition: A collective phyllome structure composed all of the carpels in a flower.
OBO_REL:0000002
label: part of
definition: C part of C' if and only if: given any c that instantiates C at a time t, there is some c' such that c' instantiates C' at time t [...]
PO:0025262
label: ovary septum
synonym: septo del ovario
definition: A septum that divides a multilocular ovary.

Domain vocabulary

- Ontologies provide a set of labels associated with the classes and relations.
- Labels are strings that are used to refer to the kind of things a class or relation represents.
- Labels may be provided in multiple languages, and multiple labels may be assigned to one class.
- Additionally, a primary label may be distinguished from secondary labels or synonyms.
  - the primary label is what is used to refer to a class or relation
  - additional labels and synonyms are used to refer to the phenomena captured by a class or a relation in other contexts.

Domain vocabulary

- If an ontology aims to cover a domain completely, the set of labels associated with the ontology classes and relations provide a large set of relevant terms within that domain.
- For example, an ontology for human anatomy such as the Foundational Model of Anatomy [20] will not only contain the classes and relations relevant to describe human anatomy,
- but also provide a large set of terms used to refer to human anatomical structures and the ways in which they may be related (as labels of the relations).

Textual definitions, descriptions and metadata

- A third feature of ontologies is the provision of information about the kind of phenomena a class or relation is supposed to capture.
- The majority of ontologies contain two main kinds of additional information:
  - the first is intended primarily for users of the ontology and provides textual definitions, examples and background information that makes the intended meaning of a class in the ontology as precise as possible to ontology users;
  - the second is additional technical information that relates one class to entries in other databases, literature or other ontologies and vocabularies.

Textual definitions

- Most ontologies in biomedicine that are primarily intended for data annotation across multiple databases provide textual definitions for their classes.
- There has been some discussion about what constitutes a 'good' textual definition in ontologies [21].
- In some domains, ontology users have opted to use Aristotelian definitions, however, other types of textual definitions are widely used as well [22].
- Ideally, the textual definitions are sufficient for an ontology user to understand exactly what kinds of phenomena a class in an ontology refers to

Aristotelian definitions

- Definitions that state the general kind of thing that a class or relation represents, coupled with the properties that distinguish it from the general kind (the 'genus–differentia' model).
- a broad category or kind (the genus) is defined and then a specification of distinctive features (the differentiae) that set it apart from all the other things of this kind are listed.

D is a **B** that **C**

- So a D is a kind of B and C are the discriminating characteristics that differentiate (in the classification sense) all Ds from other Bs

Examples

- A B cell can be defined as: a lymphocyte that expresses an immunoglubulin complex.

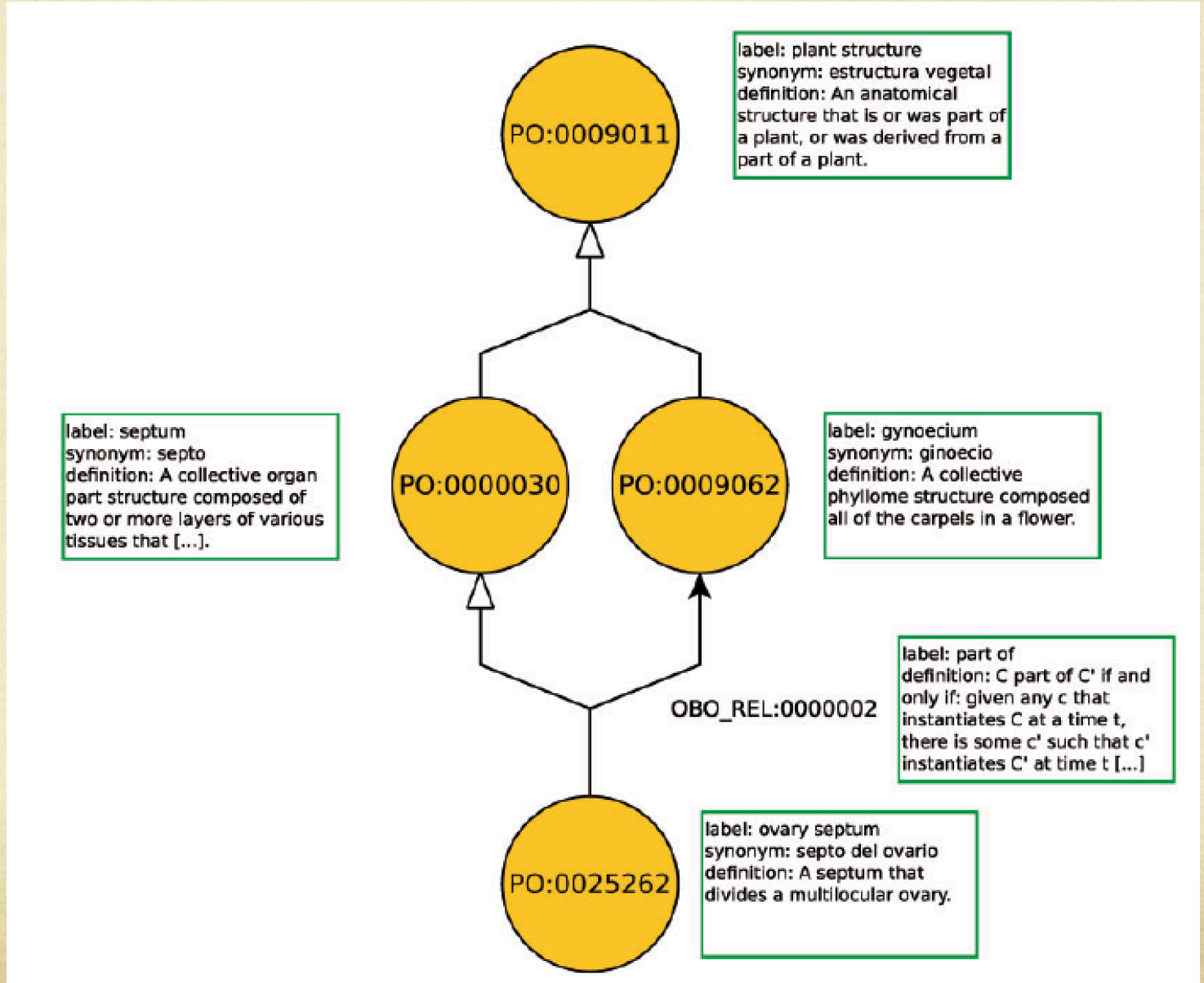
This completely disambiguates the term and ensure its hierarchical placement as well as it's error-free mapping to other ontologies.

- An 'ovary septum' can be defined as a 'septum' (the general kind) that 'divides a multilocular ovary' (the conditions or properties that separate it from others within the general kind).

Formal definitions and axioms

- Finally, ontologies provide 'formal' and 'machine-readable' definitions and axioms.
- These are some of the most valuable features of ontologies, as these may enable:
  - graph- and network-based analyses,
  - 'fuzzy' matches in searches,
  - verification of data consistency,
  - as well as provide background knowledge about a domain and reveal new knowledge through deductive inference.
- The axioms and definitions of ontologies can be represented in many forms.

In some cases, they are expressed directly as a graph structure that is intended to represent a taxonomy or a partonomy.



Ontology Formal Languages

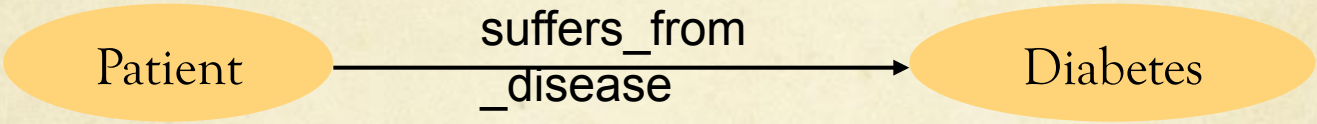
- In other cases, axioms and definitions are written in a formal language.
- Ontologies are increasingly being expressed directly in a formal language, and graph representations of ontologies are being derived dynamically from this formal representation.

Resource Description Framework (RDF)

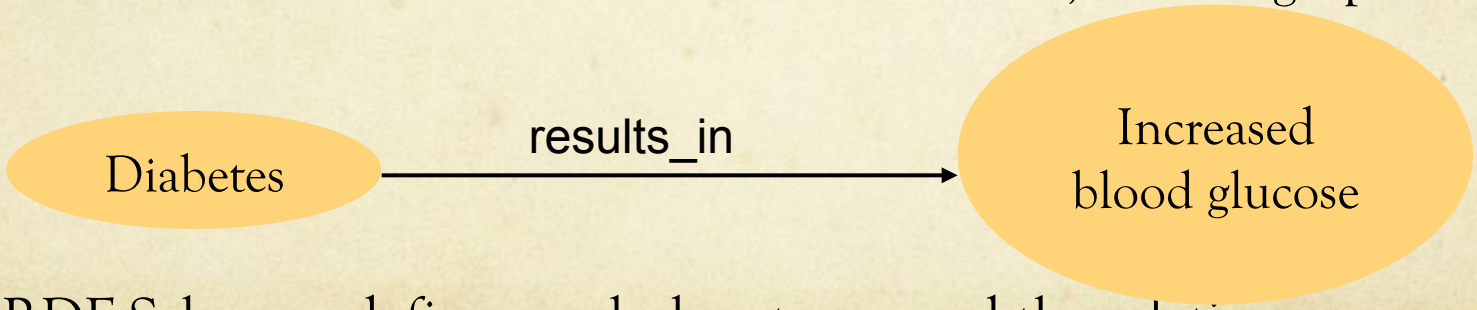
- RDF is graphical formalism
  - for representing metadata
  - for describing the semantics of information
- RDF describes resources
  - Classes and properties
  - Sub/super-classes (and properties)
  - Range and domain (of properties)
- Advantages
  - Separates data management from data presentation, making both processes more efficient
  - It can handle multiple metadata schemas in the one record
  - It is easier for computers to understand
  - It can group elements and supports complex values

RDF Data Model

- Statements are <subject, predicate, object> triples



- Statements describe properties of resources. The subject of one statement can be the object of another
- Such collections of statements form a directed, labeled graph



- RDF Schema - define vocabulary terms and the relations between those terms

RDF limitations

RDFS is too weak

- No localised range and domain constraints
  - E.g. can't define the range of the relation suffers_from_disease is human disease when applied to human patients and animal disease when applied to animals
- No existence/cardinality constraints
  - E.g. can't express that all instances of person have a mother that is also a person, or that persons have exactly 2 parents
- No transitive, inverse or symmetrical properties
  - E.g. Can't say that isPartOf is a transitive property, that affects is the inverse of is_affected_by or that touches relation is symmetrical

Such limitations proved difficult to provide reasoning support for RDF descriptions.

Requirements

- Extend existing standard
  - Such as XML, RDF, RDFS
- Easy to understand and use
- Formally specified
- Expressive power
- Reasoning support

From RDF to OWL

Ontology Inference Layer (OIL)

(a precise semantics for describing term meanings)

DAML-ONT

(builds on RDF and RDF Schema with richer modelling primitives)

DAML+OIL

(combination of features)

Web Ontology Language (OWL)

(W3C recommendation standard)

Web Ontology Language (OWL)

- Three species of OWL
  - OWL full, OWL DL, OWL Lite
- Benefits
  - Well defined semantics based on description logic
  - Formal properties
  - Reasoning support
  - Semantic Web

Formal Languages in the Biomedical Domain

- Web Ontology Language (OWL) [23], based on description logics [24, 25], is the common language for bio-ontologies nowadays.
- The graph-based OBO Flatfile Format, still used by several ontologies , has now become a sub-language of OWL [27]
- OBO can be processed with the same tools and libraries used for OWL ontologies.

OBO format

- Evolved from DAG representation initially created for Gene Ontology
- Adopted by dozens of biomedical ontologies stored in the OBO repository
- Used by most GO-based data analysis tools
- Designed for
  - human readability
  - extensibility
  - minimal redundancy

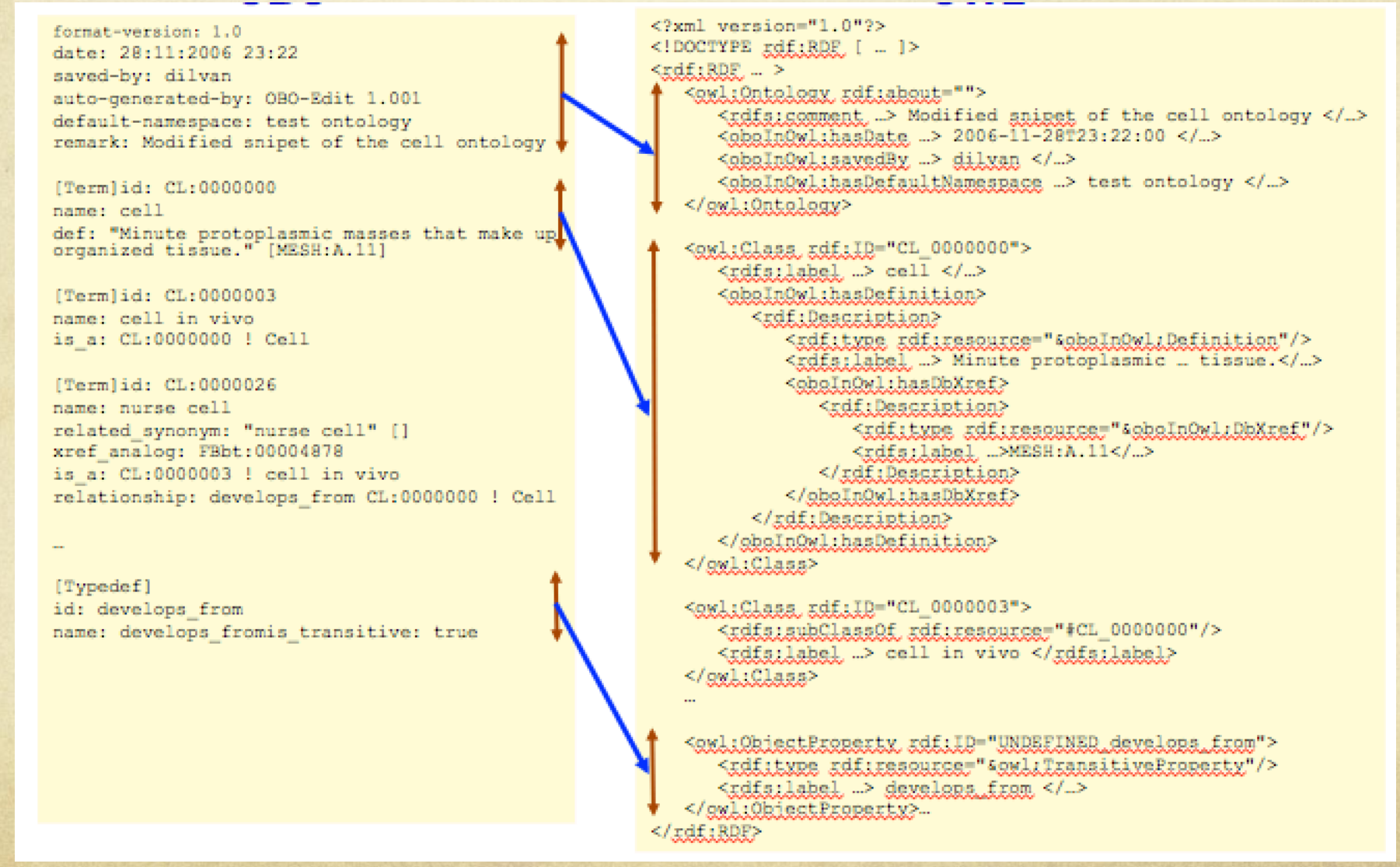
OBO to OWL mapping

- Many bio-ontologies are modeled in OWL (e.g., NCI Thesaurus, BioPAX, SNOMED-CT)
- OBO format has not been adopted outside the bio-ontology community, where OWL is the recognized standard
- Bio-ontologies need to interoperate with other ontologies used throughout e-science
- The standardization of OWL is leading to many commercial ontology-oriented tools that biologists might wish to use

Common Ground

- Semantic information
  defines classes and relationships about which computers can reason automatically
- Textual properties
  intended meanings of ontology elements for human consumption; such entries include names, textual definitions, descriptions, usage notes, and so on

Mapping example



Linked Data

- Linked Data [122] represents a method of publishing and sharing data on the web.
- data items are identified through a URI, and links to other data items are included explicitly referring to their URI.
- The URIs used to denote data items should be dereferencable, i.e. it should be possible to obtain additional information about the item through the URI (depending on the method used to access the URI, the information could be presented as HTML, RDF, JavaScript Object Notation or similar).

Proprietary graph-based ontology representation formats

- A number of graph-based representations of ontologies have been developed that primarily specify labeled graphs.
- Examples include the representation of the Medical Subject Headings thesaurus [123], the Unified Medical Language System [124] or the medical vocabulary SNOMED CT [125].

Axiomatic method

- The construction of ontologies in a formal language often follows—explicitly or implicitly—the axiomatic method [28].
- According to the axiomatic method, knowledge about a domain is formalized by first introducing
  - a **set of terms** referring to **classes and relations** in the domain (the classes and relations of the ontology),
  - and then **explicitly defining these classes and relations** by reference to other terms or relations, and possibly introducing new terms and relations.

Example

ovary septum' (PO:0025262) could be defined using the OWL language as:

```
'ovary septum' equivalentTo: septum and
divides some 'multilocular ovary'
```

This definition states that the class

ovary septum is equivalent to the expression **septum and divides some 'multilocular ovary'**

ovary septum' is now a shorthand form of the complex statement (i.e. every occurrence of 'ovary septum' could be replaced with the expression on the right).

- A definition alone does not add any information about the intended meaning of a class.
- the meaning of '**ovary septum**' now depends entirely on the meaning of '**septum**', '**multilocular ovary**' and the relation '**divides**'.
- Following the axiomatic method, we can introduce further definitions for some of these terms. For example, '**multilocular ovary**' could be further defined:

```
'multilocular ovary' equivalentTo: ovary
and has-quality some multilocular
```

- Similarly, since this takes the form of an explicit definition (through the use of the equivalentTo: keyword), we can now replace every occurrence of ‘multilocular ovary’ with the expression on the righthand side.
- Applying this property of explicit definitions, we can rewrite the definition of ‘ovary septum’ as:

```
'ovary septum' equivalentTo: septum and
divides some (ovary and has-quality
some multilocular)
```

- Now, the meaning of the class ‘ovary septum’ depends on the meaning of the classes ‘septum’, ‘ovary’, ‘multilocular’, as well as the relations ‘divides’ and ‘has-quality’.
- We could continue defining these classes by introducing additional classes and relations.
- However, inevitably, we will come up with a set of classes and relations that we cannot further define.

Axioms

- Axioms are statements that we consider to be true in the domain they are supposed to represent.
- Axioms form the features of ontologies that provide domain knowledge and fill the classes and relations with meaning.
- For example, we could state about the 'has quality' relation that,

if an entity x has the quality q, and an entity y has the quality q, then x must be identical to y (i.e. a quality is always the quality of at most one entity).

In OWL, we could state this simply as

```
ObjectProperty: 'has quality'

Characteristics: InverseFunctional
```

Axioms

- Another kind of axiom is the 'subClassOf:' axiom in which one class is asserted to be a subclass of another class.

A class X is a subclass of Y **if and only if** all instances of X are also instances of Y (i.e. all things satisfying the conditions for X also satisfy the conditions for Y).

E.g. **part of** in OWL :

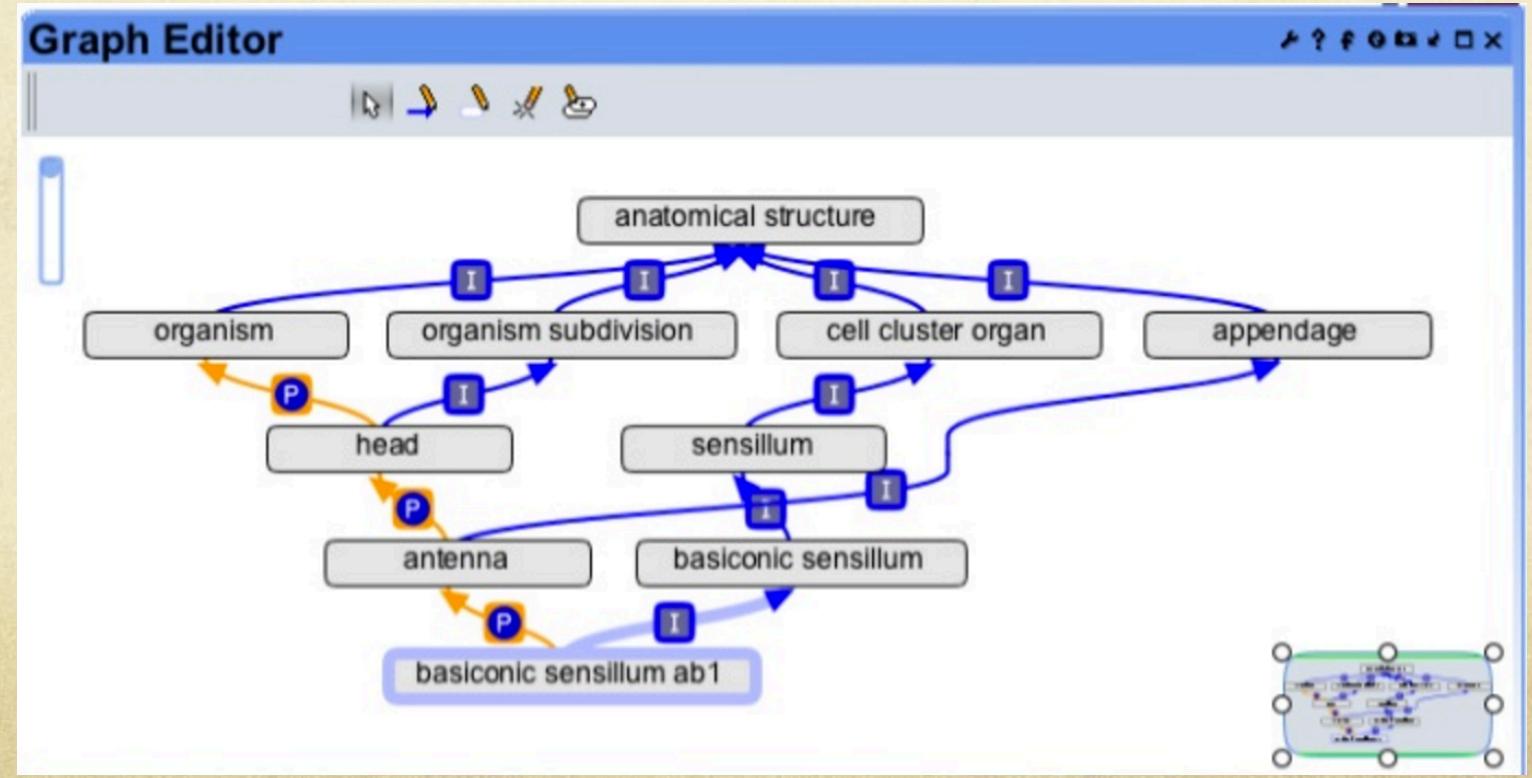
```
'ovary septum' subClassOf: 'part of'
some gynoecium
```

Use of axioms

- The axioms and definitions in ontologies can give rise to a graph structure that can be exploited using graph- and network-based algorithms.
- In these graphs, nodes commonly represent classes, and edges represent types of axioms that hold between these classes [35].
- In particular, ontologies give rise to '**taxonomic graphs**', which represent the subclass relations between the named classes in the ontology.

- Another pattern that is frequently used in generating a graph structure from ontology axioms is the existential restrictions on the 'part of' relation to give rise to a **partonomy** [36].

- Here, an edge labeled 'part of' is generated between classes X and Y if X is a subclass of **'part of' some Y**



Using Ontologies

Several tools and methods have been developed that make use of ontologies and support their use.

These tools often focus on one or two of the features of ontologies. Some of the main usage examples include:

- Annotation and data integration
- Ontologies as vocabularies
- Formalized definitions and axioms: reasoning with ontologies
- Mining and analyzing multimodal data with ontologies

Annotation and data integration

- The use of standard identifiers for classes and relations in ontologies is a key component in enabling data integration across multiple databases
- Reason → same identifiers can be used across multiple, disconnected databases, files or web sites.
- Consequently, these identifiers are widely used in structured file formats, in knowledge bases and data repositories

First application of Gene Ontology

- Differential expression screens and Serial Analysis of Gene Expression (SAGE) analyses generated data sets of often thousands of genes, which needed to be interpreted in terms of gene function.
- This provided the impetus behind the ongoing functional and structural annotation of gene products, which is now available through the GO database [38] and is a mainstay of modern bioinformatics.
- GO enabled the assignation of functions to gene products and the ability to compare these functions computationally within and across species; these features have become key tools in functional and comparative genomics.

Ontology-based annotations

- At its core, an ontology-based annotation associates an entity and an ontology class, and combines this assertion with metadata that contains, among others, information about who created the annotation, the date at which the annotation was created or the evidence that was considered.
- The entity that is annotated can be represented by an identifier in a database, referred to by a word or phrase in text, or even visually represented in an image [39, 40].
- Annotation tools are concerned with recording the annotation in standard formats, performing basic quality checks and providing the metadata for the annotations, as well as suggesting or inferring ontology-based annotations using custom algorithms.

Annotations tools

- The majority of annotation tools allow for the inclusion of provenance information, such as the evidence for an ontologybased annotation as recorded using the Evidence Code Ontology [41] or the Provenance Ontology [42].
- Tools such as Domeo [43] an annotation framework applied among others by the Neuroscience Information Framework and the OpenPhacts projects, uses the Annotation Ontology [39] to formally capture provenance information associated with ontology-based annotations.
- Furthermore, an increasing number of annotation tools use the W3C Open Annotation Data Model [44], or are able to import and export annotations in this format.

Annotation tools examples

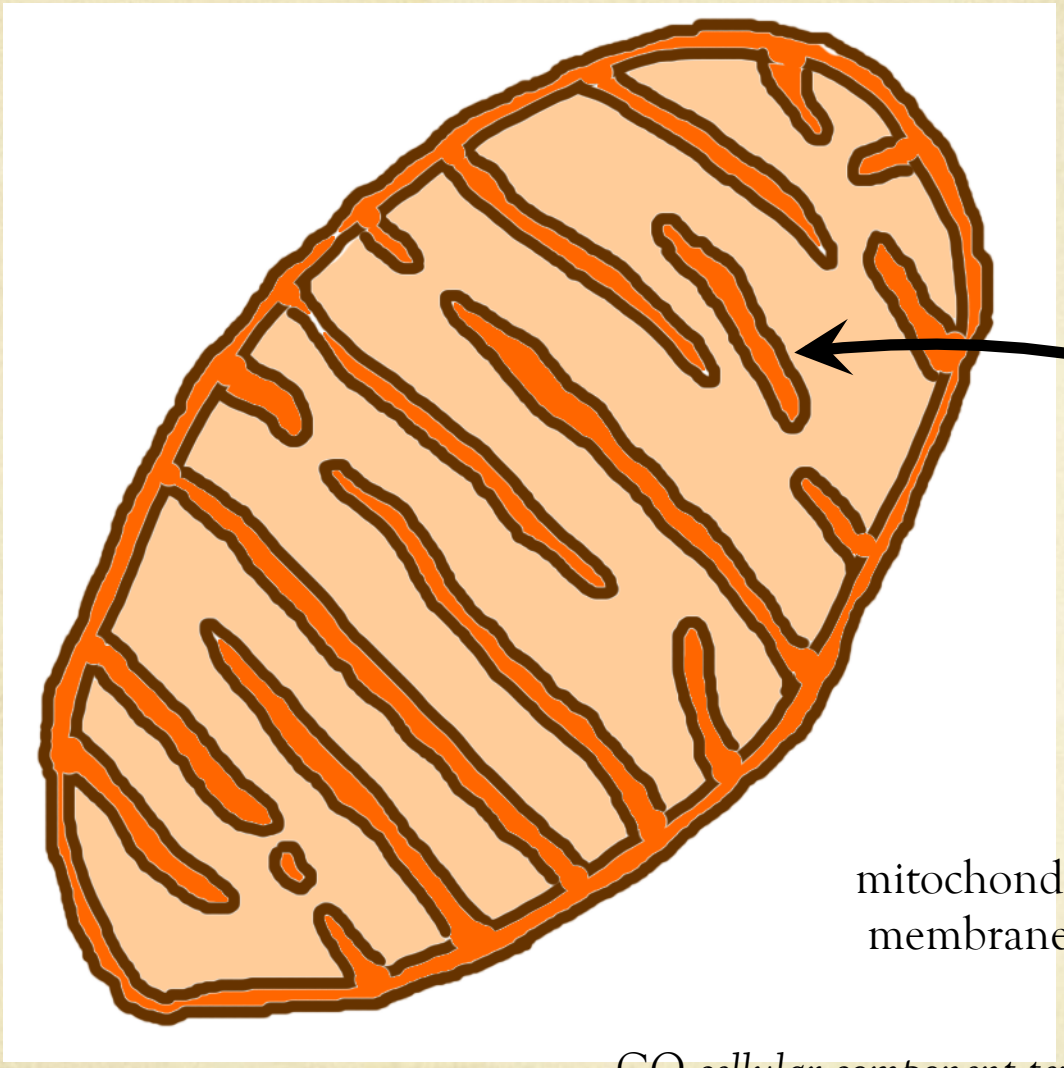
- Annotation tools that support curators through markup of literature are widely used to suggest possible annotations [45].
- Some examples include:
  - the Textpresso software tool [46] - supports literature curation for GO - extensively used in model organism databases [47].
  - the Phenex tool → phenotype annotation of character matrices in the Phenoscape project [48].
    - Phenex contains workflow elements and inbuilt reliability algorithms that aim to reduce curator workload [49].
  - the Phylogenetic Annotation and Inference Tool → assists infer annotations among members of a protein family based on sequence orthology [50],

An example...

Mitochondrial P450

(CC24 PR01238; MITP450CC24)

Where is it?



GO *cellular component term:*
GO:0005743

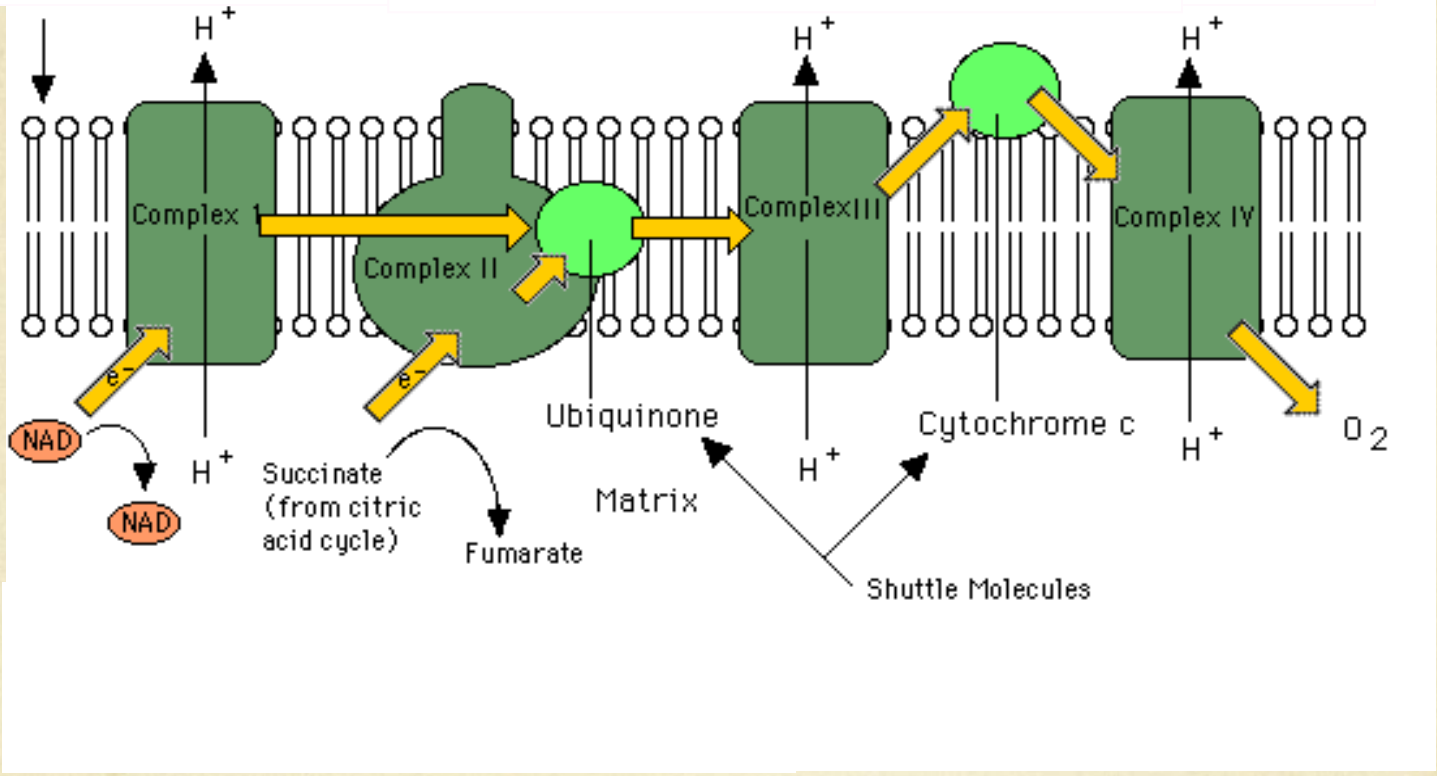
What does it do?

substrate + O_2 = CO_2 +H_20 product

monooxygenase activity

GO *molecular function term:*
GO:0004497

Which process is this?

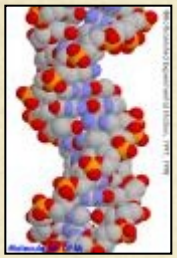


electron transport

GO biological process term:
GO:0006118

http://ntri.tamuk.edu/cell/mitochondrion/krebpic.html

GO annotations



Gene Product

PubMed

Reference

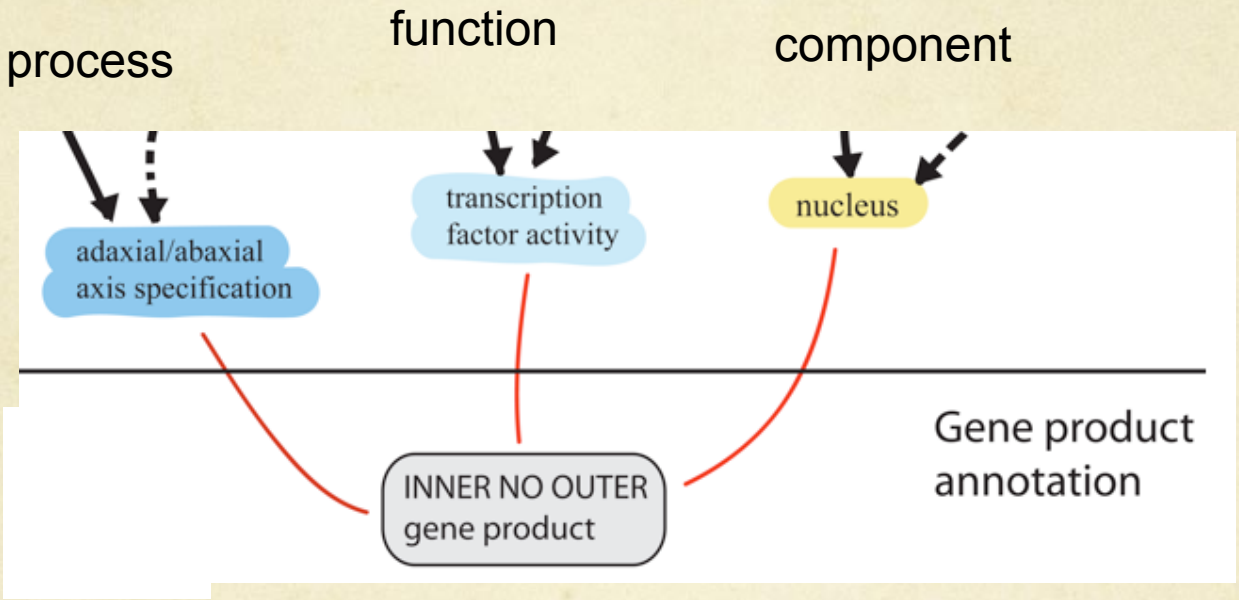
- horsetail movement
- male meiosis
- meiosis I
  - achiasmate meiosis I
  - homologous chromosome segregation
  - meiosis I nuclear envelope disassembly
  - meiosis I nuclear envelope reassembly
  - meiotic anaphase I
  - meiotic G2/MI transition
  - meiotic gene conversion
    - meiotic DNA double-strand break formation
    - meiotic DNA double-strand break processing
    - meiotic DNA recombinase assembly
    - meiotic DNA repair synthesis
    - meiotic heteroduplex formation
    - meiotic mismatch repair
    - meiotic strand displacement
    - meiotic strand invasion

GO Term

IMP, IGI, IPI, ISS, IDA, IEP, TAS, NAS, ND, RCA, IC

Evidence Code

GO annotations

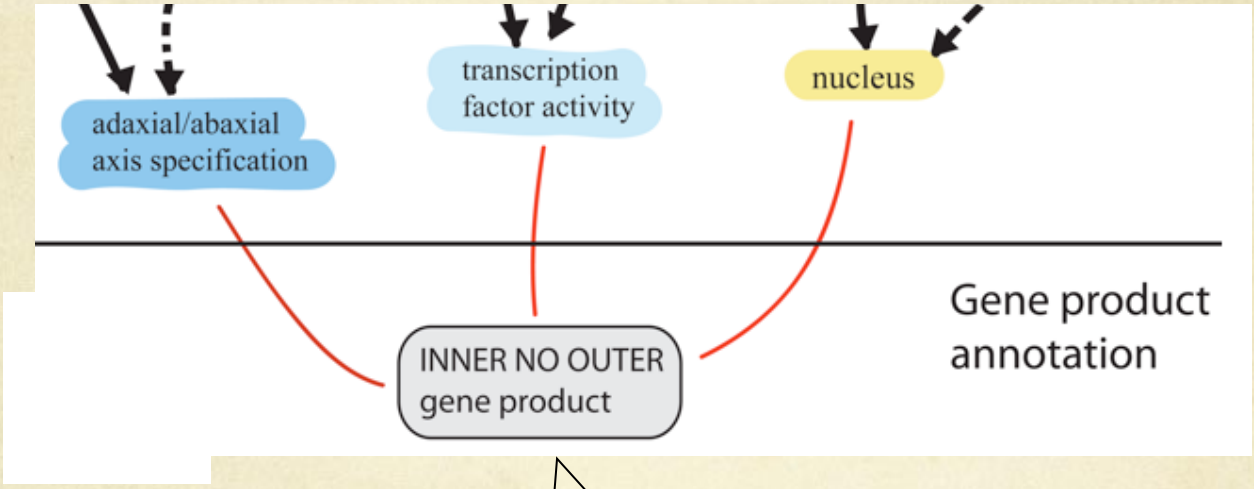


The gene product inner no outer is involved in adaxial/abaxial axis specification.

process

function

component

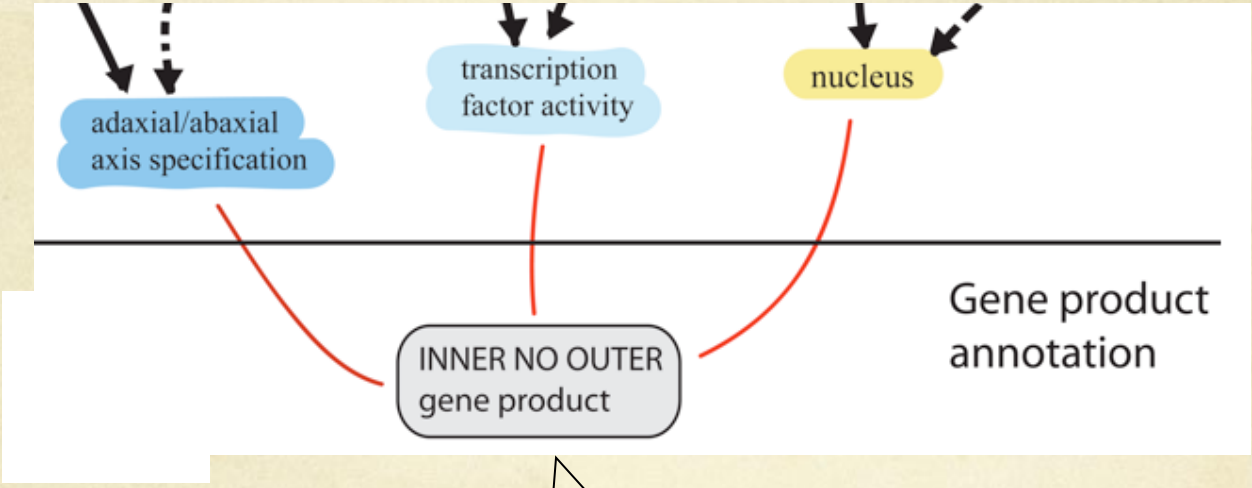


The gene product inner no outer has transcription factor activity.

process

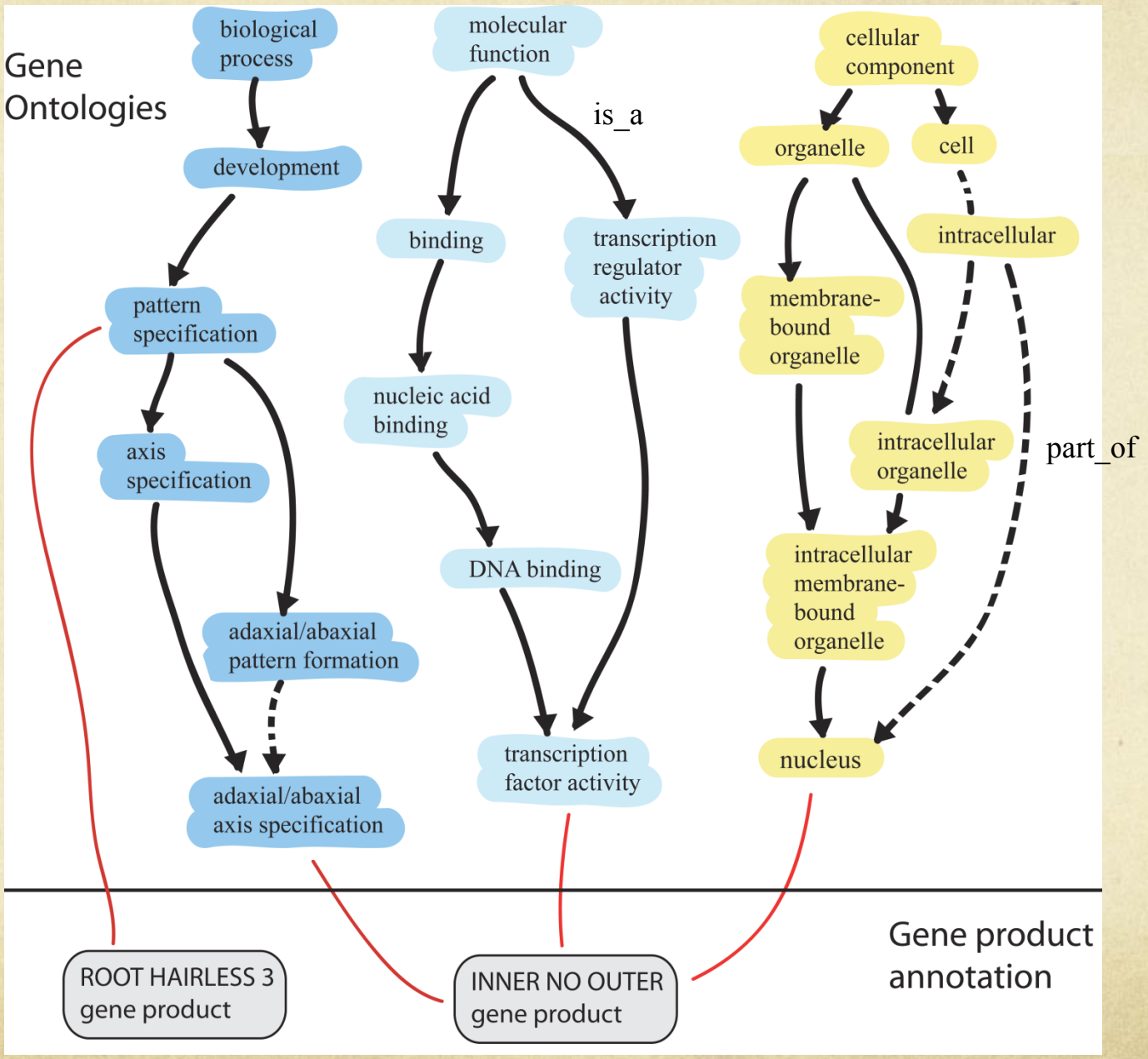
function

component



The gene product inner no outer is active in the nucleus.

Clark *et al*., 2005



Annotation vs Entity

Patient's Brain

Ontology

CT Scan

Annotation and data integration

- For complex multimodal data sets, annotation with single ontologies is often not sufficient.
- A particularly complex use-case of annotation with multiple ontologies occurs in the domain of phenotype descriptions, as applied in large-scale mutagenesis projects.
- In the Zebrafish Mutagenesis Project [51], much of the observed data is categorical and describes anatomical and physiological variation, and the phenotypic descriptions are based on anatomy and process ontologies [51].
- The International Mouse Phenotyping Consortium (IMPC) [52], on the other hand, generates both categorical data, which are assigned by investigators directly based on a phenotype ontology, and quantitative data.

- The strategy adopted by the IMPC is to express phenodeviance by assigning a class from a phenotype ontology on the basis of predetermined statistical thresholds [53, 54].
- This form of automated annotation, albeit on highly quality-controlled data, is time-efficient and facilitates data integration and mining across qualitative and quantitative information.
- When it becomes necessary to use more than a single ontology for annotation, it is beneficial to fix the ontologies that are being used to annotate a data set.
- Ontology repositories can aid in finding ontologies suitable for annotating data within a domain.

Main bio-ontology repositories

- **BioPortal** [126] is the largest ontology repository for ontologies in biology and biomedicine. It contains >400 ontologies with a total of >6 million classes.
- BioPortal can be used to find ontologies based on the ontology name or the label of a class within the ontology
- It further has a large number of web services and widgets that allow embedding of key BioPortal functions in web applications. The NCBO Annotator [127] is a part of BioPortal and can be used to find labels of ontology classes in text. BioPortal can also be accessed through a SPARQL endpoint.

http://bioportal.bioontology.org/

- **Ontobee** [128] is an ontology repository in which ontologies are presented as Linked Data.
- Ontobee provides information about the classes and relations used by the OBO project.

http://www.ontobee.org/

- The **Ontology Lookup Service** [129] consists of a repository of ontologies represented in the OBO Flatfile Format, and enables search of single ontologies, lookup of terms across multiple ontologies and browsing and visualizing the ontology graph structures.
- The Ontology Lookup Service can be accessed through a web interface and a number of web services

http://www.ebi.ac.uk/ontology-lookup/

- The **Open Biological and Biomedical Ontologies** (OBO) library [2] consists of a number of ontologies that have been developed according to a set of agreed principles including complementarity and collaborative development

  http://obofoundry.org

- **Aber-OWL [ ]** provides a framework for automatically accessing information that is annotated with ontologies or contains terms used to label classes in ontologies. When using Aber-OWL, access to ontologies and data annotated with them is not merely based on class names or identifiers but rather on the knowledge the ontologies contain and the inferences that can be drawn from it.

  http://aber-owl.net/

Ontologies as vocabularies

- Ontologies provide vocabularies of the terms used within a domain.
- Therefore, they can be used by a large variety of applications that rely on domain-specific terms.
- Example applications for the vocabulary component of ontologies include user interfaces for databases that contain ontology-based annotations, and natural language processing methods.
- Tools using the vocabularies associated with ontologies use them in two main ways.
  - Use labels to identify ontology entities or relations
  - Use labels to enable data access

Use labels to enable data access

- First, the labels of an ontology classes and relations enable access to data or text annotated with these ontologies.
- For this type of application, a link is established between a class and a user-readable name of that class.
- This link is then used to provide a way for human users of an ontology to access the information associated with the ontology class.
- Tools that use this feature include a wide range of browsers that enable access to ontology-based annotations through the class labels, such as the Amigo tool [55], which enables access to GO annotations, or GOPubMed [56], which enables access to scientific articles based on ontology classifications.

Use labels to identify ontology entities or relations

- Second, the labels in an ontology can be used to identify whether the text mentions a phenomenon characterized by a class or relation in an ontology.
- Applications of this type typically require the utilization of natural language processing technique [57].
- One example of such application is the NCBO Annotator, a tool that can recognize the labels and synonyms of ontology classes in natural language texts [58]. The National Center for Biomedical Ontologies (NCBO) Annotator implements a basic concept recognition approach [59] that generalized well across multiple vocabularies and does not require additional training.
- However, more specialized approaches have been developed, in particular in the context of recognizing descriptions of gene functions and biological processes in text [60], which can then be used to develop software tools that assist domain experts in literature-based database curation.

Ontologies and large-scale text mining

- The labels of classes in ontologies can also be used for large-scale text mining to identify system-wide associations between the phenomena to which they refer.
- Text mining based on ontologies has been used to identify the presence of disease modules based on phenotypes [61, 62], drug targets and drug indications [63, 74], drug–drug interaction [65] and candidate genes for diseases [66, 67].
- The success of these methods depends on the coverage of terms used to refer to classes in the ontology.

Text mining challenges

- The main challenge in relying on class labels to recognize the reference to an ontology class in text is that labels do not capture all of the possible linguistic variations around terms and phrases used to refer to an ontology class [68].
- Recognizing ontology classes referenced in text poses a distinct set of challenges, in particular for semantically complex classes, or classes for which no common and widely used terms have been established [69–71].

Formalized definitions and axioms: reasoning with ontologies

- The primary means to access and process ontologies semantically are automated reasoners, i.e. software tools that can directly infer knowledge from the axioms and definitions in ontologies using deductive inference.
- Automated reasoners can:
  - detect contradictions in the axioms and definitions of an ontology (consistency checking),
  - infer the most specific subclasses and superclasses for all classes in an ontology (classification) and
  - answer complex queries.
- A wide range of automated reasoners has been developed for different subsets of OWL, supporting different features and exhibiting different computational complexity for basic reasoning tasks such as answering queries

Example of reasoners

- Pellet [72] – Suports OWL 2, OWL EL

General purpose OWL reasoner with a large set of features, including specialized OWL EL reasoning, support for rules, support of epistemic operators, integration in SPARQL, explanation of inferences, incremental reasoning.

- HermiT [73] - Suports OWL 2, OWL EL

General purpose, highly optimized OWL reasoner.

- FacT++ [130] - OWL-DL, OWL 2 (partially)

Highly optimized reasoner implemented in C++

- Konklude [75] – OWL 2

Highly optimized OWL reasoner supporting parallel reasoning.

- RacerPro 2.0 [131] - OWL 2 (partially)

Optimized OWL reasoner, with integration in the AllegroGraph [132] triple store.

- TrOWL [133] - OWL 2

Scalable OWL reasoner with support for limited closed-world reasoning (negation as failure) and stream reasoning.

- ELK [74] – OWL-EL

Optimized and feature-rich OWL EL reasoner with support for incremental and parallel reasoning.

Reasoner choice

- Reasoners for subsets of OWL such as OWL-EL support less expressivity for axioms an queries in ontologies, but usually guarantee a lower computational complexity.
- For complex ontologies expressed in OWL, examples of commonly used reasoners include Pellet [72] owing to its support for a large number of features, and HermiT [73] owing to its high performance for complex ontologies.
- For ontologies expressed in the OWL-EL profile, the ELK reasoner [74] is widely used owing to its support for large ontologies and parallel reasoning.
- Recent developments include the Konklude reasoner [75], which outperforms most OWL-EL and OWL 2 reasoners even for large ontologies [76].

OWL reasoners implementation

- OWL reasoners are either implemented as stand-alone tools, or can be accessed through the OWL API [80] or the OWLLink protocol [81].
- The OWL API is a reference implementation for creating and manipulating OWL ontologies and provides interfaces for automated reasoning that the majority of OWL reasoners implement.
- OWLLink is an HTTP-based protocol for communicating with OWL reasoners.
- Reasoners can also be accessed through ontology editors such as Protege [82].

Common tools and software libraries

- Protege [82]– Suports OWL 2, OWL EL is an OWL ontology editor with full support for OWL ontologies and a large number of plug-ins that provide integration of reasoners, export and import of various ontology representation formats, or ontology visualization.
- OWL API [80] is a reference implementation and a de facto standard for processing OWL ontologies.
- owlcpp [134] is a C﬋﬋ library for processing OWL ontologies. It includes support for querying ontologies through automated reasoners.

- Brain [135] is a library based on the OWL API that provides methods for processing and reasoning with ontologies, in particular represented ones in the OWL-EL profile of OWL
- Redland RDF API - An RDF library written in C. It provides a large set of commonly used command line tools to transform or collect basic statistics about an RDF file.
- Apache Jena is a Java library and collection of tools consisting of an RDF library, integration of SPARQL queries and support for OWL ontologies.

Common analysis and visualization tools and libraries

- Gephi [136] is a generic graph-visualization tool, and can be used to visualize classes and relations in ontologies. Gephi also supports a number of algorithms for basic graph analysis, including transitive inference over edges.

  http://gephi.github.io/

- Cytoscape [137] is a tool for visualizing and analyzing interaction networks and other graphs including ontologies.

  http://www.cytoscape.org/

- The Semantic Measures Library and Toolkit [138] is a generic framework implementing a large variety of semantic similarity measures over ontologies.

  http://www.semantic-measureslibrary.org/

- Enrichment analysis uses the graph-structure underlying ontologies (usually the GO) together with transitive inference over the edges in the graph to statistically test a hypothesis. The graph structure is used to 'enrich' statistical power by propagating annotations transitively over the graph and performing a test at each level of the ontology hierarchy.

  http://geneontology.org/page/goenrichment-analysis

- OntoFUNC [139] is a software tool to perform ontology enrichment analysis over arbitrary OWL ontologies.

  http://phenomebrowser.net/ontofunc/

Reasoner-based verification of data consistency

- Most users of ontologies will not access ontologies directly through automated reasoners, but will either use the output of an automated reasoner (e.g. the inferred graph structure of an ontology) or interact with a reasoner indirectly (e.g. through a software tool that uses an automated reasoner as part of its operation).
- Nevertheless, in some approaches, automated reasoning has been applied directly to verify data consistency with respect to constraints in an ontology or reveal novel biological knowledge based on axioms in an ontology.
- The axioms in an ontology can be used to verify whether an entity described in a database is able to satisfy the conditions laid out for that kind of entity, and automated reasoning can be used to detect conflicts.

Examples

- For example, such an approach has been applied retrospectively to computational models in systems biology [83], but is increasingly being applied to ontology-based annotations at the time the annotation is made [84, 85].
- Some data exchange standards are now being designed with data verification in mind, and a prime example is the BioPAX standard for pathway data sharing, which is based on formalized knowledge in OWL [86].
- The axioms in an ontology can also be used to infer the class to which an entity belongs based on the features and descriptions of the class and the entity. An application of this is the inference of the protein family to which a protein belongs based on an ontology and automated reasoning [87].

Reasoner-based based integration

- Reasoning over ontologies can also be applied for integrating ontology-annotated data sets across different domain by systematically combining different ontologies using axioms or axiom patterns [88, 89].
- In such applications, the relationship between classes in different ontologies is identified and expressed in the form of an axiom or axiom pattern that is systematically applied to several pairs of classes.
- Prime examples of this form of integration are species-specific anatomy and phenotype ontologies [90, 91].
- Integrating data annotated with these ontologies relies on identifying homologous anatomical structures [92] and relating the classes that refer to these structures in different anatomy ontologies using axiom patterns [90, 93].

Mining and analyzing multimodal data with ontologies

- The great potential in using ontologies for data analysis lies with the possibility of combining their different functional levels, and some exciting insights into the biological properties of whole systems have been achieved by combining data through ontologies.
- For example, one of the most widely used applications for ontologies is Gene Set Enrichment Analysis [94] or similar enrichment methods.
- Such methods combine the graph structure of ontologies (axioms and definitions) with their potential for data integration (through ontology-based annotations) to provide a statistical interpretation of differences between two states with regard to the background knowledge provided by the ontology over which the enrichment analysis was performed.

Semantic Similarity

- Another analysis method specifically relying on ontologies and their annotations is the use of similarity measures to determine the ‘semantic’ distance and proximity between data items [95].
- In semantic similarity measures, the axioms and definitions of ontologies are exploited to define a similarity between annotated data items.
- Semantic similarity has widely been applied to computationally predict protein–protein interactions based on their functional similarity [96, 97], to the diagnosis of disease based on phenotypic similarity [98–100], or to the classification of chemicals based on structural similarity [101].

Machine Learning & Ontologies

- While statistical analysis of graphs or sets, or measures of semantic similarity, are well established methods that use ontologies for data mining, many machine learning and data mining algorithms that are applied to unstructured data are not yet widely used with ontologies and ontology-structured data.

- The challenges of using these methods occur both when using ontologies and ontology-annotated data as the target of a machine learning and data mining algorithm as well as when using ontologies and ontology-annotated data as features.

- When using ontologies as the target, i.e. when aiming to learn an ontology based classification for some piece of data such as the functions of a protein, several challenges arise in relation to the adoption of these traditional algorithms to ontology-based data in the biological and biomedical domains.

- These challenges primarily relate to the ‘multi-class’ nature of the problem, as ontologies have often very large numbers of classes, the ‘structured dependency relations’ between these classes (i.e. the axioms in the ontology) and, in many cases, the ‘multi-label’ nature of the classification problem as data items are usually annotated to more than one ontology class.

- When using ontologies, or ontology-annotated data, as features in a machine learning task, challenges relate to the large number of classes that are often sparsely populated (more specific classes are usually present less frequently while more general classes are used more frequently), and again the dependency relations between classes (e.g. disjointness, subclass relations and axiom patterns that exist between classes).

- The use of ontologies can help address a challenge that machine learning and data mining approaches face: the incorporation of different types of features for multimodal learning and classification [108].

- Combining information from text, images, videos, molecular data or structured data in knowledge bases to improve classification can be facilitated through the use ontologies, by first extracting relevant features from each type of information and representing the results using a single ontology that combines the information used for training a classifier.

Some future steps...

- There are now sufficient stable ontologies to permit routine reuse of classes from multiple ontologies in automated or semiautomated ontology construction algorithms [109].
- With increasing size and number of ontologies, the ability to modularize ontologies to generate application-specific 'views' while maintaining interoperability with data sets in a domain that are annotated with another module of the same ontology will become essential.
- A recent example of this is provided by the Bioassay ontology [110] or the automated generation of phenotype ontologies [111, 112].

- Coverage and quality of content in established ontologies must be further improved [113] → requires the sustained engagement of domain experts.
- One major application of exploiting multiple ontologies is to formalize the large, unstructured, multimodal and often distributed data from clinical records.
- It is now possible to capture information and knowledge related to diagnostic procedures, drugs, phenotypes, diseases and genotypes using existing ontologies, and there are efforts to create ontologies for capturing other environmental and behavioral data for patients.

- Ontologies are now being applied in a clinical setting [114], but mainly for data mining from partially structured and legacy clinical records [115].
- Incorporating ontologies directly in the electronic health record → novel methods for patient classification and stratification, and the analysis and mining of large-scale patient data.
- Increasing numbers of whole exome and genome sequences in clinics → ontology-based enrichment algorithms or incorporating results from basic biological research into clinical decision making [116].